

An approach to simplify the learning space for robot learning control¹

Kuu-young Young*, Cheng-Chu Fan

Department of Control Engineering, National Chiao-Tung University, Hsinchu 300, Taiwan

Received March 1996; revised September 1996

Abstract

Although learning controllers are considered to be capable of generalization, most robot learning control schemes either need to include conventional controllers, or need to repeat the learning process each time a new trajectory is encountered. The main reason for this deficiency is that the learning space for executing general motions of multi-joint robot manipulators is too large. In this paper, we propose an approach, motivated by the equilibrium-point hypothesis in human motor control, to simplify the learning space when learning controllers are used to govern robot motions. In the proposed approach, the motion command is formulated in the form of three square pulses in alternate directions with adjustable heights and widths. When the motion command is specified in this simple form, the learning space for dealing with variations exhibited in different movements is dramatically simplified. Thus, we can then implement a fuzzy system for robot motion control, which generates appropriate controlled parameters for the motion commands by using a reasonable number of rules. Theoretical analyses and simulations are performed to demonstrate the feasibility of the proposed approach. © 1998 Elsevier Science B.V.

Keywords: Robotics; Learning control; Learning space simplification; Human motor control; Fuzzy system

1. Introduction

Compared to conventional controllers, learning controllers are capable of tackling highly complex dynamics without explicit model dependence and identification. In addition, learning controllers are computationally efficient after training. Two well-known types of learning controllers for robot motion control are fuzzy systems and artificial neural networks [19, 22]. The structure of artificial neural networks is modeled after the organization of the brain, although the similarity between the two is

actually slight [19]. On the other hand, fuzzy systems are meant to encode pieces of knowledge presented by experts [10, 14, 18]. In some previous research involving the application of these two types of learning controllers for governing a multi-joint robot manipulator, they are used as subordinates, while a conventional control algorithm, e.g., PD or PID control, is responsible for the major portion of the control [12, 13, 15]. In this approach, the conventional control algorithm brings the system close to the desired state and the learning mechanism then compensates for the remaining error. On the other hand, some systems use learning algorithms alone to execute the control. However, although these learning controllers are considered to be capable of generalization, most of them need to repeat the learning process each

* Corresponding author.

¹This work was supported in part by the National Science Council, Taiwan, under grant NSC 83-0422-E-009-065.

time a new trajectory is encountered [9]. The main reason for this deficiency is that a huge number of training patterns will be demanded for the setup of a learning controller for executing general motions of multi-joint robot manipulators. Consequently, a neural controller would consist of a number of neurons or a fuzzy system requires numerous rules. This learning space problem severely hinders the application of learning control.

Learning control is biologically inspired and intended to emulate human motor control. Then, a problem that may be raised is how the human motor control system resolves the aforementioned learning space problem. The superiority of the human motor control system may be attributed to its salient control strategies and exceptional capability to make proper decisions using information from abundant and versatile biological sensory feedback. On the other hand, human beings are not very accurate in their movements. Then, an intuition is that the human motor control system may have the learning space simplified at the expense of accuracy in movements. Motivated by the intuition above and the equilibrium-point hypothesis in human motor control, we propose an approach to trade movement accuracy for learning space simplification. In this approach, the motion command is formulated in the form of three square pulses in alternate directions with their heights and widths adjusted according to various robot motions [11, 20, 21]. With the controlled parameters in the motion command being the heights and widths of the square pulses, the learning space for dealing with variations exhibited in different movements is dramatically simplified. The simplification of the learning space, however, is achieved at the price of accuracy in movements, because continuous control signals in conventional robot motion control schemes are approximated by signals in the form of a series of square pulses [1].

With the learning space significantly simplified, it is then feasible to design a fuzzy system for robot motion control, which can generate the heights and widths of the square pulses in the motion command by using a reasonable number of rules. The reason that we chose to use a fuzzy system for motion command generation instead of using an artificial neural network is because that empirical rules and data related to biological control systems are usually described and analyzed qualitatively instead of quantitatively [5].

Therefore, a fuzzy system is a more intuitive tool than an artificial neural network. The rest of the paper is organized as follows. Biological backgrounds and related discussions of the equilibrium-point hypothesis are given in Section 2. The feasibility and performance of the proposed approach and its application on robot motion control are demonstrated by theoretical analyses and simulations in Sections 3 and 5, respectively. The implementation of the fuzzy system for robot motion control is described in Section 4. Finally, conclusions are stated in Section 6.

2. Inspiration from equilibrium-point hypothesis

Fig. 1 shows a simplified block diagram for the neuromuscular system, which governs human limb movements. In Fig. 1, human movement is governed by a hierarchical structure [9, 17]. The central nervous system (CNS) makes movement plans according to different demands. Appropriate motor commands are then generated and sent to the peripheral neuromotor system, which modifies the motor commands via sensory feedback. The peripheral neuromotor system behaves as a local controller that adapts to different movements, loads, and environments, in addition to accepting commands from the CNS [4]. Finally, the modified commands are sent to the muscular-skeletal system for movement execution. With this hierarchical structure, the difficulty of performing complex movements is shared by the CNS at the higher level and the local controller at the lower level.

The equilibrium-point hypothesis suggests that the CNS specifies equilibrium points between agonist and antagonist muscle groups that correctly position the arm in relation to the target by indicating new sets of length–tension curves for the muscle groups [3, 7, 16, 17]. In other words, movement is treated as transition between postures. The CNS needs only select new levels for the motor commands. The subsequent result, mediated by autogenetic reflexes and the mechanical properties of the muscles, should be a smooth transition from one posture-state to another. The simplicity in the form of the control signal makes this hypothesis very attractive for motion control. However, since only one equilibrium point is selected, a control strategy based on this hypothesis would not enable us even to vary the speed of the

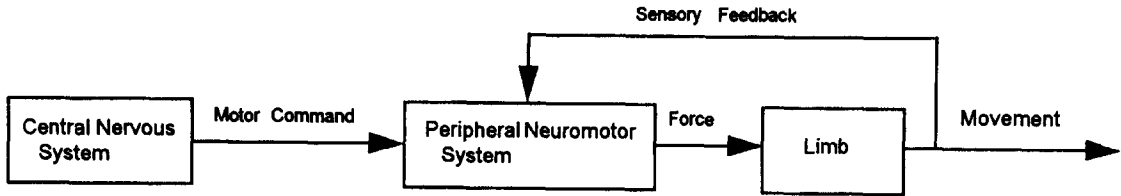


Fig. 1. A simplified block diagram of the neuromuscular system.

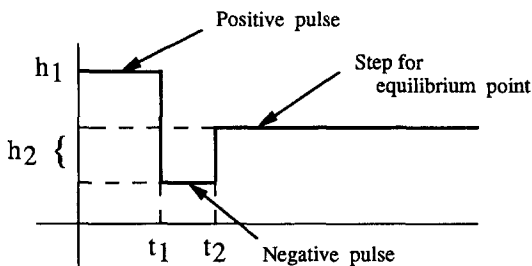


Fig. 2. Three-pulse command pattern.

movement between two given postures. To exploit the simplicity of the equilibrium-point hypothesis and make it capable of dealing with different velocities and loads in reaching various positions, the motor command can consist of a number of equilibrium points. Thus slow movements can be produced by progressive shifts of equilibrium points. Movements can be speeded up by assigning an initial shift that is larger than necessary, followed by a return to a proper static level [7].

Motivated by the equilibrium-point hypothesis, we propose a formulation of the motion command in the form of a train of square pulses in alternate directions with their heights and widths adjusted according to the demands of different tasks. Similar concepts can also be found in [11,20]. Note that the controlled parameter in the equilibrium-point hypothesis is the muscle compliance instead of the equilibrium point used in the proposed approach. Our purpose is not to propose a new biological hypothesis, but to develop control strategies inspired from human motor control for robot motion control. In light of both physiological and engineering considerations, the number of pulses in the motion command should be kept fairly small [5,11,20]. We propose to use three pulses

in the motion command, as shown in Fig. 2. The triphasic motion command pattern is clearly not an invariant pattern of muscle activation. The command pattern should be task, velocity, subject, and muscle dependent [5]. Nevertheless, the three-pulse motion command pattern has an engineering analogy in the bang-bang control, which is intended for fast movements and like the present pattern demonstrates three major force changes. The proposed command pattern also gains support from the investigation of rapid voluntary wrist movements [20], which showed that these movements normally exhibit motor commands with a three-pulse pattern for a light load: first an agonist burst, followed by an antagonist burst, followed by a second agonist burst. Correspondingly, three major alternating phases of net force were observed: a net force first in the direction of movement, then opposed to the movement, and finally in the direction of movement again. For a heavy load, however, the second agonist burst was not very apparent.

3. Proposed approach

According to the discussions above, when the three-pulse motion command pattern is used for robot motion control, the controlled parameters in the motion command will be a constrained set with few parameters. We will then develop a fuzzy system, acting as the CNS of the neuromuscular system in Fig. 1, to generate motion commands. The controlled parameters for the proposed fuzzy system are the heights and widths of the three square pulses corresponding to the equilibrium points in the motion command. It can be seen that the fuzzy system under design will face a small learning space. For a feasible

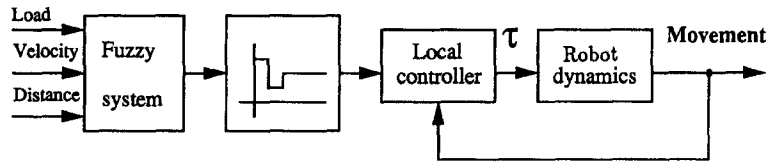


Fig. 3. Conceptual organization of the proposed control scheme.

implementation of the fuzzy system, the number of basic three-pulse command patterns for generalization will be limited. In other words, the fuzzy system will generalize a set of basic command patterns to cover a wide variety of movements. As the simplification of the learning space is at the price of inaccuracy during movement, the proposed approach demonstrates a trade-off between movement accuracy and memory utilization in learning.

A conceptual organization of the control scheme for governing motions of multi-joint robot manipulators by applying the proposed approach is as shown in Fig. 3. The inputs to the fuzzy system are distance of movement, velocity, and load corresponding to the robot motion. In turn, the fuzzy system will generate a corresponding three-pulse motion command and send it to the local controller. The local controller will then modulate the command via sensory feedback and use the resulting signal to move the robot link. Before the implementation of the fuzzy system for the robot control scheme, we will first evaluate the feasibility of applying the three-pulse motion command pattern for robot motion control from an engineering point of view.

3.1. Feasibility analysis

The three-pulse motion command pattern (shown in Fig. 2) functions as follows: first, a large burst generates a large initial speed, then a small burst brakes the movement, and finally, a step brings the limb into the equilibrium posture assigned by the CNS. In order to check whether the three-pulse motion command pattern is appropriate for controlling robot motions, we investigate the force profiles corresponding to various movements reaching the same distance with different loads and velocities. A simulation system is developed to derive the force profile from the

desired movement positions, velocities, and accelerations. Both single-joint and multi-joint movements are analyzed.

We will begin with single-joint movements. Consider the second-order system described in Eq. (1):

$$M\ddot{x} + B\dot{x} + K(x - x_d) = 0, \quad (1)$$

where M is the load, K the stiffness, B the damping coefficient, and x_d the equilibrium point. In the simulations, motion commands, and consequently the three-pulse command patterns, are introduced to the system via the assignment of the equilibrium point x_d . This second-order system can also be interpreted as having an equilibrium point at the origin and an input force specified by Kx_d . If we choose the values of K and B for a given load M properly, the system can be designed to be critically damped when the equilibrium point is fixed, which corresponds to an applied step force. Fig. 4(a) shows two desired critically damped trajectories with different natural undamped frequencies, ω_n . First we conducted a simulation to see how the applied force adjusted to deal with a heavier load when the same movement profile is preserved. The critically damped trajectory with smaller ω_n was used as a reference. The evaluated force profile, $K(x_d - x) - B\dot{x}$, can be derived by using Eq. (1) and x , \dot{x} , and \ddot{x} of the critically damped trajectory. The simulation results are shown in Fig. 4(b). The trace marked by 1 is the reference force profile when a step input is applied. The force profile corresponding to a heavier load marked by 2 exhibits a large initial bell-shaped waveform, followed by a small one in the negative direction, and finally a step at the origin. This force profile implies that the control signals are in the form of a large burst, followed by a small burst, and finally a step indicating the equilibrium point, i.e., the three-pulse motion command pattern. We performed

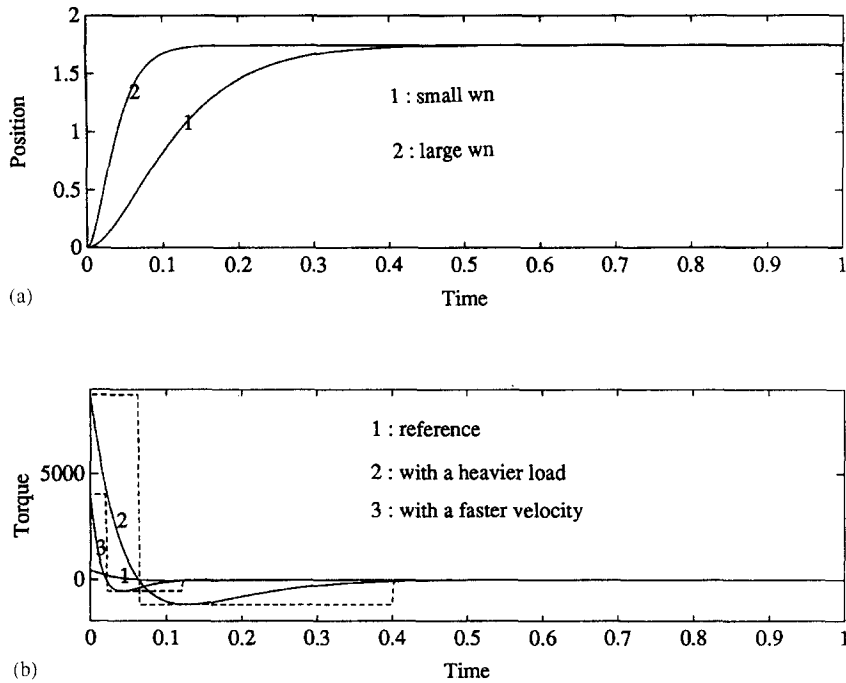


Fig. 4. Single-joint movement: (a) desired critically damped trajectories and (b) force profiles for a heavier load or a faster velocity.

another simulation to see how the force profile is affected by a larger movement velocity when the load and the distance of movement are kept the same. In this case, the critically damped trajectory with the larger natural undamped frequency ω_n , which corresponds to a smaller time constant, was used as a reference for a fast movement. Consequently, the values of K and B for this trajectory are larger than those for the previous one. The desired force profile is still derived using Eq. (1); x , \dot{x} , and \ddot{x} are for the new critically damped trajectory, and the values of K and B are those of the previous trajectory. The resulting trace is shown in Fig. 4(b), marked by 3; the trace demonstrates a force profile similar to that in the previous case.

To extend the analysis to multi-joint movements, we need to consider dynamic interaction between joints. The dynamics of multi-joint movements can be formulated as follows:

$$\tau = H(q)\ddot{q} + C(q, \dot{q}) + G(q), \tag{2}$$

where q , \dot{q} , and \ddot{q} stand for joint variables and their derivatives, $H(q)$ is the inertia matrix, $C(q, \dot{q})$ is

the vector of centrifugal and Coriolis terms, $G(q)$ is the vector of gravity terms, and τ is the vector of joint torques. The effect of gravity will be ignored in the following simulations. As we did for the single-joint movement, we generate critically damped trajectories for each of the two joints. We performed simulations to see how the force profile varies with different loads, movement velocities, and the presence of dynamic interaction. Similar to the procedure for single-joint movements, the evaluated force profile can be derived by using Eq. (2) and the critically damped trajectories for the two joints. Fig. 5 shows the simulation results for two-joint movements with a heavier load and a faster movement velocity. In Fig. 5, phenomena similar to those in Fig. 4 are observed for both joints under different load and movement velocity conditions, thus providing evidence to justify use of the three-pulse motion command pattern. In summary, the simulation results for single-joint and two-joint movements imply that the three-pulse motion command pattern captures most of the behavior of robot motion.

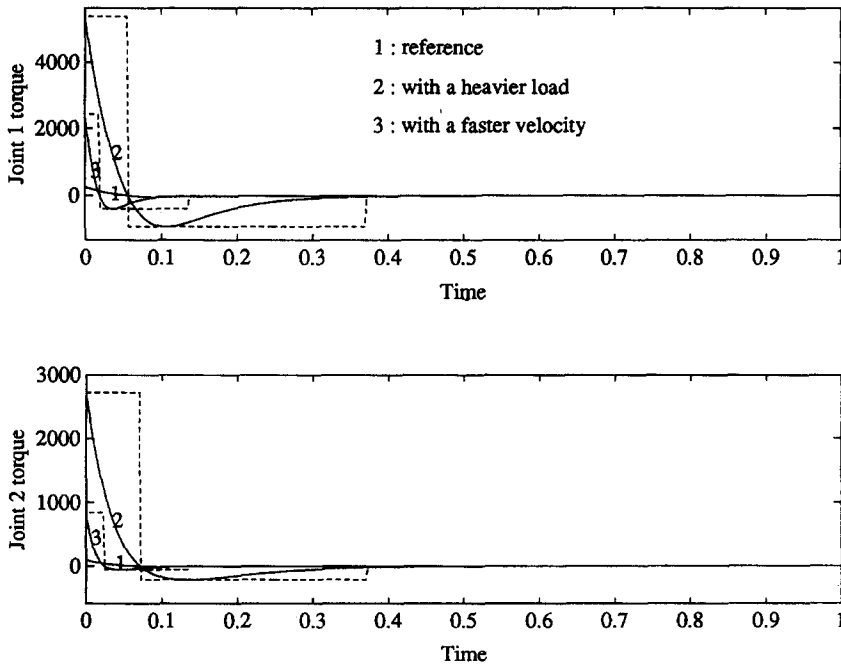


Fig. 5. Force profiles for two-joint movement with a heavier load or a faster velocity.

3.2. Local controller

As mentioned in the introduction of the neuromuscular system in Section 2, the three-pulse motion command pattern will be modified by the peripheral neuromotor system before it reaches the limb. The mechanical properties possessed by the peripheral neuromotor system play an important role for supporting the simple control strategies employed by the CNS. Especially when compliant motions are involved, for which different environments exhibit various stiffnesses and viscosities, the influence of the mechanical properties is significant [8]. To replicate the mechanical properties of the peripheral neuro-motor system adequately, a complicated local controller will be needed. In tackling free motions, however, the local control system may not be so crucial. Therefore, for the purpose of simplicity we will attribute the intelligence of the proposed scheme to the fuzzy system and the three-pulse motion command pattern, and use the local controller to deal with sensory feedback only. Because no desired velocity is specified in the three-pulse motion command pattern, we propose

a simple position control law with linear damping for this local controller:

$$\tau = \mathbf{K}_p(\mathbf{q}_d - \mathbf{q}) - \mathbf{K}_d\dot{\mathbf{q}}, \quad (3)$$

where \mathbf{q}_d stands for the vector of the equilibrium joint positions and \mathbf{K}_p and \mathbf{K}_d are symmetric positive definite matrices. The actual positions and velocities, \mathbf{q} and $\dot{\mathbf{q}}$, are obtained via sensory feedback.

3.3. Stability analysis

Another main concern of the proposed approach is stability using the three-pulse motion command pattern and the local controller. Since the first two pulses of the motion command pattern can be taken as the transient, we may in turn analyze the stability of the multi-joint system under step inputs and the control law described in Eq. (3). In the absence of gravity, the system dynamics described in Eq. (2) can be rewritten as

$$\tau = \mathbf{H}\ddot{\mathbf{q}} + \mathbf{C}'\dot{\mathbf{q}}, \quad (4)$$

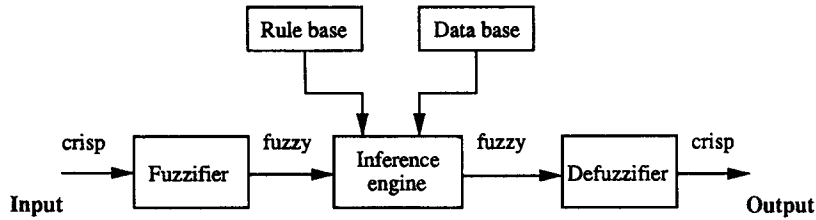


Fig. 6. Proposed fuzzy system.

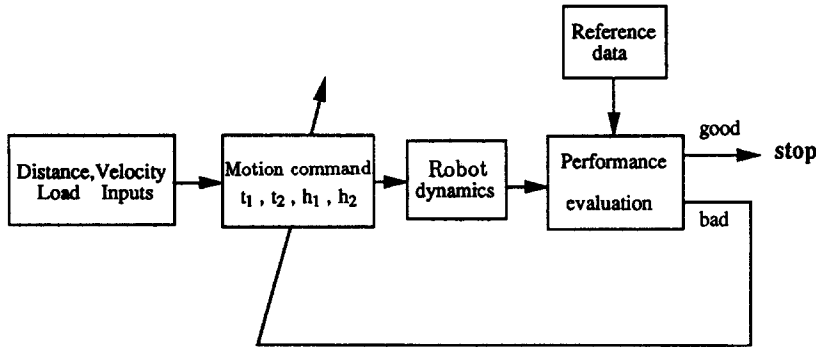


Fig. 7. Learning process for deriving reference motion commands.

where $C' = C'(q, \dot{q})$ is a matrix and $(\dot{H} - 2C')$ is antisymmetric [1]. Consider a Lyapunov function candidate

$$V = \frac{1}{2}(\dot{q}_e^T K_p q_e + \dot{q}^T H \dot{q}), \tag{5}$$

where $q_e = q_d - q$. After some manipulation using Eqs. (3)–(5), we obtain

$$\dot{V} = -\dot{q}^T K_d \dot{q} \leq 0. \tag{6}$$

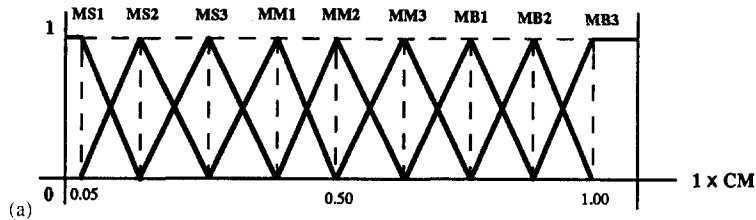
Thus the system is proved to be stable using the three-pulse motion command pattern and the local controller.

4. Implementation of the fuzzy system

In order to realize the robot control scheme in Fig. 3, we need to implement a fuzzy system for generating proper three-pulse motion commands according to different distances, velocities, and loads. As four vari-

ables, $h_1, h_2, t_1,$ and $t_2,$ need to be assigned for the motion command as indicated in Fig. 2, there will be three inputs and four outputs for this fuzzy system. We will first develop a fuzzy system to control a second-order system. Later, to control multi-joint movements, the fuzzy system for the second-order system will be applied to each of the joints via proper scaling. To deal with dynamic interaction present in multi-joint movements, in some sense our design takes advantage of the fuzzy relation incorporated in the fuzzy rules via open-loop control. It can be seen that the compensation effect via the robustness of the fuzzy system will diminish along with the increase of the number of joints. At the current stage of the study, we do not intend to develop a fuzzy system that can govern general motions of multi-joint robot manipulators. Instead, we will concentrate on demonstrating the effect of the combination of the proposed motion command pattern and the fuzzy system in learning space simplification and coverage.

Membership



Membership

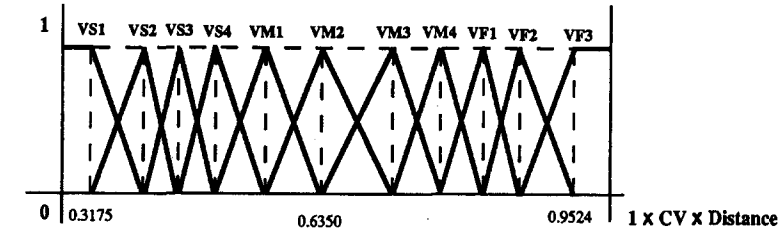


Fig. 8. Membership functions for (a) load and (b) velocity.

Fig. 6 shows the structure of the proposed fuzzy system. The system will function as follows: first, the crisp inputs are fuzzified through a fuzzifier, then the fuzzy data are processed by an inference engine which takes information from the rule base and data base, and finally, they are defuzzified to obtain the crisp outputs. Note that, although three inputs – distance, velocity, and load – are called for in this fuzzy system, only velocity and load are input to the fuzzifier. This is because the input distance will be directly proportional to the final step of the three-pulse motion command under the equilibrium-point hypothesis and so it requires no further processing. In order to construct this system, proper membership functions need to be derived for the fuzzifying process. In addition, the rule base and data base, representing the fuzzy reasoning process, have to be developed. Therefore proper motion command patterns corresponding to various conditions need to be provided, so that the membership functions, rule base, and data base can be implemented according to these reference commands.

We will generate the reference commands by engineering means, instead of collecting data by

performing extensive experiments on human limb movements. In our design, the reference motion command patterns are derived by a learning process, as shown in Fig. 7. In Fig. 7, various combinations of t_1 , t_2 , h_1 , and h_2 are first assigned for the three-pulse motion commands. Then the commands are sent to move the links. In turn, the performance of the movement solicited is evaluated by comparing it with the desired movement profiles. The desired movement profiles are generated by adjusting the values of K and B to make the linear second-order system, described by Eq. (1), behave as critically damped under different distance, velocity, and load conditions. These procedures will be repeated until enough sets of motion commands are obtained for various distance, velocity, and load conditions.

By utilizing these sets of motion commands as references, we can implement the fuzzy system. We assign 99 rules for each of h_1 , h_2 , t_1 , and t_2 . These rules are defined as fuzzy conditional statements of the form given in Eq. (7):

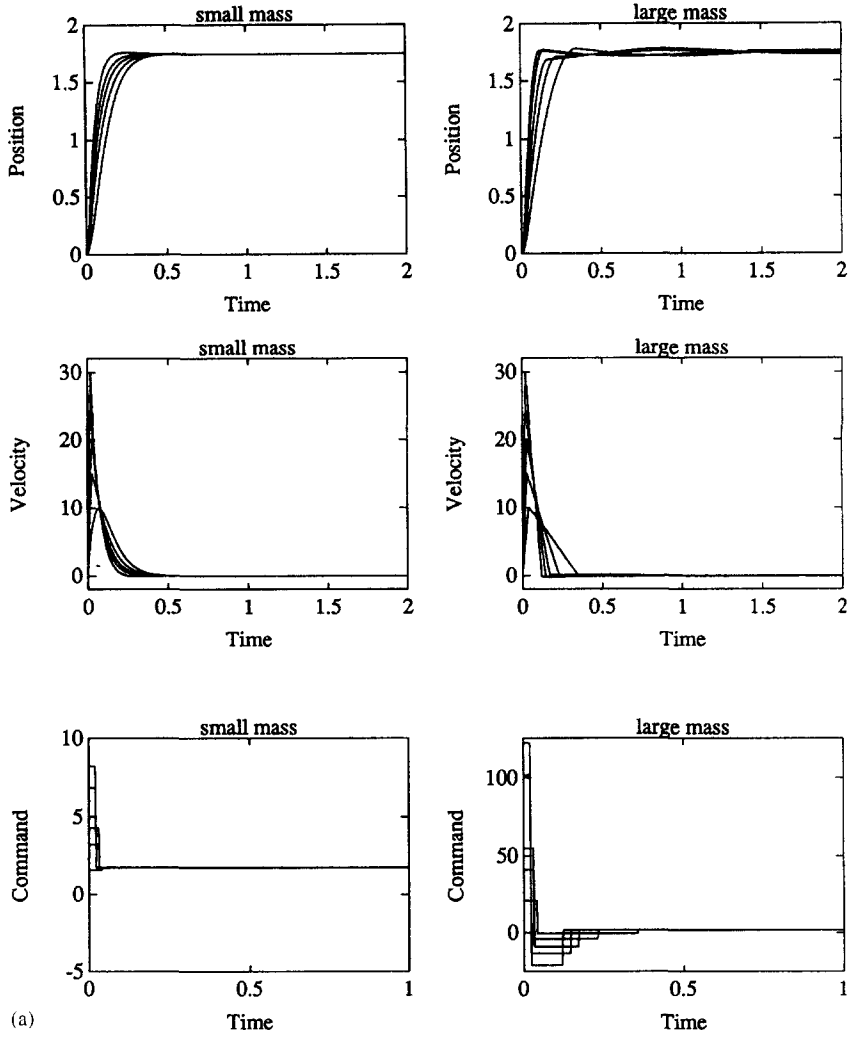


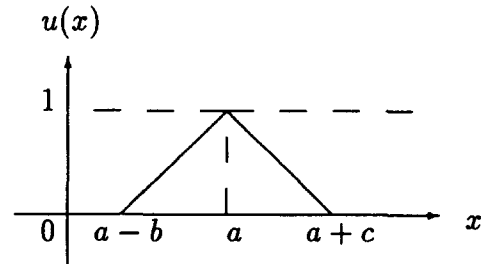
Fig. 9. Performance of the fuzzy system for a second-order system under (a) different velocity requirements and (b) different load requirements.

IF load is A_i AND velocity is B_j THEN h_1 is $C_{i,j}$,
 IF load is A_i AND velocity is B_j THEN h_2 is $D_{i,j}$,
 IF load is A_i AND velocity is B_j THEN t_1 is $E_{i,j}$,
 IF load is A_i AND velocity is B_j THEN t_2 is $F_{i,j}$,

(7)

where $i = 1, \dots, 9$, $j = 1, \dots, 11$, and $A_i, B_i, C_{i,j}, D_{i,j}, E_{i,j}$, and $F_{i,j}$ are fuzzy sets. In combining fuzzy rules, we adopt the commonly-used max-min composition [22].

The triangular membership function is used for the input fuzzy sets, as described below:



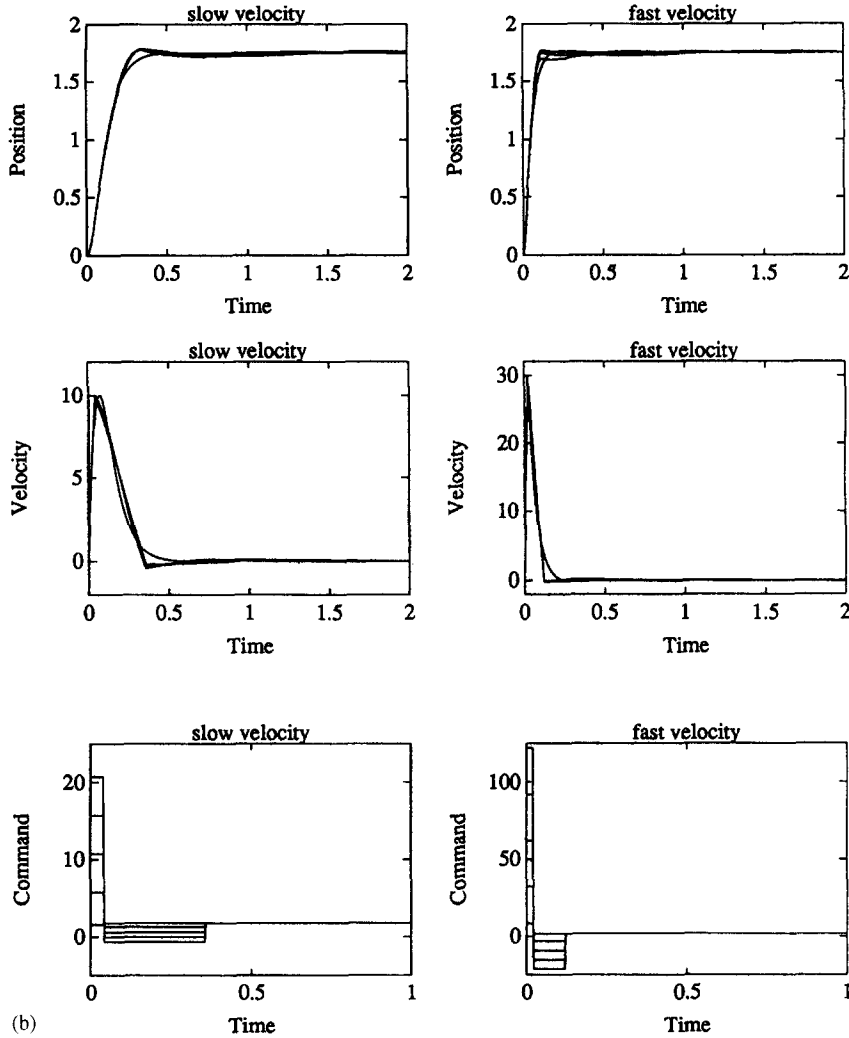


Fig. 9. Contd.

$$u(x) = \begin{cases} (b - |x - a|)/b & \text{if } a - b \leq x < a, \\ (c - |x - a|)/c & \text{if } a \leq x < a + c, \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

Different membership grades at the same crisp point can be obtained by adjusting parameters a , b , and c . The membership functions of output fuzzy sets are chosen as fuzzy singletons. A fuzzy singleton is a fuzzy set F whose support is a single point in a universal set (U) with $u_F = 1.0$ [10]. For a defuzzification strategy, we shall adopt the center of area method (COA), which generates the center of gravity of the possibility distribution of a control action. In the case

of discrete universe sets, this method yields

$$x^* = \frac{\sum_{i=1}^n u(x_i)x_i}{\sum_{i=1}^n u(x_i)}, \quad (9)$$

where n is the number of quantization levels of the outputs. The ranges for the input variables are $load \in [1, 20]$ and $velocity \in [10, 30]$ when the distance is equal to 1.75. The range of velocity will be further adjusted directly proportional to the distance, because intrinsically, the equilibrium point corresponding to a larger distance will induce a larger velocity. Fig. 8 shows the membership functions for both load and velocity. In Fig. 8, constants CV and

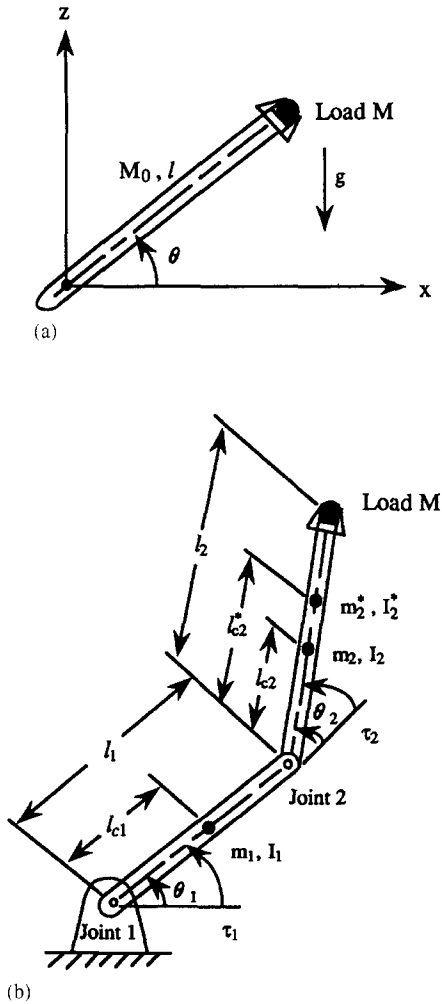


Fig. 10. (a) A single-joint manipulator and (b) a two-joint manipulator.

CM are equal to 18 and 20, respectively. These two constants are used to make the fuzzy sets correspond to the input–output ranges.

We performed a series of simulations to demonstrate the performance of the fuzzy system for a second-order system. Fig. 9(a) shows the system’s behavior in reaching the same distance under different velocity requirements. Two loads, one a small mass and one a large mass were simulated. The system performed well in both cases; the curves for the small mass are smoother. It can be seen from the figure that the heavy load involves commands with larger mag-

nitudes, as expected. Fig. 9(b) shows the system’s behavior as it reaches the same distance under different load requirements. Both slow and fast velocities were simulated, and the system again performed well in both cases. As expected, the case of fast velocity involves commands with larger magnitudes.

5. Simulations

We executed simulations for single-joint and two-joint movements. The performance of the single-joint movements can be expected to be similar to that of the fuzzy system for a second-order system in Section 4. The performance of the single-joint movements will serve as a reference for evaluating that of the two-joint movements. In two-joint movements, we dealt with both whipping and reaching movements and the related dynamic coupling.

5.1. Single-joint movements

The dynamic equation for a single-joint manipulator, as shown in Fig. 10(a), can be expressed as follows:

$$I\ddot{\theta} + B_1\dot{\theta} + \mu g \cos(\theta) = \tau, \tag{10}$$

where $I = \frac{1}{3}M_0l^2 + Ml^2$, $\mu = l(\frac{1}{2}M_0 + M)$, $M_0 = 2.815$ kg is the mass of the link, M is the load, $l = 0.3$ m is the manipulator length, B_1 is the damping coefficient, and the center of mass is located at the middle of the link. If the single link moves in a horizontal plane with input torque $\tau = -K_d\dot{\theta} - K_p(\theta - \theta_0)$ provided by the local controller, the dynamic equation in Eq. (10) can be rewritten as

$$I\ddot{\theta} + B_d\dot{\theta} + K_p(\theta - \theta_0) = 0, \tag{11}$$

where K_p and K_d are constants for the local controller, $B_d = K_d + B_1$, θ is the actual joint value, and θ_0 is the joint variable specified by the motion command. The values of B_d and K_p in Eq. (11) can be obtained via a linear scaling of the B and K used in the fuzzy system for a second-order system by comparing the inertia in Eq. (11) with the load used in the fuzzy system. The scaling of $B_d (= K_d + B_1)$ and K_p in some sense implicates the adjustment of K_d and K_p of the local controller for matching the system’s operating range. The simulation results are shown in Fig. 11. Fig. 11(a)

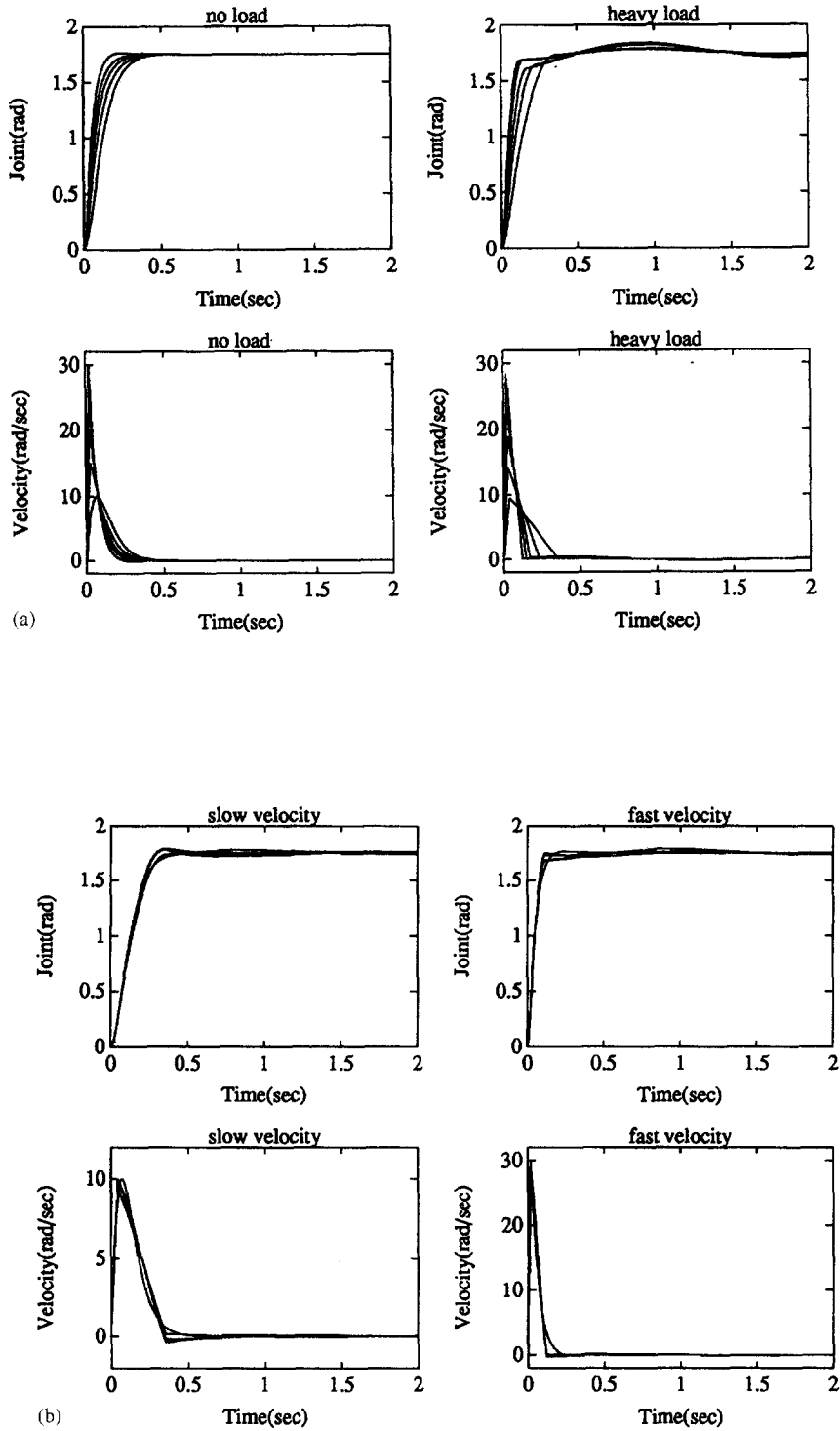


Fig. 11. Simulation results for single-joint movements under (a) different velocity requirements and (b) different load requirements

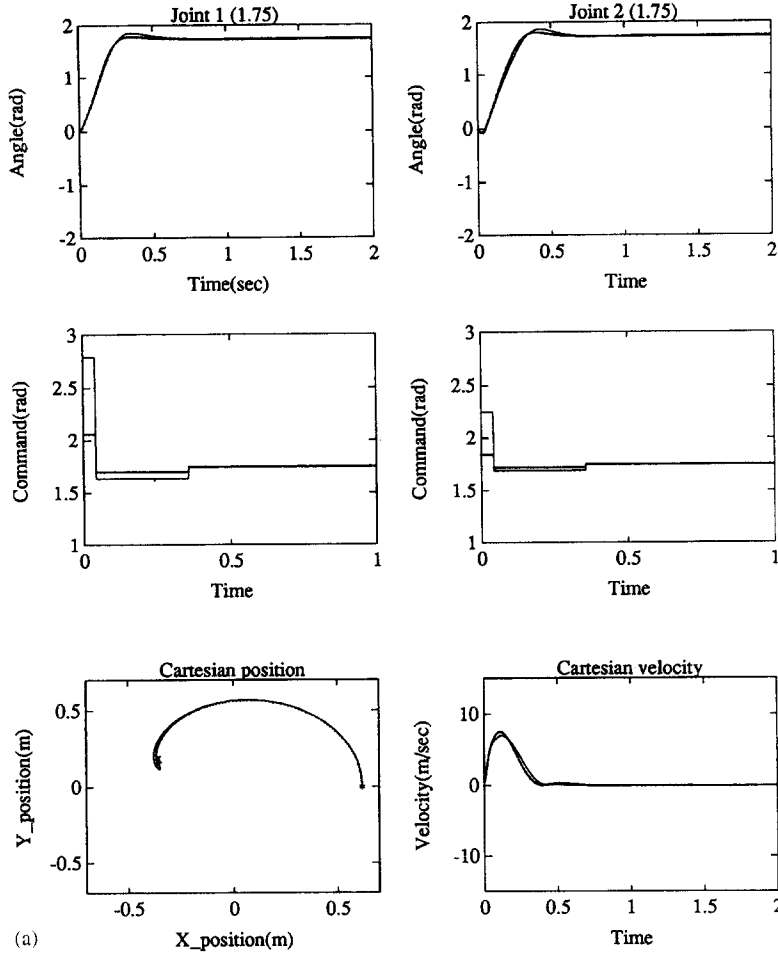


Fig. 12. Simulation results for two-joint movements: (a) whipping action and (b) reaching action.

shows the performance in reaching the same distance under different velocity requirements with no load and with a heavy load. Fig. 11(b) shows the performance in reaching the same distance under different load requirements using slow and fast velocities. The system's behavior in these simulations is similar to that shown in Fig. 9.

5.2. Two-joint movements

In the simulations for two-joint movements, first we need to map the distance and velocity from Cartesian to joint space by solving the inverse kinematics. The distance of movement and the velocity for the individual joint can then be obtained. The values of K_d and

K_p in the local controllers will also be adjusted following the procedure in the simulation for single-joint movements. However, because the inertias for the two joints are varying and dynamic coupling are present during two-joint movements, the "equivalent inertias" of the two joints used for the scaling of K_d and K_p will be in the middle range of the varying inertias plus the effect of the dynamic coupling.

The dynamic equations for a two-joint planar manipulator, as shown in Fig. 10(b), can be expressed as follows

$$\tau_1 = H_{11}\ddot{\theta}_1 + H_{12}\ddot{\theta}_2 - H\dot{\theta}_2^2 - 2H\dot{\theta}_1\dot{\theta}_2 + G_1, \quad (12)$$

$$\tau_2 = H_{21}\ddot{\theta}_1 + H_{22}\ddot{\theta}_2 + H\dot{\theta}_1^2 + G_2, \quad (13)$$

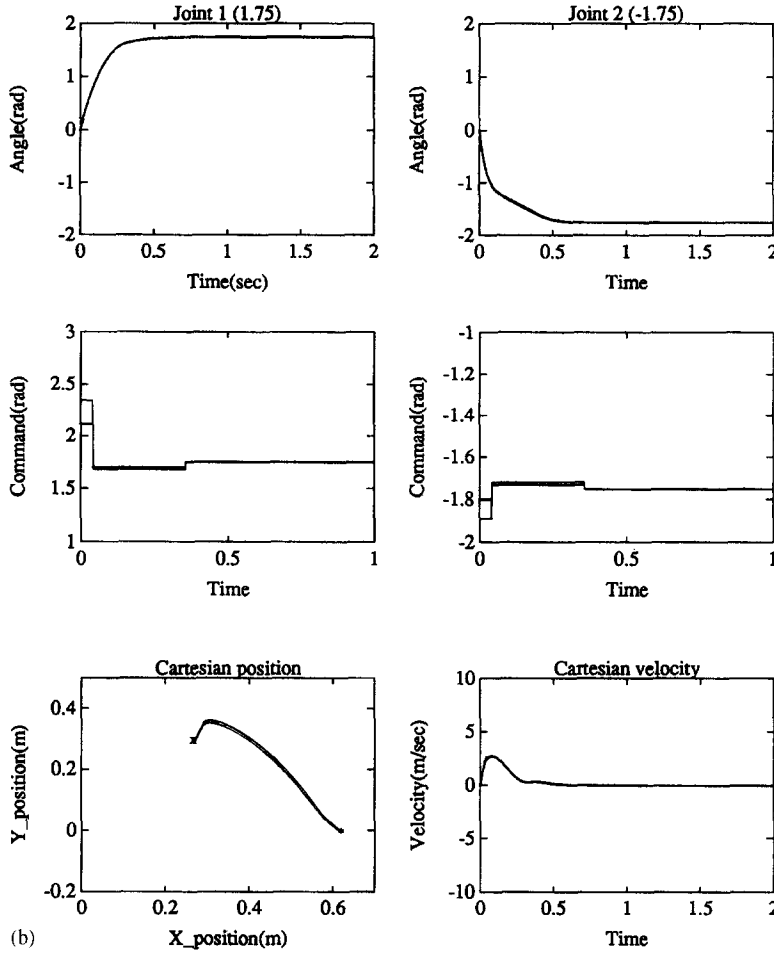


Fig. 12. Contd.

where

$$H_{11} = m_1 l_{c1}^2 + I_1 + m_2^* [l_1^{*2} + l_{c2}^{*2} + 2l_1 l_{c2}^* \cos(\theta_2)] + I_2^*, \quad (14)$$

$$H_{22} = m_2^* l_{c2}^{*2} + I_2^*, \quad (15)$$

$$H_{12} = m_2^* l_1 l_{c2}^* \cos(\theta_2) + m_2^* l_{c2}^{*2} + I_2^*, \quad (16)$$

$$H_{21} = H_{12}, \quad (17)$$

$$H = m_2^* l_1 l_{c2}^* \sin(\theta_2), \quad (18)$$

$$G_1 = m_1 l_{c1} g \cos(\theta_1) + m_2^* g [l_{c2}^* \cos(\theta_1 + \theta_2) + l_1 \cos(\theta_1)], \quad (19)$$

$$G_2 = m_2^* l_{c2}^* g \cos(\theta_1 + \theta_2) \quad (20)$$

and

$$m_2^* = m_2 + M, \quad (21)$$

$$l_{c2}^* = \frac{m_2 l_{c2} + M l_2}{m_2^*}, \quad (22)$$

$$I_2^* = I_2 + m_2 (l_{c2}^* - l_{c2})^2 + M (l_2 - l_{c2}^*)^2, \quad (23)$$

with $m_1 = 2.815$ kg, $m_2 = 1.640$ kg, $l_1 = 0.30$ m, $l_2 = 0.32$ m, $l_{c1} = 0.15$ m, $l_{c2} = 0.16$ m, and $I_1 = I_2 = 0.0234$ kg m².

The effect of gravity was ignored in the simulation, since these two links were assumed to move in a horizontal plane. In the presence of dynamic coupling, the

two-joint movements will be divided into whipping and reaching actions [6]. A whipping action involves flexion of the upper and lower links in the same direction; a reaching action involves flexion of two links in opposite directions. These two actions induce different coupling effects. We simulated whipping actions with two different loads; the corresponding traces overlap, as shown in Fig. 12(a). Since the two links rotate in the same direction, the joint velocities of the two joints have the same signs. The induced dynamic coupling can be deduced from Eqs. (12)–(13) [6]. In Fig. 12(a), the position trace in Cartesian space is a curve with a small hook near the destination. Fig. 12(b) shows the simulation results for the reaching action. In this case, the two links move in opposite directions, so the Coriolis and centripetal torques have a diminished influence compared with their influence on a whipping action. In Fig. 12(b), the reaching movement demonstrates a straighter position trace than the whipping movement. The simulation results in both cases show that the motion commands generated by the fuzzy system possess certain degree of robustness and can still yield stable and smooth Cartesian trajectories in the presence of dynamic coupling.

6. Conclusions

In this paper, we proposed an approach to simplify the learning space for robot learning control. Theoretical analyses and simulations demonstrate the feasibility of our approach. At the current stage of the study, a robot control scheme based on the proposed approach is not available for application on general industrial robot manipulators. Nevertheless, the combination of the proposed motion command pattern and the fuzzy system leads to learning space simplification and coverage, and is with a potential to achieve the goal of governing general industrial robot manipulator by using the learning mechanism as the main control module. Future study will concentrate on tackling dynamic interactions present in motions of multi-joint robot manipulators. Another worthwhile task will be to develop a more intelligent fuzzy system to approximate the reasoning process of an expert, such that the fuzzy system can deal with inputs beyond the specified range and refine itself from inputs, in

addition to the generalization of the reference motion commands [2].

References

- [1] J.J. Craig, *Introduction to Robotics* (Addison-Wesley, Reading, MA, 1989).
- [2] S. Dutta and P.P. Bonissone, Integrating case- and rule-based reasoning, *Internat. J. Approx. Reasoning* **8**(3) (1993) 163–203.
- [3] T. Flash, The control of hand equilibrium trajectories in multi-joint arm movements, *Biol. Cybernet.* **57** (1987) 257–274.
- [4] C.C.A.M. Gielen and J.C. Houk, A model of the motor servo: incorporating nonlinear spindle receptor and muscle mechanical properties, *Biol. Cybernet.* **57** (1987) 217–231.
- [5] G.L. Gottlieb, D.M. Corcos and G.C. Agarwal, Organizing principles for single-joint movements I. A speed-insensitive strategy, *J. Neurophysiol.* **62**(2) (1989) 342–357.
- [6] J.M. Hollerback and T. Flash, Dynamic interactions between limb segments during planar arm movement, *Biol. Cybernet.* **44** (1982) 67–77.
- [7] J.C. Houk and W.Z. Rymer, Neural control of muscle length and tension, in: *Handbook of Physiology – The Nervous System II*, Vol. II (American Physiol. Soc., Bethesda, MD, 1981) Sec. 1, Ch. 8, 257–323.
- [8] J.C. Houk, C.H. Wu, K.Y. Young and L.E. Miller, Nonlinear damping of limb motion, in: Winters and Woo, Eds., *Multiple Muscle Systems* (Springer, New York, 1990) 214–235.
- [9] M. Kawato, Y. Uno, M. Isobe and R. Suzuki, Hierarchical neural network model for voluntary movement with application to robotics, *IEEE Control Systems Mag.* **8**(2) (1988) 8–16.
- [10] C.C. Lee, Fuzzy logic in control systems: fuzzy logic controller – Part I, *IEEE Trans. Systems Man Cybernet.* **20**(2) (1990) 404–418.
- [11] S.L. Lehman, Input identification depends on model complexity, in: Winters and Woo, Eds., *Multiple Muscle Systems* (Springer, New York, 1990) 94–100.
- [12] C.M. Lim and T. Hiyama, Application of fuzzy logic control to a manipulator, *IEEE Trans. Robotics Automat.* **7**(5) (1991) 688–691.
- [13] W.T. Miller III, F.H. Glanz and L.G. Kraft III, Application of a general learning algorithm to the control of robotic manipulators, *Internat. J. Robotics Res.* **6**(2) (1987) 84–98.
- [14] A. Nedungadi and D.J. Wenzel, A novel approach to robot control using fuzzy logic, *IEEE Internat. Conf. Systems Man Cybernet.* (1991) 1925–1930.
- [15] T. Ozaki, T. Suzuki, T. Furuhashi, S. Okuma and Y. Uchikawa, Trajectory control of robotic manipulators using neural networks, *IEEE Trans. Industrial Electronics* **38**(3) (1991) 195–202.
- [16] A. Polit and E. Bizzi, Characteristics of motor programs underlying arm movements in monkeys, *J. Neurophysiol.* **42**(1) (1979) 183–194.

- [17] R.A. Schmidt, *Motor Control and Learning: A Behavioral Emphasis* (Human Kinetics Publishers, Champaign, IL, 1988).
- [18] B.A.M. Wakileh and K.F. Gill, Use of fuzzy logic in robotics, *Comput. in Indust.* **10** (1988) 35–46.
- [19] P.D. Wasserman, *Neural Computing – Theory and Practice* (Van Nostrand Reinhold, New York, 1989).
- [20] C.H. Wu, K.Y. Young, K.S. Hwang and S. Lehman, Voluntary movements for robotic control, *IEEE Control Systems Mag.* **12(1)** (1992) 8–14.
- [21] K.Y. Young and C.C. Fan, Control of voluntary limb movements by using a fuzzy system, *IEEE Conf. Decision and Control* (1993) 1759–1764.
- [22] H.J. Zimmermann, *Fuzzy Set Theory and Its Applications* (Kluwer Dordrecht; 3rd ed., 1996).