ELSEVIER

Fuzzy Sets and Systems 124 (2001) 155–170

FUZZY sets and systems

www.elsevier.com/locate/fss

Robot motion similarity analysis using an FNN learning mechanism [☆]

Kuu-young Young [*], Jyh-Kao Wang

Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu, 30050, Taiwan

Received 12 August 1998; received in revised form 21 March 2000; accepted 26 April 2000

Abstract

Learning controllers are usually subordinate to conventional controllers in governing multiple-joint robot motion, in spite of their ability to generalize, because learning space complexity and motion variety require them to consume excessive amount of memory when they are employed as major roles in motion governing. We propose using a fuzzy neural network (FNN) to learn and analyze robot motions so that they can be classified according to similarity. After classification, the learning controller can then be designed to govern robot motions according to their similarities without consuming excessive memory resources. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Robot learning control; Learning space complexity; Motion similarity analysis; Fuzzy neural network

1. Introduction

Learning controllers are able to tackle highly complex dynamics without explicit model dependence and identification [7,13,20,21]. In addition, they are also considered capable of generalization [1,11]. However, learning controllers are usually used as subordinates to conventional controllers in governing robot motions [10,12,14]. The conventional controller is responsible for the major portion of the control, and brings the system close to the desired state, after which the learning controller compensates for the remaining error. Some learning control schemes do however use learning controllers alone to execute motion control. But, most of them need to repeat the learning process each time a new trajectory is encountered [8]. This learning controller deficiency results mainly from the complexity of motions associated with various task requirements, e.g., different movement distances, velocities, and loads. Consequently, when a learning controller is given a major role in governing the general motion of a multiple-joint robot manipulator, the learning space it must deal with is extremely complicated [15,17].

In order to simplify the complexity of the learning space in using learning control to govern robot motions, we propose, in this paper, performing similarity analysis of robot motions using a fuzzy neural network (FNN) learning mechanism to classify robot motions according to their similarities. The FNN is basically a fuzzy system that uses a neural network structure, such that the fuzzy system parameters can be adjusted automatically [3,5,11]. During analysis, the FNN is first used to learn to govern various robot motions. The FNN may require different

[☆] This work was supported in part by the National Science Council, Taiwan, under grant NSC 87-2213-E-009-145.

[*] Corresponding author. Tel.: +886-3-5712-121; fax: +886-3-5715-998.

E-mail address: kyoung@cc.nctu.edu.tw (K. Young).

numbers of rules and shapes of membership functions to govern each specific motion. The similarities between motions are evaluated according to the numbers of rules and the shapes of the membership functions the FNN requires to govern. After classification, robot motions can then be governed by using learning controllers which are allocated memory resources according to motion similarities. In particular, groups of robot motions with high degrees of similarity will demand learning controllers with smaller memory sizes because these motions correspond to similar fuzzy parameters in the FNN. By contrast, when robot motions are randomly arranged, learning controllers with larger memory sizes will be needed to govern motions.

Application of the proposed similarity analysis on robot motion classification can therefore lead to an organized and simplified learning space for motion governing. In addition, because both the similarity analysis and motion governing use the learning mechanism, motion analysis and governing are consistent, making the classification more effective. At the current stage of the study, the proposed approach is not ready for similarity analysis on general motions of general industrial robot manipulators. Motion features that can properly represent robot motion characteristics in learning for serving as motion classification indices also remain to be found. Instead, the main focus of this paper is to demonstrate how to classify robot motions via the means of learning. The proposed motion similarity analysis is discussed in Section 2. Motion classification based on similarity analysis is presented in Section 3. In Section 4, simulations based on use of a two-joint robot manipulator are reported. Finally, discussions and conclusions are given in Section 5.

2. Proposed motion similarity analysis

Motion similarity can be defined according to different characteristics. For example, a number of arbitrary robot motions can be categorized into classes of motions with similar movement distances, velocities, or loads [22]. However, this classification cannot guarantee that motions in the same class will correspond to similar fuzzy parameters when governed using an FNN. In the proposed approach, we aim to group similar motions to simplify the complexity in the learning space. Therefore, from the standpoint of learning, in this paper, *similar motion* is defined as

Definition 1 (Similar motion). Two motions governed using an FNN are said to be similar if the numbers of fuzzy rules they require are the same, and the similarity among the shapes of their corresponding membership functions is above a pre-specified threshold.

According to Definition 1, Fig. 1 shows the conceptual organization of the proposed motion similarity analysis. An FNN is used to learn to govern the entire trajectory of an input motion. Initially, a large number of FNN linguistic labels are used in the learning. The learning process will terminate when the FNN can successfully govern the motion up to a pre-specified accuracy. During learning, redundant fuzzy rules in the FNN are eliminated, and the final FNN fuzzy rule number required and the corresponding membership function distribution for governing the motion are then determined. According to the FNN fuzzy rule numbers and the similarity between the membership functions by comparing the areas covered by the corresponding fuzzy sets, the degrees of similarity between motions are then obtained. Finally, the motions input in arbitrary fashion are classified into groups of motions according to their similarities.

For motion governing using an FNN in Fig. 1, the system in the block is not only with an FNN, but also includes a local controller connected in series with the FNN, as shown in Fig. 2 [22]. With this hierarchical structure, the complexity in motion governing is shared by the FNN and the local controller at different levels. By contrast, in previous approaches learning controllers are usually connected in parallel with conventional controllers, and used as subordinates to conventional controllers [10,12,14]. It can be seen that without the local controller in Fig. 2, the FNN would have to govern the robot manipulator directly. In other words, the control signal from the FNN is in the torque level, and thus very sensitive to fluctuations. Therefore, the system structure in Fig. 2 allows the FNN to function at a higher level, and thus generate more abstract, robust control signals [6,22]. Then, the fuzzy parameters in the FNN will be more significant for similarity evaluation. Fig. 2 shows the reference position and velocity trajectories, θ_r and $\dot{\theta}_r$, of an input

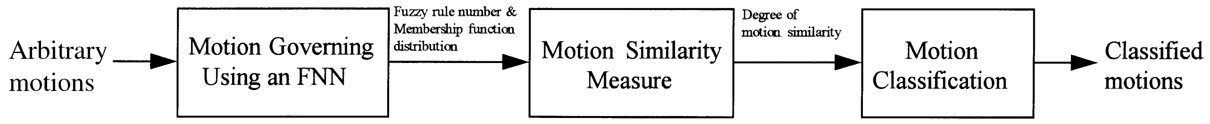

Arbitrary motions
Motion Governing Using an FNN
Fuzzy rule number & Membership function distribution
Motion Similarity Measure
Degree of motion similarity
Motion Classification
Classified motions

Fig. 1. Conceptual organization of the proposed motion similarity analysis.

motion being fed into the FNN, which in turn generates motion commands $\boldsymbol{C}_{\rm m}$ and sends them to the local controller. The local controller then modulates the motion commands via position and velocity feedbacks, θ and $\dot{\theta}$, and uses the resultant torque τ to move the robot manipulator. In our design, only the desired position is specified in the motion command. A simple position control law with linear damping is then used for the local controller [19]:

$$\tau = \boldsymbol{K}_{\rm p}(\boldsymbol{C}_{\rm m} - \theta) - \boldsymbol{K}_{\rm d}\dot{\theta}, \qquad (1)$$

where $\boldsymbol{K}_{\rm p}$ and $\boldsymbol{K}_{\rm d}$ are symmetric positive definite matrices for stability considerations [16].

The system shown in Fig. 2 is used to derive the FNN fuzzy parameters for motion similarity evaluation. Those motions evaluated to be with high degrees of similarity can then be governed by using very similar fuzzy parameters. Those with medium degrees of similarity can have their fuzzy parameters generalized to deal with a wider range of motions using a learning mechanism with a memory allocated according to the degrees of nonlinearity exhibited. In [22], we reported that a CMAC-type neural network can be used to generalize fuzzy parameters from sets of FNN fuzzy parameters appropriate for governing a number of sampled motions in a class to govern the whole class of motions. In some sense, the FNN fuzzy parameter generalization implies that qualitative fuzzy rules are generalized, and it tends to cover a larger learning space. And those with low degrees of similarity may demand learning mechanisms with larger memory sizes for generalization. Learning controllers can thus be designed and allocated appropriate memory sizes to govern the classified robot motions with differing degrees of similarity.

3. Motion classification based on similarity

Fig. 3 shows the block diagram of the proposed motion classification scheme in which the FNN learning mechanism discussed in Section 3.1 is used in two learning processes involving motion classification. The first learning process is intended to eliminate redundant linguistic labels for each input motion, which will make the resulting FNN structure more concise and allow evaluation of membership function distributions to be more meaningful. A second learning process using an FNN with a new structure to obtain new membership function distributions that are then used for motion similarity measurement between input motions. Both learning processes require similarity analysis, as discussed in Sections 3.2 and 3.3, respectively.

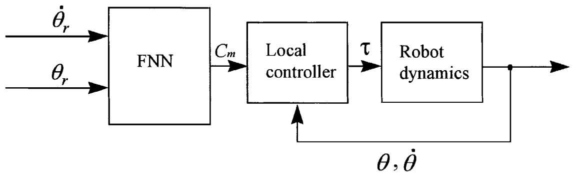


Fig. 2. Block diagram of motion governing using an FNN.

The operation of the proposed motion classification scheme is as follows. As Fig. 3 shows, in the first learning process a large number of FNN linguistic labels are initially chosen in arbitrary fashion and normal fuzzy sets are used as membership functions. The learning process terminates when the FNN can govern motion successfully; i.e., the position mean-square error (M.S.E.) is less than a prespecified value (e_1). After the input motion has been learned, the similarities between membership functions corresponding to this motion are evaluated pair by pair. When membership functions are very similar, it indicates that some of the linguistic labels are unnecessary and can be eliminated. Therefore, after the first learning process, the FNN will have a simplified structure and be sent to the second learning process. The second learning process also terminates when the FNN can govern motion successfully; i.e., when M.S.E. $< e_2$. The resulting membership function distributions for all input motions can then be used for

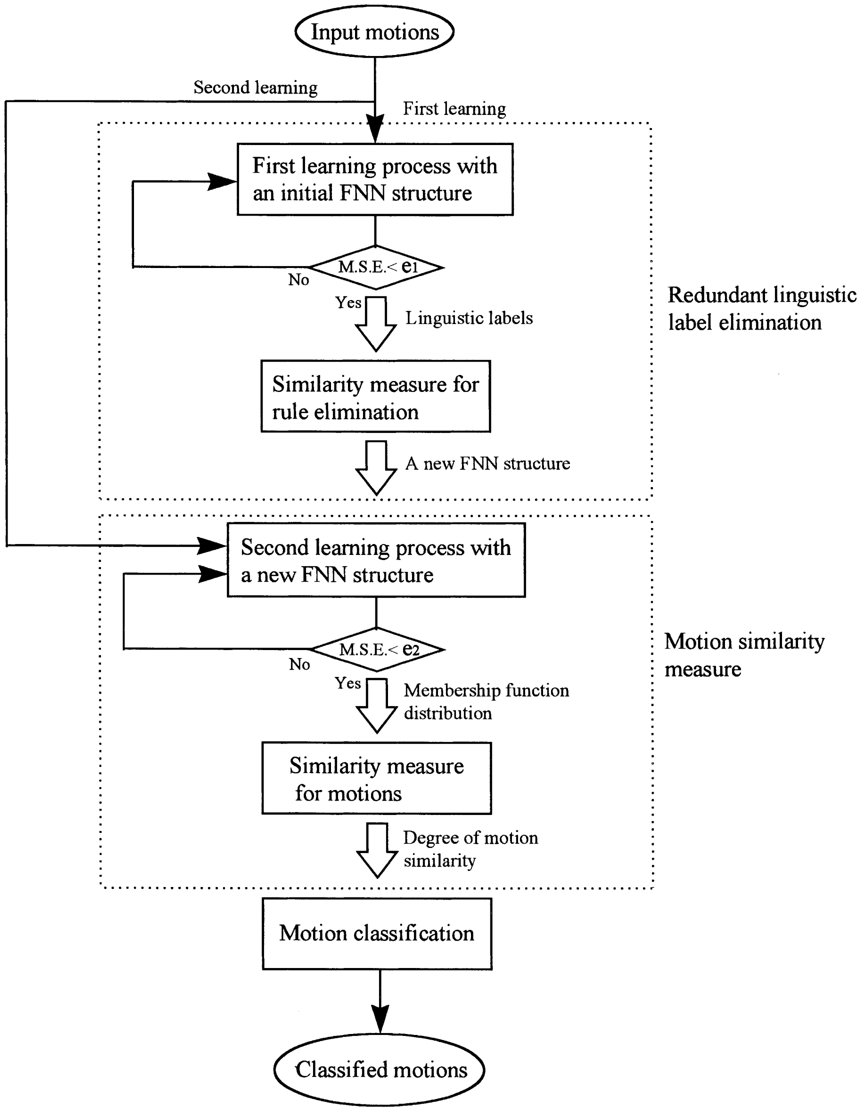

Input motions
Second learning
First learning
First learning process with an initial FNN structure
M.S.E.< e1
No
Yes
Linguistic labels
Similarity measure for rule elimination
A new FNN structure
Redundant linguistic label elimination
Second learning process with a new FNN structure
M.S.E.< e2
No
Yes
Membership function distribution
Similarity measure for motions
Degree of motion similarity
Motion similarity measure
Motion classification
Classified motions

Fig. 3. Block diagram of the proposed motion classification scheme.

motion similarity measurement. Before the similarity measurement, membership functions are first normalized to place the similarity evaluation of membership function distributions on the same scale. This is necessary because various input motions may correspond to different ranges of movement distances and velocities. Since the normalization involves only linear amplification or compression of the membership functions, their characteristics are not altered. Accordingly, input motions are classified into groups of motions with differing degrees of similarity.

3.1. The FNN learning mechanism

The FNN learning mechanism used in the proposed scheme is as shown in Fig. 4. The representation of a fuzzy system using a fuzzy neural network enables us to take advantage of the learning capability of the

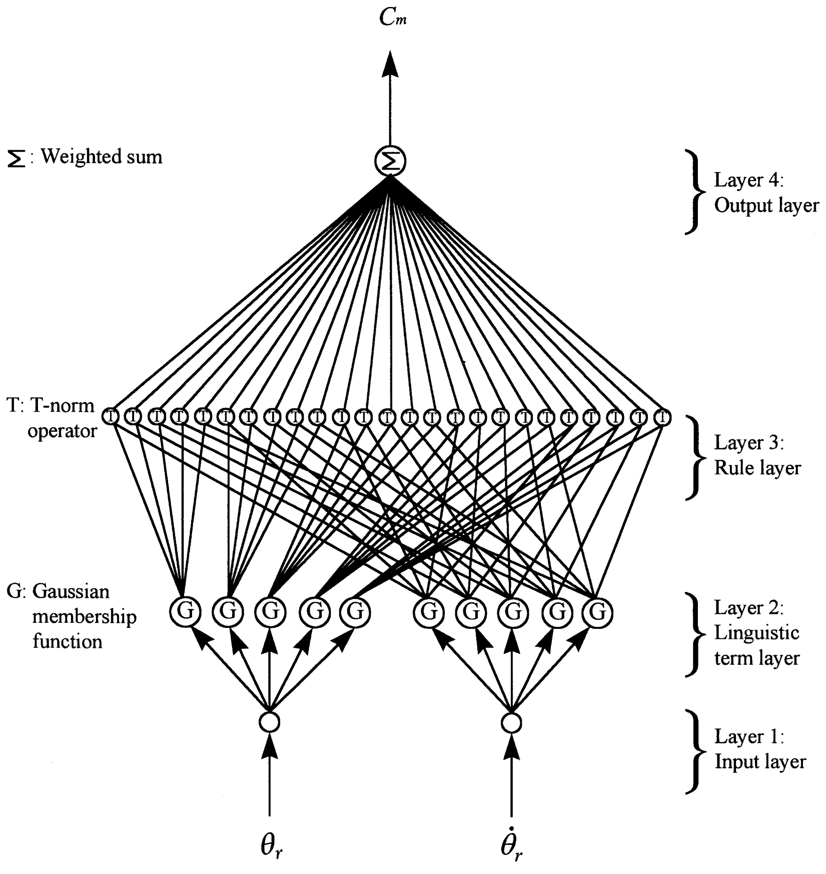
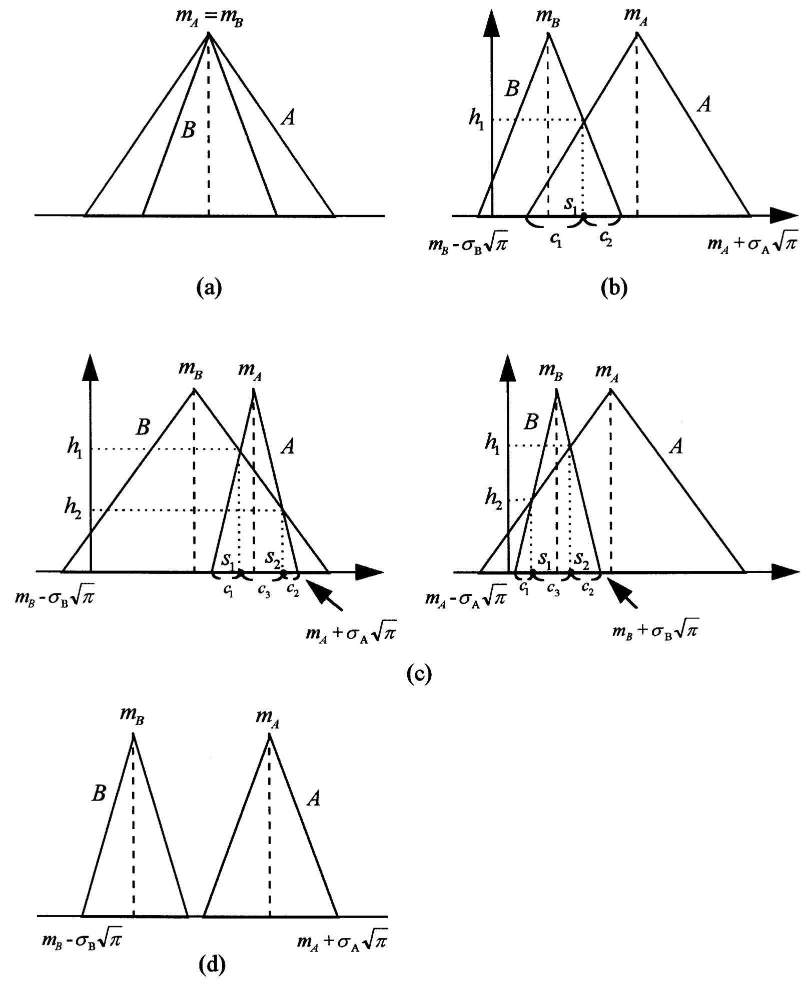


Fig. 4. The structure of the FNN.

neural network for automatic tuning of the parameters in the fuzzy system. The fuzzy reasoning parameters are thus expressed in the connection weights or node functions of the neural network [3,5,11]. In the proposed scheme, we choose an FNN with a structure similar to that in [5], of course, other types of FNN can also be used. As Fig. 4 shows, the inputs to the FNN are position and velocity trajectories of input motions, θ_r and $\dot{\theta}_r$, and the outputs are motion commands C_m. There are four layers in the FNN: the input layer, the linguistic layer, the rule layer, and the output layer. Gaussian functions with adjustable means and variances are used as membership functions. Parameters to be learned in this FNN are premise parameters in the second layer, representing the means and variances of the Gaussian functions, and consequence parameters in the fourth layer, representing the weights for the consequence links connected to the output node. A gradient-descent-based back-propagation algorithm is employed for learning [13]. More detailed discussions of the structure and learning process of this FNN can be found in Appendix A.

3.2. *Similarity measure for redundant rule elimination*

To eliminate redundant rules in using an FNN to govern a motion, the similarities between membership functions after learning are evaluated pair by pair. The method for similarity measure of fuzzy sets proposed in [4,11] was adopted for evaluation. In [4,11], similarities between fuzzy sets are computed by comparing the areas covered by fuzzy sets according to geometric points. The similarity measure between



$m_A = m_B$
B
A
(a)
m_B
m_A
B
A
h_1
s_1
$m_B - \sigma_B\sqrt{\pi}$
c_1
c_2
$m_A + \sigma_A\sqrt{\pi}$
(b)
m_B
m_A
B
A
h_1
h_2
s_1
s_2
$m_B - \sigma_B\sqrt{\pi}$
c_1 c_3 c_2
$m_A + \sigma_A\sqrt{\pi}$
$m_A - \sigma_A\sqrt{\pi}$
$m_B + \sigma_B\sqrt{\pi}$
(c)
m_B
m_A
B
A
$m_B - \sigma_B\sqrt{\pi}$
$m_A + \sigma_A\sqrt{\pi}$
(d)

Fig. 5. Similarity measure of two fuzzy sets, A and B: (a) $B \subseteq A$, (b) A and B have one intersection point, (c) A and B have two intersection points, (d) $A \cap B = \emptyset$.

two fuzzy sets A and B is thus defined as

$$E(A,B) = \frac{M(A \cap B)}{M(A \cup B)}, \tag{2}$$

where $E(A,B)$ is the degree of similarity between A and B, $\cap$ and $\cup$ denote the intersection and union operators, respectively, and $M(\cdot)$ is the size of a fuzzy set, i.e., the area it covers. Note that $0 \leqslant E(A,B) \leqslant 1$; $E(A,B) = 1$, when $A = B$, and $E(A,B) = 0$, when they do not intersect.

Because Gaussian functions are used as membership functions, the computation of the intersection and union in Eq. (2) involves two Gaussian functions with nonlinear shapes. To simplify computation, the triangular function described in Eq. (3) is used to approximate the Gaussian function:

$$\exp\left[-\frac{(x-m)^2}{\sigma^2}\right] \to \left[0, \frac{\sigma\sqrt{\pi} - |x-m|}{\sigma\sqrt{\pi}}\right], \tag{3}$$

where m and σ are the mean and variance of the Gaussian function, and m and $\sigma\sqrt{\pi}$ the center and width of the triangular function, respectively. Via the approximation, the similarity measure between A and B can be divided into four cases, as shown in Fig. 5. Detailed discussions can be found in Appendix B.

3.3. Motion similarity measure

The similarity measure described in Section 3.2 is used to eliminate redundant rules for each input motion, and is based on the area covered by two fuzzy sets. To evaluate similarities between motions, the measure is extended to deal with group of fuzzy sets corresponding to various motions. Thus, the areas covered by all membership functions that govern input motions are used. In addition, the fuzzy sets need to be normalized to place the similarity evaluation of membership function distributions on the same scale, because various input motions may correspond to different ranges of movement distances and velocities.

Assume there are two motions, m_1 and m_2, having several inputs. For a specific input i, the similarity measure between m_1 and m_2 for input i is defined as

$$E_i(m_1, m_2) = \frac{M_i(m_1 \cap m_2)}{M_i(m_1 \cup m_2)}, \tag{4}$$

where $M_i(m_1 \cap m_2)$ and $M_i(m_1 \cup m_2)$ stand for the intersection and union areas of the membership functions of m_1 and m_2 for input i. Fig. 6 shows an example of two sets of membership functions, mm_1 and mm_2, having three linguistic labels, to which two motions corresponding to one specific input belong; the shaded areas are the intersection area between mm_1 and mm_2. By considering all the inputs and performing a minimum operation (min) upon similarity measures defined in Eq. (4), the similarity measure between m_1 and m_2 can be defined as

$$E(m_1, m_2) = \min_i \{E_i(m_1, m_2)\}. \tag{5}$$

In Eq. (5), the use of the min operation guarantees that the similarity measures for all the inputs will be above certain threshold. As an example, the similarity measure between m_1 and m_2 for a two-joint robot manipulator with four inputs, θ_{r1}, θ_{r2}, $\dot{\theta}_{r1}$, and $\dot{\theta}_{r2}$, can be found using

$$E(m_1, m_2) = \min\{ E_{\theta_{r1}}(m_1, m_2), E_{\dot{\theta}_{r1}}(m_1, m_2), E_{\theta_{r2}}(m_1, m_2), E_{\dot{\theta}_{r2}}(m_1, m_2)\}. \tag{6}$$

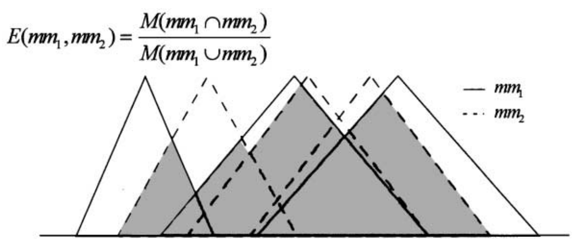


Fig. 6. Similarity measure of two groups of fuzzy sets with three linguistic labels.

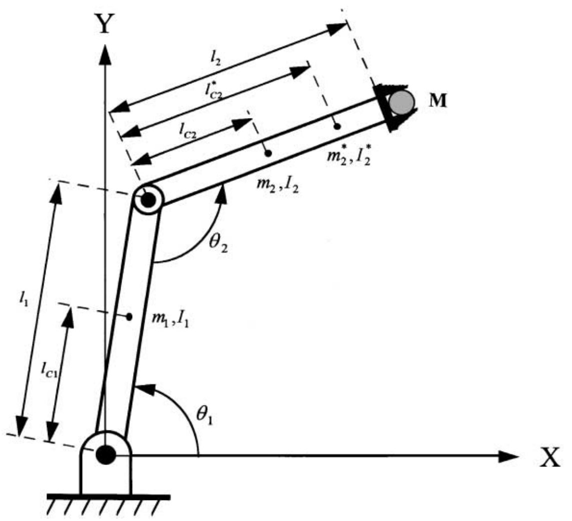


Fig. 7. A two-joint robot manipulator.

4. Simulation

Simulations were performed to demonstrate the effectiveness of the proposed motion similarity analysis based on use of the two-joint robot manipulator shown in Fig. 7. The dynamic equations for this two-joint robot manipulator are expressed as follows:

$$\tau_1 = H_{11}\ddot{\theta}_1 + H_{12}\ddot{\theta}_2 - H\dot{\theta}_2^2 - 2H\dot{\theta}_1\dot{\theta}_2 + G_1, \tag{7}$$

$$\tau_2 = H_{21}\ddot{\theta}_1 + H_{22}\ddot{\theta}_2 + H\dot{\theta}_1^2 + G_2, \tag{8}$$

where

$$H_{11} = m_1 l_{c1}^2 + I_1 + m_2^*[l_1^2 + l_{c2}^{*2} + 2l_1 l_{c2}^* \cos(\theta_2)] + I_2^*, \tag{9}$$

$$H_{22} = m_2^* l_{c2}^{*2} + I_2^*, \tag{10}$$

$$H_{12} = m_2^* l_1 l_{c2}^* \cos(\theta_2) + m_2^* l_{c2}^{*2} + I_2^*, \tag{11}$$

$$H_{21} = H_{12}, \tag{12}$$

$$H = m_2^* l_1 l_{c2}^* \sin(\theta_2), \tag{13}$$

$$G_1 = m_1 l_{c1} g \cos(\theta_1) + m_2^* g[l_{c2}^* \cos(\theta_1 + \theta_2) + l_1 \cos(\theta_1)], \tag{14}$$

$$G_2 = m_2^* l_{c2}^* g \cos(\theta_1 + \theta_2) \tag{15}$$

and

$$m_2^* = m_2 + M, \tag{16}$$

$$l_{c2}^* = \frac{m_2 l_{c2} + M l_2}{m_2^*}, \tag{17}$$

$$I_2^* = I_2 + m_2(l_{c2}^* - l_{c2})^2 + M(l_2 - l_{c2}^*)^2 \tag{18}$$

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, $I_1 = I_2 = 0.0234$ kg m^2, and M represents the mass of the load. The effect of gravity was ignored in the simulation. To provide various input motions, a second-order system was used, as described below:

$$L\ddot{\theta} + B\dot{\theta} + K(\theta - \theta_d) = 0, \tag{19}$$

where L is the load, K the stiffness, B the damping coefficient, and θ and θ_d the actual and desired joint positions for each joint, respectively. Different motions were generated by varying L, B, K, θ_d, damping ratio η, and undamped natural frequency W_n. For motion control, each joint of the robot manipulator was equipped with an FNN and a local controller. The gains of the local controller in Eq. (1) were set to $K_p = 30$ and 8 N m/rad and $K_d = 5$ and 1 N m/(rad/s) for joints one and two, respectively.

In the first simulation, applying the proposed motion classification scheme shown in Fig. 3 resulted in two sets of motions being classified as having high and low degrees of similarity, as shown in Fig. 8. Initially, the number of linguistic labels for all motions input to the FNN was set at five, and thus there were 25 fuzzy rules in the rule layer of each FNN. In the first learning process for eliminating redundant linguistic labels, the similarity threshold between membership functions was set at $E > 0.9$. After redundant rule elimination was performed on all motions in Figs. 8(a) and (b), the numbers of linguistic labels for inputs, θ_{r1}, θ_{r2}, $\dot{\theta}_{r1}$, and $\dot{\theta}_{r2}$, reduced to 5, 5, 3, and 2, respectively, for a total of 15 and 10 fuzzy rules for joints one and two, respectively. Performing the second learning operation on these motions resulted in the degrees of similarity listed in Tables 1 and 2, respectively. Table 1 shows the similarities among motions shown in Fig. 8(a) above 0.9; Table 2 shows those shown in Fig. 8(b) below 0.7. Note that if the numbers of linguistic labels for some motion inputs differed from those of other motions, the motions were taken to be dissimilar directly, and did not need the second learning phase.

In the second simulation, we intended to show that learning controllers with smaller memory allocations can be designed to govern motions with high degrees of similarity. Table 3 shows the corresponding means and variances (the premise parameters in the second

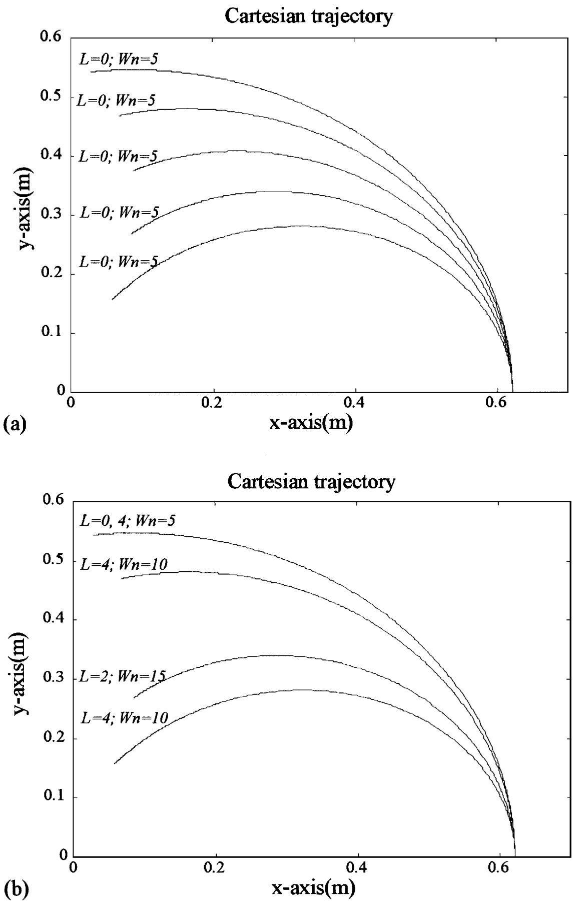


Fig. 8. (a) Motions with high degrees of similarity. (b) Motions with low degrees of similarity.

Table 1
Similarities between motions with high degrees of similarity in Fig. 8(a)

Motion number					Motion number
1	2	3	4	5	
1.0	0.963	0.966	0.943	0.921	1
	1.0	0.927	0.938	0.962	2
		1.0	0.975	0.953	3
			1.0	0.917	4
				1.0	5

Table 2
Similarities between motions with low degrees of similarity in Fig. 8(b)

Motion number					Motion number
1	2	3	4	5	
1.0	0.324	0.245	0.420	0.113	1
	1.0	0.442	0.376	0.531	2
		1.0	0.482	0.691	3
			1.0	0.287	4
				1.0	5

layer) in the FNN governing those motions with high degrees of similarity in Fig. 8(a). In Table 3, the differences between these means and variances are very small, indicating that the input membership function distributions are very similar. We thus designed a learning controller to govern the entire group of motions in Fig. 8(a), in which the same set of premise parameters, i.e., the average means and variances corresponding to the motions, were used for all the governing FNNs and the consequence parameters in the fourth layer of the FNN were generated by generalizing those consequence parameters corresponding to each separate motion. Under this design, the number of FNN parameters needed to be managed was greatly reduced, because only the consequence parameters had to be dealt with. Simulation results show that this learning controller could successfully govern those motions in Fig. 8(a). We further applied the learning controller to govern some test motions, shown in Fig. 9, that differ from the motions in Fig. 8(a) in movement distance. In Fig. 9, the generated trajectories approximate the reference ones quite well. The results demonstrate that this learning controller, with smaller memory allocation, was able to govern a group of similar motions and generalize their corresponding FNN parameters to govern other similar motions.

Finally, in the third simulation, a group of arbitrary motions with different movement distances, velocities, loads, etc., were used for classification, as shown in Fig. 10(a). These motions were divided into the six classes listed in Table 4, and are shown in Fig. 10(b): motions in classes 1–4 have similarities above 0.85, and motions in classes 5–6 have lower degrees of similarity. The results show that motions with different movement distances, velocities, and loads can be similar, and that motions with the same distances or loads may not be similar. This demonstrates that motion classification based on using an FNN learning

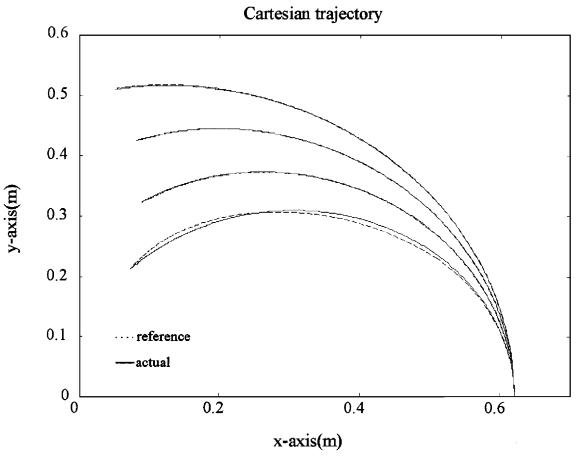


Fig. 9. Test motion governing using a learning controller with smaller memory allocation.

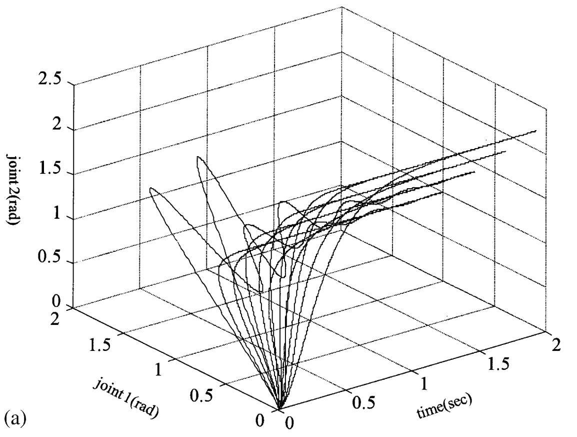
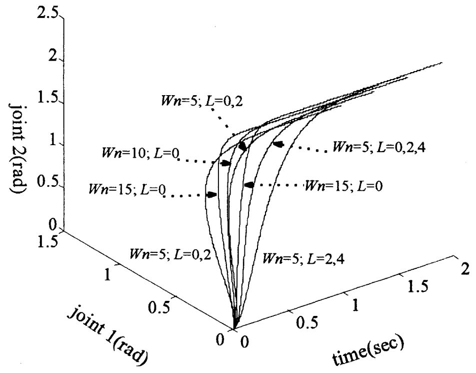
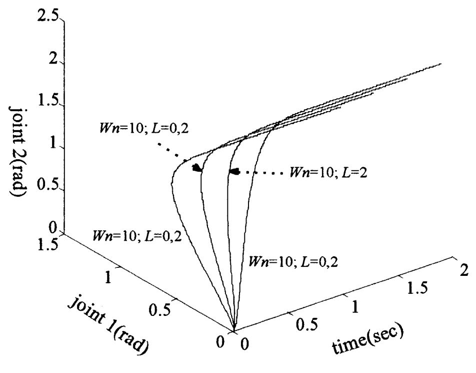


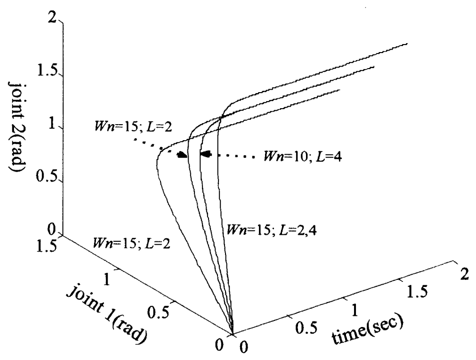
Fig. 10. (a) A group of arbitrary motions. (b) Motions after classification.



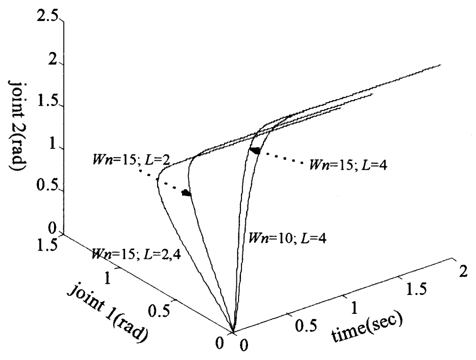
Class 1: Rule number=25, similarity>0.85.



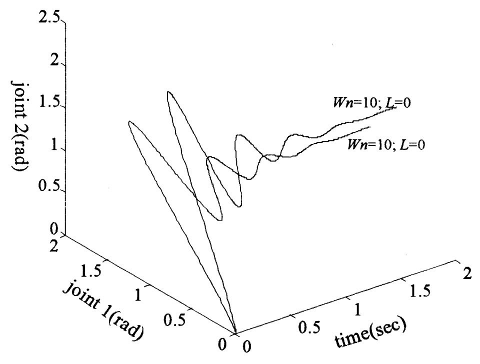
Class 2: Rule number=25, similarity>0.85.



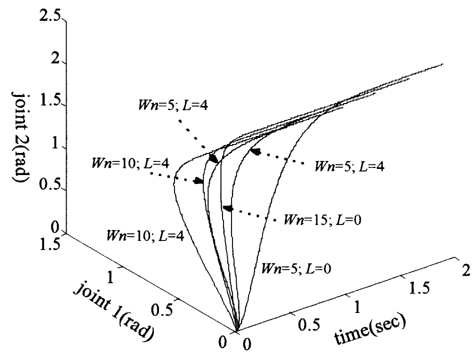
Class 3: Rule number=25, similarity>0.85.



Class 4: Rule number=25, similarity>0.85.



(b) Class 5: Rule number=27, similarity>0.85.



Class 6: Rule number=25, low similarity.

Fig. 10. Continued.

Table 3
The corresponding means and variances in the FNN for governing of motions with high degrees of similarity in Fig. 8(a)

Input θ_{d1}

Motion number	Linguistic label									
	NB		NS		Z		PS		PB	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
1	− 3.999	1.999	− 2.000	2.000	0.001	2.000	2.000	2.000	4.000	2.000
2	− 4.000	2.000	− 2.000	2.000	0.000	2.000	2.000	2.000	4.000	2.000
3	− 4.000	2.000	− 2.000	2.000	0.000	2.000	2.000	2.000	4.000	2.000
4	− 4.000	2.000	− 2.000	2.000	0.000	2.000	2.000	2.000	4.000	2.000
5	− 4.000	2.000	− 2.000	2.000	− 0.001	2.000	2.000	2.000	4.000	2.000
Average	− 3.999	1.999	− 2.000	2.000	0.000	2.000	2.000	2.000	4.000	2.000

Input θ_{d2}

Motion number	Linguistic label									
	NB		NS		Z		PS		PB	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
1	− 4.001	2.001	− 2.000	2.001	0.002	1.999	2.001	2.001	4.001	1.999
2	− 3.999	1.975	− 2.001	1.993	0.012	1.999	2.001	2.001	3.997	2.001
3	−3.998	2.001	− 2.002	2.004	− 0.002	2.001	2.001	2.001	3.989	2.003
4	− 4.000	2.001	− 1.999	2.001	−0.001	2.001	2.001	2.001	4.001	2.000
5	− 4.013	1.935	− 1.999	2.001	0.044	2.001	2.001	1.999	3.859	1.999
Average	− 4.002	1.982	− 2.000	2.000	0.011	2.000	2.001	2.001	3.969	2.000

Input $\dot{\theta}_{d1}$

Motion number	Linguistic label					
	S		Z		B	
	Mean	Variance	Mean	Variance	Mean	Variance
1	− 2.996	2.596	− 0.045	2.531	2.596	3.640
2	− 2.946	2.595	− 0.032	2.483	2.645	3.620
3	− 2.999	2.499	0.001	2.496	2.732	3.212
4	− 2.999	2.499	− 0.001	2.499	2.844	2.956
5	− 2.804	2.721	− 0.004	2.534	2.915	2.935
Average	− 2.948	2.582	− 0.016	2.508	2.746	3.272

Input $\dot{\theta}_{d2}$

Motion number	Linguistic label			
	S		B	
	Mean	Variance	Mean	Variance
1	− 1.980	2.534	1.869	2.852
2	− 1.863	2.762	1.901	3.023
3	− 2.166	2.238	1.862	2.893
4	− 1.999	2.507	1.833	2.827
5	− 1.824	2.644	1.942	2.912
Average	−1.966	2.537	1.881	2.901

Table 4
Motion index $((\theta_{1f},\theta_{2f})\ W_n\ L\ \eta)$ after classification in Fig. 10(b)

Motion number	Class																							
	1				2				3				4				5				6			
1	(1.0, 1.0)	5	0	1	(1.0, 1.0)	10	0	1	(1.0, 1.0)	15	0	1	(1.0, 1.0)	15	2	1	(1.0, 1.0)	10	0	0.2	(1.0, 1.0)	5	4	1
2	(1.0, 1.0)	5	2	1	(1.0, 1.0)	10	2	1	(0.7, 1.4)	15	0	1	(1.0, 1.0)	15	4	1	(0.7, 1.4)	10	0	0.2	(0.7, 1.4)	5	4	1
3	(0.7, 1.4)	5	0	1	(0.7, 1.4)	10	0	1	(0.7, 1.4)	15	4	1	(0.7, 1.4)	15	2	1					(0.1,2.2)	5	0	1
4	(0.7, 1.4)	5	2	1	(0.7, 1.4)	10	2	1	(0.4, 1.8)	15	2	1	(0.1,2.2)	15	4	1					(1.0, 1.0)	10	4	1
5	(0.4, 1.8)	5	0	1	(0.4, 1.8)	10	2	1	(0.4,1.8)	10	4	1	(0.1, 2.2)	10	4	1					(0.7, 1.4)	10	4	1
6	(0.4,1.8)	5	2	1	(0.1, 2.2)	10	0	1													(0.4, 1.8)	15	4	1
7	(0.4,1.8)	5	4	1	(0.1, 2.2)	10	2	1																
8	(0.1, 2.2)	5	2	1																				
9	(0.1, 2.2)	5	4	1																				
10	(0.4, 1.8)	10	0	1																				
11	(0.4, 1.8)	15	0	1																				
12	(0.1, 2.2)	15	0	1																				

mechanism does not correspond only to the kinematic or dynamic features.

5. Discussion and conclusion

This paper has proposed motion similarity analysis from the standpoint of learning. Similar motions were defined as those corresponding to the same number of fuzzy rules and similar membership function shapes when governed using an FNN learning mechanism. By classifying motions according to their similarities, learning controllers can then be designed and allocated appropriate memory sizes for motion governing. Simulations performed verified the effectiveness of the proposed approach.

At the current stage of the study, simulation and analysis are based on use of a two-joint robot manipulator. As one of our future works, the proposed approach will be used to analyze motion similarities for general industrial robot manipulators. From our preliminary simulation results, we found when those motions, originally executed by two-joint robot manipulators, were executed by six-joint robot manipulators, the fuzzy rule numbers in the rule layers of the six FNNs used for governing the six joints slightly increased from teens to still under 20. However, the time required for the FNN learning process much increased in tackling the highly complicated dynamic couplings present among joints. Because the fuzzy rule numbers and the corresponding membership function distributions are still manageable for similarity analysis and motion classification, we consider that the proposed approach is able to deal with general industrial robot manipulators at the expense of learning time.

As the gradient-descent-based back-propagation algorithm is used for learning in the FNN, the learning may fall into different local minima when the FNN learns to govern a motion. It means that the number of rules and shapes of membership functions in the FNN may be different in governing a motion, depending on the learning result. Then, a question that may be raised is how the proposed motion similarity analysis and governing will be affected under this situation. Although a motion may be categorized into motion classes with differing similarities according to different learning results, this motion still belongs to a class of motions with similar fuzzy parameters in their governing FNNs. Thus, we can still design a learning controller with smaller memory allocation to perform the subsequent motion governing.

Because the motions used for simulations in this paper were generated by step signals, an issue of interest is how the system's performance will be affected when the proposed motion classification scheme is applied to motions generated by different kinds of input signals. The simulation results show that when motions were generated by input signals in the form of steps, low-order polynomial functions, or sinusoids with different frequencies, the FNN could successfully govern the motions by using reasonable numbers of fuzzy rules, because these motions were in some sense *well-behaved*. We also let the FNN learn to guide the robot manipulator to track the trajectories of human handwriting. The results show that the FNN required more fuzzy rules and learning time in handwriting trajectory tracking than in previous cases, due to the irregularity of the trajectory contours and the corresponding complicated dynamics.

A point that also deserves discussion is about the effects of adopting different types of FNNs or local controllers for the proposed motion classification scheme. In the proposed scheme, the FNN and local controller were chosen from the current existing ones. It can be expected that when different types of FNNs or local controllers were used for similarity analysis, the resulting analysis and subsequent motion classification might be somewhat different. However, we consider which types of FNNs or local controllers to be used in the proposed scheme may not be that crucial, because our intention is to show how to apply FNNs to implement the proposed similarity analysis and how to use local controllers to make fuzzy parameters in the FNN be more meaningful for similarity evaluation.

Finally, an interesting future work will focus on how to properly classify general robot motions over the entire learning space. Simulation results in Section 4 demonstrate that motion classification via the means of learning does not correspond only to the kinematic or dynamic features, and proper features for motion classification remain to be found. Therefore, classification of general robot motions may demand further investigation into similarities among the general motions, proper motion feature selection, and concomitant reorganization of the learning space.

Appendix A: Description of the FNN

The FNN adopted for the proposed scheme consists of four-node layers, in which all nodes are of the same type within each layer [5]. Each of the four layers performs one stage of the fuzzy inference process, as described below:

Layer 1 (*Input layer*): Inputs in this layer are transmitted to the next layer directly without any computation. As Fig. 4 shows, there are two nodes for two inputs θ_r and $\dot{\theta}_\mathrm{r}$ for motions with a single degree of freedom.

Layer 2 (*Linguistic term layer*): In this layer, crisp data are transformed into fuzzy data through linguistic labelling (small, large, etc.). Each node i in this layer has the node function

$$O_i^2 = \mu_{A_i}(x), \tag{A.1}$$

where $\mu : X \to [0,1]$ is a membership function, x is the input to node i, and A_i is the linguistic label associated with this node function. The Gaussian function is chosen for $\mu_{A_i}(x)$, because it is a differentiable function suitable for use in the learning process, described as

$$\mu_{A_i}(x) = \exp\left\{-\left(\frac{x-c_i}{a_i}\right)^2\right\}, \tag{A.2}$$

where a_i and c_i represent the variance and mean of the Gaussian function, respectively, and are referred to as premise parameters. Different membership grades at the same crisp point can be obtained by adjusting the parameter set $\{a_i, c_i\}$. The Gaussian function is not the only choice as the membership function; other continuous and piecewise differential functions can also be used.

Layer 3 (*Rule layer*): This layer is intended for implementation of the fuzzy rules. Each node in this layer corresponds to a rule, and has only one antecedent link from a linguistic-term node of a linguistic label to the output node. There are $n = n_{\theta_\mathrm{r}} \times n_{\dot{\theta}_\mathrm{r}}$ rules in the FNN, where n_{θ_r} and $n_{\dot{\theta}_\mathrm{r}}$ are the numbers of linguistic labels for inputs θ_r and $\dot{\theta}_\mathrm{r}$, respectively, and n the total number of fuzzy rules. In this layer, each node outputs the firing strength of the rule, O_i^3, by using the T-norm operator to perform the fuzzy AND process [18]:

$$O_i^3 = \frac{(x_1^\gamma + x_2^\gamma)^{1/\gamma}}{2}, \tag{A.3}$$

where x_1 and x_2 are the outputs from the linguistic term layer and $\gamma \geqslant 1$. Note that there is no weight to be adjusted in this layer.

Layer 4 (*Output layer*): This layer is the output layer, and has as many nodes as there are output action variables. In Fig. 4, only one node is needed for a single motion command C_m. All consequence links are fully connected to the output node, and each connected link has its own weight w_i, referred to as the consequence parameter. The defuzzification approach adopted is the centroid defuzzification method [9]:

$$O^4 = \sum_{i=1}^{n} w_i O_i^3. \tag{A.4}$$

The parameters to be learned in this FNN include the premise parameters $\{a_i, c_i\}$, representing the means and variances of the Gaussian function, and consequence parameters $\{w_i\}$, representing the weights of the consequence links. A gradient-descent-based back-propagation algorithm is employed to learn these parameters [13]. For the determination of the parameters via the learning process for generating motion commands C_m corresponding to input motions, an error rate is first specified in the last layer (Layer 4). This error rate is then back-propagated to adjust the parameters sequentially from layer to layer. Because a concise form of the inverse dynamic model of the robot manipulator is not available, the error rate cannot be obtained directly by differentiating the error between the desired motion and the actual motion relative to the motion command C_m. Instead, we use the combined feedback error of position (e) and velocity ($\dot{e}$) between the desired and actual motions, denoted as $E = G_\mathrm{p}e + G_\mathrm{d}\dot{e}$, to derive the error rate $\partial E/\partial C_\mathrm{m}$ [2,8]:

$$\begin{aligned}\frac{\partial E}{\partial C_\mathrm{m}} &= \frac{\partial E}{\partial O^4}\\ &= \mu(G_\mathrm{p}e + G_\mathrm{d}\dot{e}),\end{aligned} \tag{A.5}$$

where μ is a learning rate and G_p and G_d are gains. The error rate $\partial E/\partial C_\mathrm{m}$ in Eq. (A.5) is estimated, but not exact, for describing the differential relationship between the motion command C_m and the resultant motion. Nevertheless, the results in [2,8] and also ours show that the use of this error rate is appropriate for the learning. Using the error rate $\partial E/\partial C_\mathrm{m}$ and some straightforward manipulation, we can derive updates for the parameters a_i, c_i, and w_i.

Appendix B: Similarity measure of two fuzzy sets

Suppose that A and B are two fuzzy sets with corresponding centers, m_A and m_B, and widths, σ_A and σ_B. The similarity measure between A and B can be divided into four cases, as shown in Fig. 5. In Cases 2–4, it is assumed that $m_A > m_B$; if $m_A < m_B$, then m_A and m_B, and σ_A and σ_B are switched.

Case 1: $m_A = m_B$ and $\sigma_A \geqslant \sigma_B$. In this case, A and B have the same center and no intersection point, as shown in Fig. 5(a). Using Eq. (2), the similarity measure can be derived from

$$\begin{aligned}E(A,B) &= \frac{M(B)}{M(A)}\\ &= \frac{\sigma_B}{\sigma_A},\end{aligned} \tag{B.1}$$

where

$$M(A \cap B) = M(B), \tag{B.2}$$

$$M(A \cup B) = M(A) + M(B) - M(A \cap B) = M(A). \tag{B.3}$$

From Eq. (B.1), the degree of similarity between A and B is equal to the ratio of σ_B to σ_A. In particular, if $\sigma_A = \sigma_B$, then $E(A,B) = 1$; i.e., $A = B$.

Case 2: $|\sigma_A - \sigma_B|\sqrt{\pi} \leqslant m_A - m_B \leqslant (\sigma_A + \sigma_B)\sqrt{\pi}$ and $m_A > m_B$. In this case, A and B have an intersection point at (s_1, h_1), as shown in Fig. 5(b). From Fig. 5(b), we can obtain

$$M(A) + M(B) = (\sigma_A + \sigma_B)\sqrt{\pi}, \tag{B.4}$$

$$M(A \cap B) = \tfrac{1}{2}(c_1 + c_2)h_1, \tag{B.5}$$

where

$$c_1 = \frac{\sigma_A(m_B - m_A) + \sigma_A(\sigma_A + \sigma_B)\sqrt{\pi}}{\sigma_A + \sigma_B}, \tag{B.6}$$

$$c_2 = \frac{\sigma_B(m_B - m_A) + \sigma_B(\sigma_A + \sigma_B)\sqrt{\pi}}{\sigma_A + \sigma_B}, \quad (B.7)$$

$$h_1 = \frac{(m_B - m_A) + (\sigma_A + \sigma_B)\sqrt{\pi}}{(\sigma_A + \sigma_B)\sqrt{\pi}}. \quad (B.8)$$

From Eqs. (B.4) and (B.5), the similarity measure can be derived as

$$E(A,B) = \frac{(c_1 + c_2)h_1}{2(\sigma_A + \sigma_B)\sqrt{\pi} - (c_1 + c_2)h_1}. \quad (B.9)$$

Case 3: $m_A - m_B \leqslant |\sigma_B - \sigma_A|\sqrt{\pi}$ and $m_A > m_B$. In this case, A and B have two intersection points at (s_1, h_1) and (s_2, h_2), as shown in Fig. 5(c). There are two conditions in this case: (i) $\sigma_A \leqslant \sigma_B$ and (ii) $\sigma_A > \sigma_B$. From Fig. 5(c), we can obtain

$$M(A \cap B) = \tfrac{1}{2}[c_1 h_1 + c_2 h_2 + c_3(h_1 + h_2)]. \quad (B.10)$$

For (i) $\sigma_A \leqslant \sigma_B$:

$$c_1 = \frac{\sigma_A(m_B - m_A) + \sigma_A(\sigma_A + \sigma_B)\sqrt{\pi}}{\sigma_A + \sigma_B}, \quad (B.11)$$

$$c_2 = \frac{\sigma_A(m_B - m_A) + \sigma_A(\sigma_B - \sigma_A)\sqrt{\pi}}{\sigma_B - \sigma_A}, \quad (B.12)$$

$$c_3 = 2\sigma_A\sqrt{\pi} - (c_1 + c_2), \quad (B.13)$$

$$h_1 = \frac{(m_B - m_A) + (\sigma_A + \sigma_B)\sqrt{\pi}}{(\sigma_A + \sigma_B)\sqrt{\pi}}, \quad (B.14)$$

$$h_2 = \frac{(m_B - m_A) + (\sigma_B - \sigma_A)\sqrt{\pi}}{(\sigma_B - \sigma_A)\sqrt{\pi}}. \quad (B.15)$$

For (ii) $\sigma_A > \sigma_B$:

$$c_1 = \frac{\sigma_A(m_B - m_A) + \sigma_B(\sigma_A - \sigma_B)\sqrt{\pi}}{\sigma_B - \sigma_A}, \quad (B.16)$$

$$c_2 = \frac{\sigma_B(m_A + m_B) + \sigma_B(\sigma_B - \sigma_A)\sqrt{\pi}}{\sigma_A + \sigma_B}, \quad (B.17)$$

$$c_3 = 2\sigma_B\sqrt{\pi} - (c_1 + c_2), \quad (B.18)$$

$$h_1 = \frac{(m_B - m_A) + (\sigma_A - \sigma_B)\sqrt{\pi}}{(\sigma_A - \sigma_B)\sqrt{\pi}}, \quad (B.19)$$

$$h_2 = \frac{(m_B - m_A) + (\sigma_A + \sigma_B)\sqrt{\pi}}{(\sigma_A + \sigma_B)\sqrt{\pi}}. \quad (B.20)$$

The similarity measure can be derived as

$$E(A,B) = \frac{c_1 h_1 + c_2 h_2 + c_3(h_1 + h_2)}{2(\sigma_A + \sigma_B)\sqrt{\pi} - [c_1 h_1 + c_2 h_2 + c_3(h_1 + h_2)]}. \quad (B.21)$$

Case 4: $m_A - m_B > (\sigma_A + \sigma_B)\sqrt{\pi}$ and $m_A > m_B$. In this case, A and B have no intersection, as shown in Fig. 5(d). Thus,

$$M(A \cap B) = 0. \quad (B.22)$$

And the similarity measure is

$$E(A,B) = 0. \quad (B.23)$$

References

[1] J.S. Albus, A new approach to manipulator control: the cerebellar model articulation controller (CMAC), ASME J. Dyn. Systems Meas. Control (1975) 220–227.

[2] S. Arimoto, Robustness of learning control for robot manipulators, IEEE Internat. Conf. Robotics Automat., 1990, pp. 1528–1533.

[3] H.R. Berenji, P. Khedkar, Learning and tuning fuzzy logic controllers through reinforcements, IEEE Trans. Neural Networks 3 (5) (1992) 724–740.

[4] C.T. Chao, Y.J. Chen, C.C. Teng, Simplification of fuzzy-neural systems using similarity analysis, IEEE Trans. Systems Man Cybernet. Part B: Cybernet. 26 (2) (1996) 344–354.

[5] J.S.R. Jang, ANFIS: adaptive-network-based fuzzy inference system, IEEE Trans. Systems Man Cybernet. 23 (3) (1993) 665–685.

[6] S. Jung, T.C. Hsia, On reference trajectory modification approach for cartesian space neural network control of robot manipulators, IEEE Internat. Conf. Robotics Automat., 1995, pp. 575–580.

[7] A. Karakasoglu, S.I. Sudharsanan, Identification and decentralized adaptive control using dynamical neural networks with application to robotic manipulators, IEEE Trans. Neural Networks 4 (6) (1993) 919–930.

[8] M. Kawato, Y. Uno, M. Isobe, R. Suzuki, Hierarchical neural network model for voluntary movement with application to robotics, IEEE Control Systems Mag. 8 (2) (1988) 8–16.

[9] C.C. Lee, Fuzzy logic in control systems: fuzzy logic controller – Part I, IEEE Trans. Systems Man Cybernet. 20 (2) (1990) 404–418.

[10] C.M. Lim, T. Hiyama, Application of fuzzy logic control to a manipulator, IEEE Trans. Robotics Automat. 7 (5) (1991) 688–691.

[11] C.-T. Lin, A neural fuzzy control system with structure and parameter learning, Fuzzy Sets and Systems 70 (1995) 183–212.

[12] W.T. Miller III, F.H. Glanz, 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.

[13] W.T. Miller III, R.S. Sutton, P.J. Werbos (Eds.), Neural Networks for Control, MIT Press, Cambridge, MA, 1990.

[14] T. Ozaki, T. Suzuki, T. Furuhashi, S. Okuma, Y. Uchikawa, Trajectory control of robotic manipulators using neural networks, IEEE Trans. Ind. Electron. 38 (3) (1991) 195–202.

[15] T.D. Sanger, Neural network learning control of robot manipulators using gradually increasing task difficulty, IEEE Trans. Robotics Automat. 10 (3) (1994) 323–333.

[16] L. Sciavicco, B. Siciliano, Modeling and Control of Robot Manipulators, McGraw-Hill, Singapore, 1996.

[17] T. Shibata, T. Fukuda, Hierarchical intelligent control for robotic motion, IEEE Trans. Neural Networks 5 (5) (1994) 823–832.

[18] C.T. Sun, Rule-base structure identification in an adaptive-neural-based fuzzy inference system, IEEE Trans. Fuzzy Systems 2 (1) (1994) 64–73.

[19] M. Takegaki, S. Arimoto, A new feedback method for dynamic control of manipulators, ASME J. Dyn. Systems Meas. Control 103 (2) (1981) 119–125.

[20] B.A.M. Wakileh, K.F. Gill, Use of fuzzy logic in robotics, Comput. Ind. 10 (1988) 35–46.

[21] K.Y. Young, C.C. Fan, An approach to simplify the learning space for robot learning control, Fuzzy Sets and Systems 95 (1) (1998) 23–38.

[22] K.Y. Young, S.J. Shiah, An approach to enlarge learning space coverage for robot learning control, IEEE Trans. Fuzzy Systems 5 (4) (1997) 511–522.