Biologically Inspired Motion Modeling and Neural Control for Robot Learning From Demonstrations

Chenguang Yang, *Senior Member, IEEE*, Chuize Chen, Ning Wang, *Member, IEEE*, Zhaojie Ju, *Senior Member, IEEE*, Jian Fu, and Min Wang, *Member, IEEE*

***Abstract*—In this paper, we propose a biologically inspired framework for robot learning based on demonstrations. The dynamic movement primitive (DMP), which is motivated by neurobiology and human behavior, is employed to model a robotic motion that is generalizable. However, the DMP method can only be used to handle a single demonstration. To enable the robot to learn from multiple demonstrations, the DMP is combined with the Gaussian mixture model (GMM) to integrate the features of multiple demonstrations, where the conventional GMM is further replaced by the fuzzy GMM (FGMM) to improve the fitting performance. Also, a novel regression algorithm for FGMM is derived to retrieve the nonlinear term of the DMP. Additionally, a neural network-based controller is developed for the robot to track the generated motions. In this network, the cerebellar model articulation controller is employed to compensate for the unknown robot dynamics. The experiments have been performed on a Baxter robot to demonstrate the effectiveness of the proposed methods.**

***Index Terms*—Cerebellar model articulation controller (CMAC), dynamic movement primitive (DMP), fuzzy Gaussian mixture model (FGMM), Gaussian mixture regression (GMR), neural control, robot learning from demonstrations (LfDs).**

I. Introduction

In THE recent decades, robots have been widely applied in both industrial manufacturing and the daily life of individuals. For industrial robots, learning from demonstrations (LfDs) [1] is one of the most efficient and dominant ways to acquire skills that can be directly used in manufacturing. This way is also applicable and even more important for robots operating in the context of daily life, such as the humanoid robot assistants, the robotic prostheses and the robotic exoskeletons [2]. For humanoid robot assistants, human-like actions that are imitated from the demonstrator make them more friendly to users [3], and the complexity of the global motion planning can be reduced through LfD. For prosthetic manipulators, the control methods usually require continuously updated commands sent from the human; for example, the signals collected from the motor cortex [4] in the brain computer interface, which heighten the load of the operator. Employing the visual evoked potentials to generate motion commands can reduce this load partly [5]. And to further simplify the control, storing the motions learned from the demonstrations would be an alternative method. However, the flexibility of the robot will be limited if the motions are preplanned. Hence, it is necessary to develop an effective model for LfD to generalize motions.

The dynamical system (DS) is a powerful tool to model the state evolution of a system. It can be employed to represent a set of trajectories that have the same attractor, which serves as the target position of the motion. Then the target of the motion can be adjusted automatically using the object recognition technologies, such as the multimodal perception [6], [7]. The neural networks (NNs), for instance, extreme learning machine, have been utilized to learn the DS-based model [8]–[11]. However, the usage of the NNs complicates the internal structure of the DS and makes the motion learning inefficient.

The dynamic movement primitive (DMP) [12], [13] offers a compact implementation of the motion model using DS. The concept of motor primitives that human behavior is composed of a sequence of basic actions [14], [15] is introduced in this model, enabling the robot to reproduce human-like motions. Additionally, this model can be used to learn the trajectories with uncertainties when combined with the reinforcement learning [16]. Hence in this paper, the DMP is chosen as the basis of our motion model. The generalization ability of the DMP is guaranteed by a second-order DS coupled with a nonlinear term, and the nonlinear term is the objective of model learning.

The traditional DMP model [12] can only be used to handle a single demonstration. However, multiple demonstrations are necessary because optimal motion is difficult to obtain through only one-time teaching, even for an expert [17]. Additionally, more information can be extracted from multiple demonstrations, such as the order of the subtasks [18] and the

Manuscript received February 28, 2018; revised June 5, 2018; accepted August 11, 2018. Date of publication August 21, 2018; date of current version June 10, 2019. This work was supported in part by the National Nature Science Foundation under Grant 61473120, Grant 61773169, and Grant 61811530281, in part by the Science and Technology Planning Project of Guangzhou under Grant 201607010006, in part by the State Key Laboratory of Robotics and System under Grant SKLRS-2017-KF-13, and in part by the Fundamental Research Funds for the Central Universities under Grant 2017ZD057. *(Corresponding author: Chenguang Yang.)*

C. Yang, C. Chen, and M. Wang are with the Key Laboratory of Autonomous Systems and Networked Control, College of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China (e-mail: cyang@ieee.org; aucchen@outlook.com; auwangmin@scut.edu.cn).

N. Wang is with the Centre for Robotics and Neural Systems, University of Plymouth, Plymouth PL4 8AA, U.K. (e-mail: ning.wang@plymouth.ac.uk).

Z. Ju is with the School of Computing, University of Portsmouth, Portsmouth PO1 2DJ, U.K. (e-mail: zhaojie.ju@port.ac.uk).

J. Fu is with the School of Automation, Wuhan University of Technology, Wuhan 430070, China (e-mail: fujian@whut.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCDS.2018.2866477

features that store the variation of the human motions [19]. To cognize more features, the data captured from multiple demonstrations should be integrated into the nonlinear term of the DMP model. The probabilistic methods have shown their feasibility to tackle this problem [20]–[23]. For example, the Gaussian mixture regression (GMR) [24], which is based on a probabilistic model named the Gaussian mixture model (GMM), has been employed to extract the important features of the task [23]. The GMR has also been utilized to construct the DS model called the stable estimator of DSs for stable motions [20]. Inspired by these works and further considering the fitting performance of the GMM, the fuzzy GMM (FGMM) [25] is employed to fuse the features of multiple demonstrations into the nonlinear term of the DMP in this paper, which has been proposed to improve the learning efficiency of the active curve axis GMM [26] and has shown better nonlinearity fitting performance than the conventional GMM. A novel regression algorithm for the FGMM is further developed to retrieve the nonlinear term, according to the geometric significance of the GMR.

In practice, only considering the motion modeling is not sufficient for a stable LfD framework because of the dynamic and unstructured environments, which will result in many disturbances and the variation of the robot's dynamics. If the manipulator is controlled using a model-based control method [27], the situation mentioned above will affect the control performance and even make the system unstable. Considering the uncertainties of the robot dynamics, various approximation tools, such as NNs [28]–[30] and fuzzy logic systems [31] have been integrated into the control design to approximate the uncertainties. Recently, NN has served as a promising computational tool in various fields; for example, the primal-dual neural network has been employed to solve a complicated quadratic programming problem [32].

For the dynamics controllers that employ the NNs, the learning efficiency is an important aspect that should be considered, because there is a tradeoff between the approximation accuracy and the efficiency of the NNs. The cerebellar model articulation controller (CMAC) is a type of NNs that have been adopted widely in dynamics control design [33]–[36]. The structure of the CMAC is inspired by the information processing mode of the cerebellum [35]. This NN is not fully connected to associative memory; thus, local weights are updated during each learning cycle to provide faster learning compared to fully connected NNs, without function approximation loss [34]. In this paper, we have also developed a CMAC-NN-based controller to guarantee that the generated motions can be performed accurately and steadily under the output constraint. This constraint exists commonly in real-world robotic systems, such as nonholonomic mobile robots [37]–[39], and its effect can be compensated with the help of a barrier Lyapunov function (BLF). The CMAC is employed to approximate the unknown dynamics of the robot.

The main contribution of this paper is the development of a complete robot learning framework. The control scheme is shown in Fig. 1. The DMP combined with the FGMM is used to model the demonstrations in Cartesian space. Then the generated motions are transformed into the trajectories in joint space using the inverse kinematics, and a CMAC-NN-based controller is developed to track the trajectories. Until now, most of this papers on this subject have only concentrated on the motion modeling without considering the performance of the dynamics controller; however, this paper accounts for these two aspects to develop a more complete robot LfD framework.

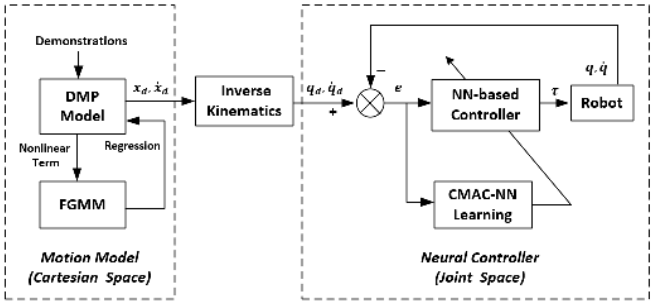


Fig. 1. Control scheme.

The remainder of this paper is organized as follows. Section II introduces the motion model and the learning method. In Section III, the NN-based controller for trajectory tracking is designed with the proof of the stability. The experiments are presented in Section IV. And Section V concludes this paper. For the convenience of the readers, the main notations used in this paper are presented in Table I.

II. Motion Modeling

A. Dynamic Movement Primitive

The DMP can be employed to represent the evolutions of various state variables in skill transfer, for example, position signals [12] and stiffness signals [40], [41]. For the robotic motion in Cartesian space, both the position and the orientation of the end-effector can be modeled with the DMP to achieve the generalization of the motion.

The DMP model for task-space motion is defined as follows [12]:

$$\begin{aligned} \tau_s \dot{v} &= d_1\big(x_g - x\big) - d_2 v - d_1\big(x_g - x_0\big)s + d_1 f(s) \\ \tau_s \dot{x} &= v \end{aligned} \tag{1}$$

where $x \in R$ denotes the position variable in Cartesian space with initial position x_0 and goal position x_g, $v \in R$ is the Cartesian velocity, $\dot{v} \in R$ is the Cartesian acceleration, $d_1, d_2 \in R$ are the positive constants to be specified, $\tau_s > 0$ is the temporal-scaling factor, and $s \in R$ is defined as the state of the following DS called the canonical system [12]:

$$\tau_s \dot{s} = -\alpha_s s \tag{2}$$

where $\alpha_s > 0$ denotes the decay rate. Usually, the initial value of s is set as $s_0 = 1$; $f(s)$ is a continuous nonlinear function defined as follows [12]:

$$f(s) = \sum_{i=1}^{n_s} \omega_{si} \psi_i(s) s \tag{3}$$

with

$$\psi_i(s) = \frac{\exp\big[-(s - b_{si})^2/(2a_{si})\big]}{\sum_{i=1}^{n_s} \exp\big[-(s - b_{si})^2/(2a_{si})\big]} \tag{4}$$

TABLE I
NOMENCLATURE

x	Cartesian position of the motion.
x_0	Initial position of the motion.
x_g	Goal position of the motion.
v	Cartesian velocity of the motion.
$\dot{v}$	Cartesian acceleration of the motion.
τ_s	Temporal-scaling factor.
s	State of the canonical system of the DMP.
$f(s)$	Nonlinear function of the DMP.
$\bar{f}^*(s)$	Expected value of $f(s)$ calculated by GMM.
$\bar{F}_R(s)$	Expected value of $f(s)$ calculated by FGMM.
$\hat{f}(s)$	Estimate of $f(s)$ calculated by LSM.
$\mathcal{O}_b$	Observed dataset generated from the DMP.
o_t	The t-th data of the dataset $\mathcal{O}_b$.
$p(\mathcal{O}_b\|\Theta)$	Probability density function of GMM/FGMM.
$p(o_t\|\Theta)$	Probability of data o_t given parameter Θ.
n_s	Number of the Gaussian functions in DMP.
n_d	Number of the demonstrations.
n_p	Number of the data o_t.
n_g	Number of the Gaussian distributions in GMM/FGMM.
n	Degrees of freedom of the robot.
n_l	Number of the layouts of the CMAC-NN.
q	Joint position of the robot.
$\dot{q}$	Joint velocity of the robot.
$\ddot{q}$	Joint acceleration of the robot.
x_d	Desired end-effector pose of the robot.
q_d	Desired joint position of the robot.
e_1	Joint position tracking error.
e_2	Joint velocity tracking error.
$\bar{e}$	Tracking errors defined as $[e_1, e_2]^T$
τ	Control torque.
τ_d	External torque caused by the disturbance.

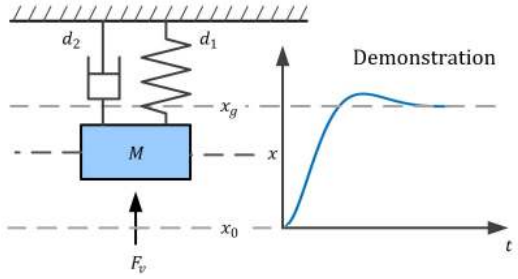


Fig. 2. Physical significance of DMP.

where $\psi_i(s)$ is the normalized Gaussian function with the mean $b_{si} \in R$ and the variance $a_{si} \in R$, n_s is the number of the Gaussian functions. $\omega_{si} \in R$ is the weight of the ith Gaussian function.

As shown in Fig. 2, the DMP model (1) can be regarded as a spring-damper system propelled by a nonlinear force, the magnitude of which is

$$F_v = -d_1\big(x_g - x_0\big)s + d_1 f(s) \tag{5}$$

where $(x_g - x_0)$ serves as the spatial-scaling factor. According to (2), s is monotonically decreasing with the initial value $s_0 > 0$ and will converge to zero. Therefore, $f(s)$ and F_v will converge to zero, and the position variable x will reach the attractor x_g, which means that the goal of the motion can be modulated by changing the value of x_g. Additionally, the duration of the motion is determined by the factor τ_s. These two characteristics are essential for a generalizable motion model.

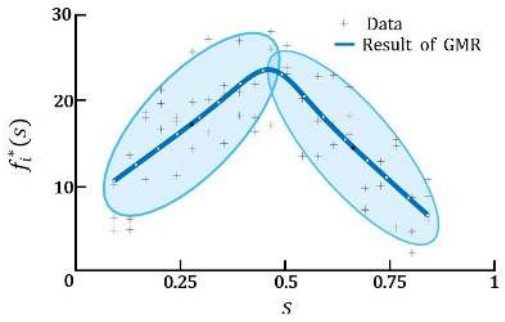


Fig. 3. GMM and regression.

B. Learning DMP Model From Multiple Demonstrations

The conventional method used for learning in a DMP model is to solve a linear regression problem, where the demonstration is assumed as the data generated from the model and the expected nonlinear function of $f(s)$ is defined as follows [12]:

$$f^*(s) = \frac{\tau_s \ddot{x}(\bigtriangledown_s) + d_2 \dot{x}(\bigtriangledown_s)}{d_1} - \big(x_g - x(\bigtriangledown_s)\big) + \big(x_g - x_0\big)s \tag{6}$$

where $x(\cdot)$ is the function of a given demonstration trajectory, $\bigtriangledown_s$ denotes the inverse function of $s(t) = s_0 \exp(-\alpha_s t/\tau_s)$, which is the solution of (2). With the data obtained from (6), the weight vector $\omega_s = \{\omega_{s1}, \ldots, \omega_{sn_s}\}$ can be estimated by using the least squares method (LSM).

When n_d demonstration trajectories $\{x_i(t)\}$ are given, multiple expected nonlinear functions, $\{f_i^*(s)\}$, for $i = 1, 2, \ldots, n_d$, can be obtained. Then the GMR, which is based on the GMM, can be employed to fuse the data obtained from these functions (see Fig. 3).

Assume that $\mathcal{O}_b = \{o_1, \ldots, o_t, \ldots, o_{n_p}\}$ with $o_t = [o_{1t}, o_{2t}] \in R^2$ is an observed dataset generated from the mapping sets $\{f_1^*, \ldots, f_{n_d}^*\}$ through discretization, where $o_{1t} \in s$, $o_{2t} \in f_i^*(o_{1t})$, and n_p is the number of the data o_t. The distribution of $\mathcal{O}_b$ is modeled by the GMM with finite Gaussian distributions, the probability density of which is [23]

$$p(\mathcal{O}_b|\Theta) = \prod_{t=1}^{n_p} p(o_t|\Theta) = \prod_{t=1}^{n_p}\left(\sum_{i=1}^{n_g} \alpha_i p(o_t|\theta_i)\right) \tag{7}$$

where $\Theta = (\alpha_1, \ldots, \alpha_{n_g}, \theta_1, \ldots, \theta_{n_g})$, $\alpha_i \in R$ is the mixing weight with $\sum_{i=1}^{n_g} \alpha_i = 1$, n_g is the number of the Gaussian distributions, and $\theta_i = (\mu_i, \sigma_i)$ is the parameter of the ith Gaussian distribution

$$p(o_t|\theta_i) = \frac{\exp\Big(-0.5(o_t - \mu_i)^T \sigma_i^{-1}(o_t - \mu_i)\Big)}{2\pi\sqrt{|\sigma_i|}} \tag{8}$$

where $\mu_i \in R^2$ is the mean and $\sigma_i \in R^{2\times 2}$ is the covariance matrix

$$\mu_i = \begin{bmatrix}\mu_{1i}\\ \mu_{2i}\end{bmatrix},\ \sigma_i = \begin{bmatrix}\sigma_{1i} & \sigma_{12i}\\ \sigma_{12i} & \sigma_{2i}\end{bmatrix}. \tag{9}$$

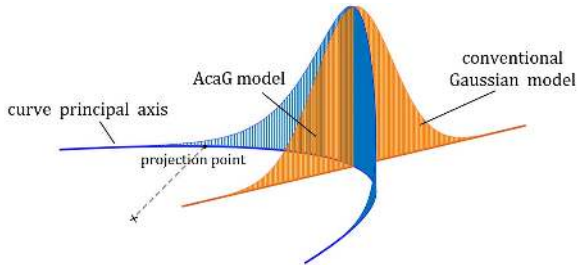


Fig. 4. Cross sections of the generalized SGM in principal plane.

The maximum likelihood estimation is employed to estimate the parameters of the GMM. Then the GMR is utilized to retrieve the composite expected function $\bar{f}^*(s)$, which is defined as [24]

$$\bar{f}^*(s) = \sum_{i=1}^{n_g} \beta_i(s)\eta_i(s) \tag{10}$$

with

$$\beta_i(s) = \frac{\alpha_i \mathcal{G}(s|\mu_{1i}, \sigma_{1i})}{\sum_{i=1}^{n_g} \alpha_i \mathcal{G}(s|\mu_{1i}, \sigma_{1i})} \tag{11}$$

$$\eta_i(s) = \mu_{2i} + \frac{\sigma_{12i}}{\sigma_{1i}}(s - \mu_{1i}) \tag{12}$$

where $\mathcal{G}(s|\mu_{1i}, \sigma_{1i})$ denotes the 1-D Gaussian distribution function with the mean μ_{1i} and the variance σ_{1i}. Using the data obtained from (10) and the LSM, the weights in (3) can be estimated.

C. Improvement to Fitting Performance Using FGMM

Considering the nonlinearity of the demonstrations, the conventional GMM is replaced by the FGMM to improve the fitting performance, which is based on the generalized single Gaussian model (SGM).

1) Generalized SGM: For a 2-D generalized SGM [26], one of its axes is beeline and the other is bent, which correspond to the conventional Gaussian model and the AcaG model, respectively, (see Fig. 4). The 2-D plane that the curve principal axis is located in is referred to as the principal plane. The observations are first transformed to the principal plane by the principal component analysis (PCA) method [26]

$$r_t = Q(o_t - T) \tag{13}$$

where the PCA is used for coordinate transformation, $T \in R^2$ is the translation vector that includes the means of the sample, $Q \in R^{2\times 2}$ is the rotation matrix which is composed of the eigenvectors of the covariance matrix, and $r_t \in R^2$ is the transformed point of o_t which is located in the principal plane. The curve principal axis is chosen as a parabola to fit the point set $\{r_t\}$ [26]

$$\bar{r}_{2t} = a_p \bar{r}_{1t}^2 + b_p \tag{14}$$

where $\bar{r}_t = [\bar{r}_{1t}, \bar{r}_{2t}]$ denotes the point in the curve principal axis, and $a_p, b_p \in R$ are computed by using the weighted least squares method.

To derive the probability density of $\mathcal{O}_b$, we first consider the AcaG model, the axis of which is located in the curve (14).

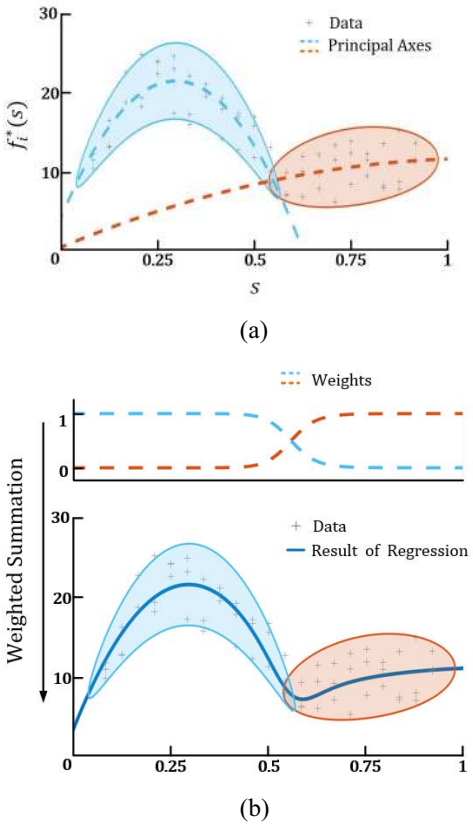


Fig. 5. Regression for FGMM. (a) FGMM and the axes of its Gaussian components. (b) Result of regression.

Assume that the projection points of r_t in the curve is $z_t = \{z_{1t}, \ldots, z_{jt}, \ldots, z_{J_t t}\}$, where J_t is the number of the projection points of r_t. In the principal plane, the center of the AcaG is $(0, 0)$; thus, the probability of the point z_{jt} is [26]

$$p_1(z_{jt}) = \frac{\exp\left(-0.5\ l_{aj}^2(r_t)\bar{\sigma}_1^{-1}\right)}{\sqrt{2\pi|\bar{\sigma}_1|}} \tag{15}$$

where $\bar{\sigma}_1 \in R$ is the variance of the AcaG model, and $l_{aj}(r_t)$ denotes the arc length between $(0, 0)$ and z_{jt}.

For the conventional Gaussian model in generalized SGM, its center is located in the projection point z_{jt}. Therefore, given z_{jt}, the probability density is [26]

$$p_2(o_t|z_{jt}) = \frac{\exp\left(-0.5\ l_{bj}^2(r_t)\bar{\sigma}_2^{-1}\right)}{\sqrt{2\pi|\bar{\sigma}_2|}} \tag{16}$$

where $\bar{\sigma}_2 \in R$ is the variance of the conventional Gaussian model, and $l_{bj}(r_t)$ denotes the distance between z_{jt} and r_t.

Then the probability distribution of the generalized SGM can be computed by

$$p(o_t|\theta) = \sum_{j=1}^{J_t} \frac{\exp\left(-0.5\ l_{aj}^2(r_t)\bar{\sigma}_1^{-1} - 0.5\ l_{bj}^2(r_t)\bar{\sigma}_2^{-1}\right)}{2\pi\sqrt{|\bar{\sigma}_1\bar{\sigma}_2|}} \tag{17}$$

where $\theta = (Q, T, a_p, b_p, \bar{\sigma}_1, \bar{\sigma}_2)$ includes the parameters of the generalized SGM.

2) Fuzzy GMM: Assume that the parameter set of the ith generalized SGM is $\theta_i = (Q_i, T_i, a_{pi}, b_{pi}, \bar{\sigma}_{1i}, \bar{\sigma}_{2i})$, and the transformed point is r_{it}, for $i = 1, 2, \ldots, n_g$. The FGMM constructs a novel mixture model by replacing (8) with (17), and introduces the fuzzy membership in the EM algorithm to effectively solve the problem of the parameter estimation. The iterative procedure of the EM algorithm for FGMM is described in [25] and [42], which improves the learning performance of the conventional method.

D. Regression for FGMM

Since the GMR employed for the learning of the DMP is based on the conventional GMM, a new regression algorithm for FGMM should be derived. The parameters of the FGMM (T_i, Q_i) involve the means of the Gaussian models in the original data space, i.e., the information of the means, which is employed in GMR, is implied in (T_i, Q_i) through the PCA transformation. Therefore, the regression algorithm for FGMM cannot be derived from (10) directly.

To derive the regression algorithm for FGMM, the geometrical significance of GMR is first discussed. The result of the GMR, (10), can be rewritten as

$$\bar{f}^*(s) = \sum_{i=1}^{n_g} \beta_i(s)(a_{ri}s + b_{ri}) \tag{18}$$

where $a_{ri} = \sigma_{12i}/\sigma_{1i}$ and $b_{ri} = \mu_{2i} - \mu_{1i}\sigma_{12i}/\sigma_{1i}$. Note that the item $(a_{ri}s + b_{ri})$ is a linear function, and its geometric representation is a beeline, where the axis of the ith Gaussian model is located. Therefore, the GMR can be regarded as the weighted summation of a set of linear functions, where the weight $\beta_i(s)$ is the normalized probability of the Gaussian model along the axis of the input.

For the FGMM, the corresponding axis, denoted by $y_{ci}(s)$, can be obtained through the PCA inverse transformation of the curve principal axis

$$\bar{c}_{it} = Q_i^{-1} c_{it} + T_i \tag{19}$$

where $c_{it} \in R^2$ and $\bar{c}_{it} \in R^2$ denote the points in the curve principal axis and in axis $y_{ci}(s)$, respectively. The weight of the point $\bar{c}_{it}$ is computed by

$$\beta_{ci}(s) = \frac{\alpha_i \sum_{j=1}^{J_{it}} \mathcal{G}\left(l_{ij(1)}(c_{it}) | 0, \bar{\sigma}_{1i}\right)}{\sum_{i=1}^{n_g} \alpha_i \sum_{j=1}^{J_{it}} \mathcal{G}\left(l_{ij(1)}(c_{it}) | 0, \bar{\sigma}_{1i}\right)}. \tag{20}$$

According to the geometrical significance discussed above, the regression for FGMM (see Fig. 5) can be written as

$$\bar{F}_R(s) = \sum_{i=1}^{n_g} \beta_{ci}(s) y_{ci}(s). \tag{21}$$

III. Adaptive CMAC-NN-Based Control

To design a joint-space controller for tracking the movement generated from the motion model, the movement is first transformed into an n-dimensional trajectory $q_d \in R^n$ in joint space using the inverse kinematics, which is bounded and smooth. In the design process of the controller, the CMAC-NN is utilized to estimate the dynamics uncertainties, such as unknown nonlinearities and varying payloads, and the BLF is employed to facilitate the control design with the output being constrained.

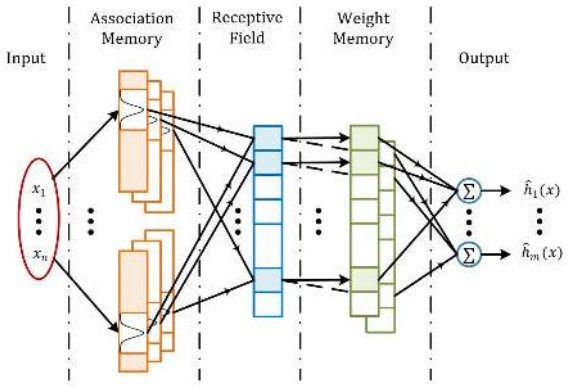


Fig. 6. Structure of CMAC.

A. Dynamics of Robot Manipulator

The dynamics of an n–link manipulator is described as follows:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau + \tau_d \tag{22}$$

where $q \in R^n$, $\dot{q} \in R^n$, and $\ddot{q} \in R^n$ denote the joint position, the joint velocity, and the joint acceleration, respectively. $M(q) \in R^{n\times n}$ is the inertia matrix which is symmetric positive definite, $C(q, \dot{q}) \in R^{n\times n}$ is the Coriolis and centripetal matrix, and $G(q) \in R^n$ is the gravity vector. $\tau \in R^n$ is the control torque and $\tau_d \in R^n$ is assumed as a bounded external torque caused by the unknown disturbance, with $\|\tau_d\| \le \bar{\tau}_d$, where $\bar{\tau}_d > 0$ is a known bound. The matrix $(\dot{M} - 2C)$ is skew-symmetric, thus we have

$$\nu^T\left(\dot{M} - 2C\right)\nu = 0, \forall \nu \in R^n. \tag{23}$$

B. CMAC Neural Networks

The CMAC-NN is an efficient functional approximator with fast learning rate. The structure of the NN is shown in Fig. 6, which consists of five spaces. Considering the estimation error, any C^1-function $H(\mathcal{X}) : R^{n_I} \to R^{n_o}$ approximated by the CMAC-NN is presented as follows [34]:

$$H(\mathcal{X}) = W^T B(\mathcal{X}) + \epsilon \tag{24}$$

where $\mathcal{X} = [\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_{n_I}]^T \in R^{n_I}$ is the input vector, $W \in R^{n_l \times n_o}$ is the weight matrix, n_l is the number of the layouts of the NN, $B(\mathcal{X}) = [B_1(\mathcal{X}), \ldots, B_k(\mathcal{X}), \ldots, B_{n_l}(\mathcal{X})]^T \in R^{n_l}$ is the receptive field function vector with

$$B_k(\mathcal{X}) = \exp\left[\sum_{i=1}^{n_I} \frac{-\left(\mathcal{X}_i - \bar{b}_{ik}\right)^2}{2\bar{a}_{ik}^2}\right] \tag{25}$$

where $\bar{b}_{ik}$ is the mean and $\bar{a}_{ik}^2$ is the variance. $\epsilon = [\epsilon_1, \ldots, \epsilon_i, \ldots, \epsilon_{n_o}]^T \in R^{n_o}$ is the approximation error with $|\epsilon_i| \le \bar{\epsilon}_i$, and $\bar{\epsilon}_i$ is a known bound.

C. Barrier Lyapunov Function

The BLF is utilized to solve the constraint problem of the manipulator controller, which is defined as follows [43]:

$$V(\xi) = \frac{k_v^2}{\pi}\tan\left(\frac{\pi\xi^2}{2k_v^2}\right) \tag{26}$$

where $\xi \in R$ denotes the state of the system, with $|\xi(0)| < k_v$, and $k_v > 0$ as a constant constraint.

Lemma 1: For a positive Lyapunov candidate function $V(t)$, which is continuously differentiable with bounded initial value $V(0)$, the solution is uniformly bounded if the following inequality holds [43]:

$$\dot{V} \le -\sigma_v V + \rho_v \tag{27}$$

where $\sigma_v > 0$ and $\rho_v > 0$.

D. Control Design

Define the tracking errors as $e_1 = [e_{11}, e_{12}, \ldots, e_{1n}]^T = q - q_d$, and $e_2 = [e_{21}, e_{22}, \ldots, e_{2n}]^T = \dot{q} - \gamma$, where γ designed later in (31) is a function of e_1 and $\dot{q}_d$. Then the closed-loop dynamics is written as follows:

$$M\dot{e}_2 + Ce_2 = \tau + \tau_d - M\dot{\gamma} - C\gamma - G. \tag{28}$$

Considering that the output is constrained by $|e_{1i}| < k_{bi}$, for $i = 1, 2, \ldots, n$, the first part of the Lyapunov candidate function $V = V_1 + V_2$ is chosen as follows [43]:

$$V_1 = \sum_{i=1}^{n}\frac{k_{bi}^2}{\pi}\tan\left(\frac{\pi e_{1i}^2}{2k_{bi}^2}\right) + \frac{1}{2}e_2^T M e_2 \tag{29}$$

where the BLF is employed. Taking the derivative of V_1 yields

$$\dot{V}_1 = \sum_{i=1}^{n}\frac{e_{1i}(e_{2i} + \gamma_i - \dot{q}_{di})}{\cos^2\left(\frac{\pi e_{1i}^2}{2k_{bi}^2}\right)} + e_2^T(\tau + \tau_d - M\dot{\gamma} - C\gamma - G). \tag{30}$$

Inspired by Lemma 1, γ is designed as

$$\gamma = -K_e\begin{bmatrix}\frac{k_{b1}^2}{e_{11}\pi}\sin\left(\frac{\pi e_{11}^2}{2k_{b1}^2}\right)\cos\left(\frac{\pi e_{11}^2}{2k_{b1}^2}\right)\\ \cdots\\ \frac{k_{bn}^2}{e_{1n}\pi}\sin\left(\frac{\pi e_{1n}^2}{2k_{bn}^2}\right)\cos\left(\frac{\pi e_{1n}^2}{2k_{bn}^2}\right)\end{bmatrix} + \dot{q}_d \tag{31}$$

where $K_e = \mathrm{diag}(k_{e1}, \ldots, k_{ei}, \ldots, k_{en})$, with $k_{ei} > 0$. Then (30) is rewritten as

$$\dot{V}_1 = -\sum_{i=1}^{n}k_{ei}\frac{k_{bi}^2}{\pi}\tan\left(\frac{\pi e_{1i}^2}{2k_{bi}^2}\right) + \sum_{i=1}^{n}\frac{e_{1i}e_{2i}}{\cos^2\left(\frac{\pi e_{1i}^2}{2k_{bi}^2}\right)} + e_2^T(\tau + \tau_d - M\dot{\gamma} - C\gamma - G). \tag{32}$$

Considering the uncertainties of the matrices M, C, and G, we further define the uncertain terms in (32) as

$$H^*(\bar{e}, q_r) = M\dot{\gamma} + C\gamma + G \tag{33}$$

where $\bar{e} = [e_1, e_2]^T$ and $q_r = [q_d, \dot{q}_d, \ddot{q}_d]^T$. For simplification, it is assumed that the trajectory is tracked perfectly, i.e., $\bar{e} = \mathbf{0} \in R^{2\times n}$. Then the function $H^*(\bar{e}, q_r)$ can be rewritten as $H(q_r)$, which results in the mismatching error being defined as

$$\varepsilon_h = H(q_r) - H^*(\bar{e}, q_r). \tag{34}$$

And according to the mean value theorem, we have [44]

$$\|\varepsilon_h(\bar{e}, q_r)\| \le g(\|\bar{e}\|)\|\bar{e}\| \tag{35}$$

where $g : R \to R$ is a strictly increasing and globally invertible function. The function $H(q_r)$ is then approximated by the CMAC-NN

$$H(q_r) = W^{*T}B(q_r) + \epsilon_h \tag{36}$$

where $W^* \in R^{n_l\times n}$ is the ideal NN weights matrix, $B(q_r) \in R^{n_l}$ is the receptive-field function vector, n_l is the number of the layouts of the CMAC-NN, and ϵ_h is the approximation error. W^* is defined as

$$W^* = \arg\min_{\hat{W}}\left(\sup_{q_r\in\Omega_d}\|H(q_r) - \hat{W}^T B(q_r)\|\right) \tag{37}$$

where $\hat{W}$ is the estimate of W^* and $\tilde{W} = W^* - \hat{W}$.

Design the control torque as

$$\tau = -\begin{bmatrix}\frac{e_{11}}{\cos^2\left(\frac{\pi e_{11}^2}{2k_{b1}^2}\right)}\\ \cdots\\ \frac{e_{1n}}{\cos^2\left(\frac{\pi e_{1n}^2}{2k_{bn}^2}\right)}\end{bmatrix} - (K_{p1} + K_{p2})e_2 - K_s\mathrm{sgn}(e_2) + \hat{W}^T B \tag{38}$$

where the first two terms guarantee the constraints satisfaction, the third term improves the robustness to disturbance, and the last term is used to compensate for the dynamics uncertainties.

Substituting (33), (34), (36), and (38) into (32), we have

$$\dot{V}_1 = -\sum_{i=1}^{n}k_{ei}\frac{k_{bi}^2}{\pi}\tan\left(\frac{\pi e_{1i}^2}{2k_{bi}^2}\right) - e_2^T(K_{p1} + K_{p2})e_2 - e_2^T\left(K_s\mathrm{sgn}(e_2) + \tilde{W}^T B + \delta - \varepsilon_h\right) \tag{39}$$

where K_{p1}, K_{p2}, and K_s are the $n\times n$ positive definite diagonal matrices, with $k_{p2} = \lambda_{\min}(K_{p2})$ and $k_s = \lambda_{\min}(K_s)$. $\delta := \epsilon_h - \tau_d$ and $\bar{\delta} := \bar{\epsilon}_h + \bar{\tau}_d$. Selecting $k_{p2} > g(\|\bar{e}\|)$, we have

$$e_2^T\varepsilon_h \le \|\bar{e}\|^2 g(\|\bar{e}\|) < \|\bar{e}\|^2 k_{p2} \le e_2^T K_{p2} e_2. \tag{40}$$

Selecting $k_s > \bar{\delta}$, we have

$$-e_2^T(K_s\mathrm{sgn}(e_2) + \delta) < 0. \tag{41}$$

The second part of the V is further chosen as

$$V_2 = \frac{1}{2}\mathrm{tr}\left(\tilde{W}^T\Gamma^{-1}\tilde{W}\right) \tag{42}$$

where Γ is the positive definite matrix. Using (40) and (41), the derivative of V is written as

$$\dot{V} = \dot{V}_1 + \dot{V}_2 \le -\sum_{i=1}^{n}k_{ei}\frac{k_{bi}^2}{\pi}\tan\left(\frac{\pi e_{1i}^2}{2k_{bi}^2}\right) - e_2^T K_{p1} e_2 - \mathrm{tr}\left[\tilde{W}^T\left(Be_2^T + \Gamma^{-1}\dot{\hat{W}}\right)\right]. \tag{43}$$

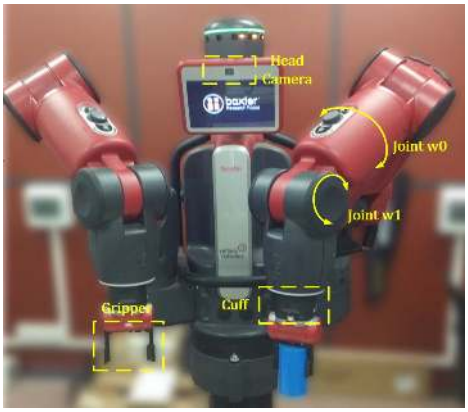
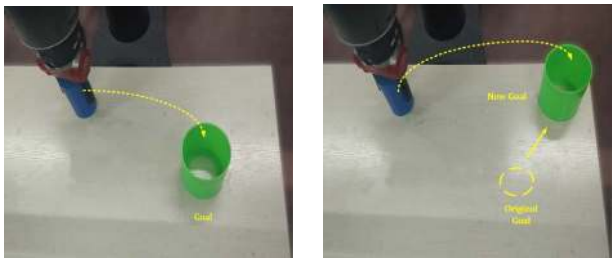


Fig. 7. Experiment platform.



(a) (b)

Fig. 8. (a) Pick-and-place task demonstrated by a human tutor. (b) New situation.

Design the update law of the NN weights as follows:

$$\dot{\hat{W}} = -\Gamma\left(Be_2^T + \kappa\hat{W}\right) \tag{44}$$

where $\kappa > 0$ is the parameter to be adjusted. According to the Young's inequality, we have: $tr(\tilde{W}^T\hat{W}) \leq (-\|\tilde{W}\|_F^2 + \|W^*\|_F^2)/2$. Then (43) is further derived as

$$\begin{aligned}\dot{V} &\leq -\sum_{i=1}^{n} k_{ei}\frac{k_{bi}^2}{\pi}\tan\left(\frac{\pi e_{1i}^2}{2k_{bi}^2}\right) - e_2^T K_{p1} e_2 \\ &\quad - \frac{1}{2}\kappa\|\tilde{W}\|_F^2 + \frac{1}{2}\kappa\|W^*\|_F^2 \\ &\leq -\rho_1 V + \rho_2\end{aligned} \tag{45}$$

where $\rho_2 = (1/2)\kappa\|W^*\|_F^2$ and ρ_1 is defined as

$$\rho_1 = \min\left(\lambda_{\min}(K_e), \frac{2\lambda_{\min}\left(K_{p1}\right)}{\lambda_{\max}(M)}, \frac{\kappa}{\lambda_{\max}\left(\Gamma^{-1}\right)}\right). \tag{46}$$

Multiplying both sides of (45) by an exponential term $e^{\rho_1 t}$, we have: $e^{\rho_1 t}\dot{V} + \rho_1 e^{\rho_1 t}V \leq e^{\rho_1 t}\rho_2$. Then the following inequality is obtained through an integral operation:

$$\frac{1}{2}e_1^T e_1 \leq V \leq e^{-\rho_1 t}V(0) + \frac{\rho_2}{\rho_1}\left(1 - e^{-\rho_1 t}\right) \leq V(0) + \frac{\rho_2}{\rho_1}. \tag{47}$$

Therefore, considering the closed-loop system including the robot dynamics (22), the control torque (38) and the NN update law (44), the tracking error e_1 will converge asymptotically to the compact set

$$\Omega_s = \left\{e_1 \middle| \|e_1\|^2 \leq 2V(0) + \frac{2\rho_2}{\rho_1}\right\} \tag{48}$$

providing that $V(0) \in L_\infty$.

IV. Experimental Verification

A. Experiment Platform

The effectiveness of the proposed method has been verified on a Baxter robot, which is a collaborative robot developed by Rethink Robotics. As shown in Fig. 7, the Baxter has two 7-DOF arms and multiple sensors; for example, the head camera that can be employed to detect objects. In our experiments, the linear electric grippers are attached to the end positions of two arms such that the robot is able to perform pick-and-place tasks. The built-in Zero-G mode of the robot is utilized for demonstrations, which allows the arms to be moved freely by the demonstrator. This mode can be activated by grasping the cuff of the robot.

B. Motion Learning and Generalization

This group of experiments aims to test the performance of the proposed DMP-based motion model. The pick-and-place task is first demonstrated five times by the demonstrator. As shown in Fig. 8(a), the task is to place the blue cylinder object into the green canister. The orientation of the gripper is fixed as: $\zeta_{rpy} = [-\pi, 0, 0]$(rad), and the movement trajectories of the end-effector along each direction are recorded during the demonstrations. The demonstration data is shown in Fig. 9(a), which is used to learn the motion models. In this group of experiments, the motion models are employed for the modeling of the trajectories along the x-axis, y-axis, and z-axis, respectively. The parameters of the models are set as: $d_1 = 25$, $d_2 = (25/2)^2$, $\tau_s = 1.0$, and $\alpha_s = 25/3$. The number of the normalized Gaussian functions is selected as 10, the centers of which are distributed evenly in the interval [0, 1]. The duration of motions is normalized as 1s in the learning phase, and the sampling rate is set as 0.01 s. The nonlinear terms are, respectively, modeled by the GMM and the FGMM to compare the performance of these two methods. The number of the Gaussians in each mixture model is selected as 2. The learned motion models are then employed to generate new motions.

Fig. 9(b) shows the fusion result that is learned with the GMM, while in Fig. 9(c), the FGMM is employed. The results show that the FGMM has better performance than the GMM when the numbers of the Gaussians are equal. At the beginning of the motion generated by the GMM, the characteristic changes obviously; thus, the GMM has poor performance when the number of the Gaussians is insufficient. We increase the number of the Gaussians in the GMM and as shown in Fig. 9 (d), the performance of the motion model is improved. We use the mean square error (MSE) to quantify the accuracy of the estimations

$$\text{MSE} = \frac{1}{n_d}\sum_{i=1}^{n_d}\|x_i - \hat{x}\|^2$$

where n_d is the number of the demonstration trajectories, x_i denotes the demonstration trajectory, and $\hat{x}$ denotes the learned

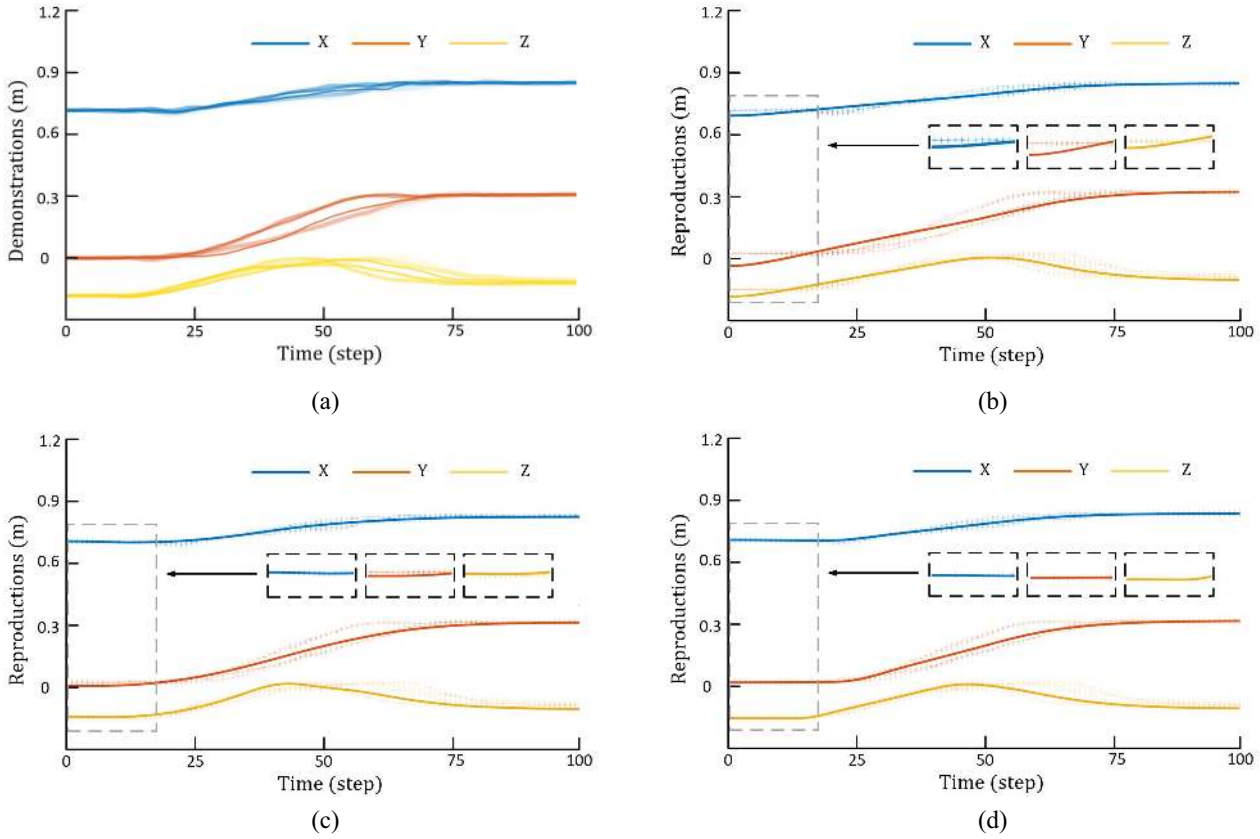


Fig. 9. Demonstrations and the reproductions generated from the motion models. (a) Demonstration trajectories along x-axis, y-axis, and z-axis. Learned motion using (b) GMM with two Gaussians, (c) FGMM with two Gaussians, and (d) GMM with three Gaussians.

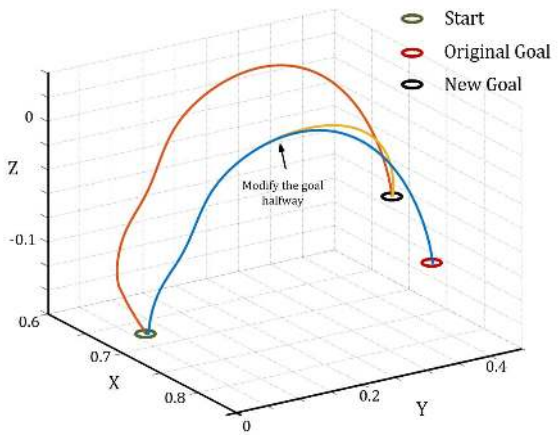


Fig. 10. Generalization.

trajectory. The results are shown in Table II, showing that the errors of the FGMM are smaller than the errors of the GMM when the numbers of the Gaussians are equal.

The generalization ability of the model is verified in the second experiment. As shown in Fig. 8(b), the canister is moved to a new place, and we take this action at the beginning of the motion and halfway. The head camera of the robot is employed to measure the position of the canister, and the coordinate of the canister is used to adjust the goal parameter x_g of the motion model. In both situations, the motions all evolve to the new goal as shown in Fig. 10, which indicates that the generalization ability of the DMP model is inherited.

TABLE II
PERFORMANCE COMPARISON OF THE PROPOSED METHODS

Method	MSE(x axis)	MSE(y axis)	MSE(z axis)
GMM(2 models)	0.1289	0.3141	0.2192
FGMM(2 models)	0.0876	0.2295	0.2003
GMM(3 models)	0.0782	0.1972	0.1783

C. Verification of the NN-Based Controller

In this group of experiments, the performance of the proposed NN-based controller is verified. Considering the learning efficiency of the NN, two joints ($w0$ and $w1$) of the robot are selected to track the given trajectories, which are defined as follows:

$$\begin{cases} \zeta_{w0} = 0.5\sin(2\pi t/3) + 1.0 \\ \zeta_{w1} = 0.5\cos(2\pi t/3) + 0.2 \text{ (rad)}. \end{cases}$$

The weights of the NN are initialized as 0, and the centers of the receptive-field functions are distributed evenly in the intervals [0.2, 1.8] and [0.0, 1.0].

In the first experiment, a controller without NN learning is used to track the given trajectories, while in the second experiment, the neural learning of the controller is enabled. The experiment results are shown in Fig. 11(a)–(f), where the

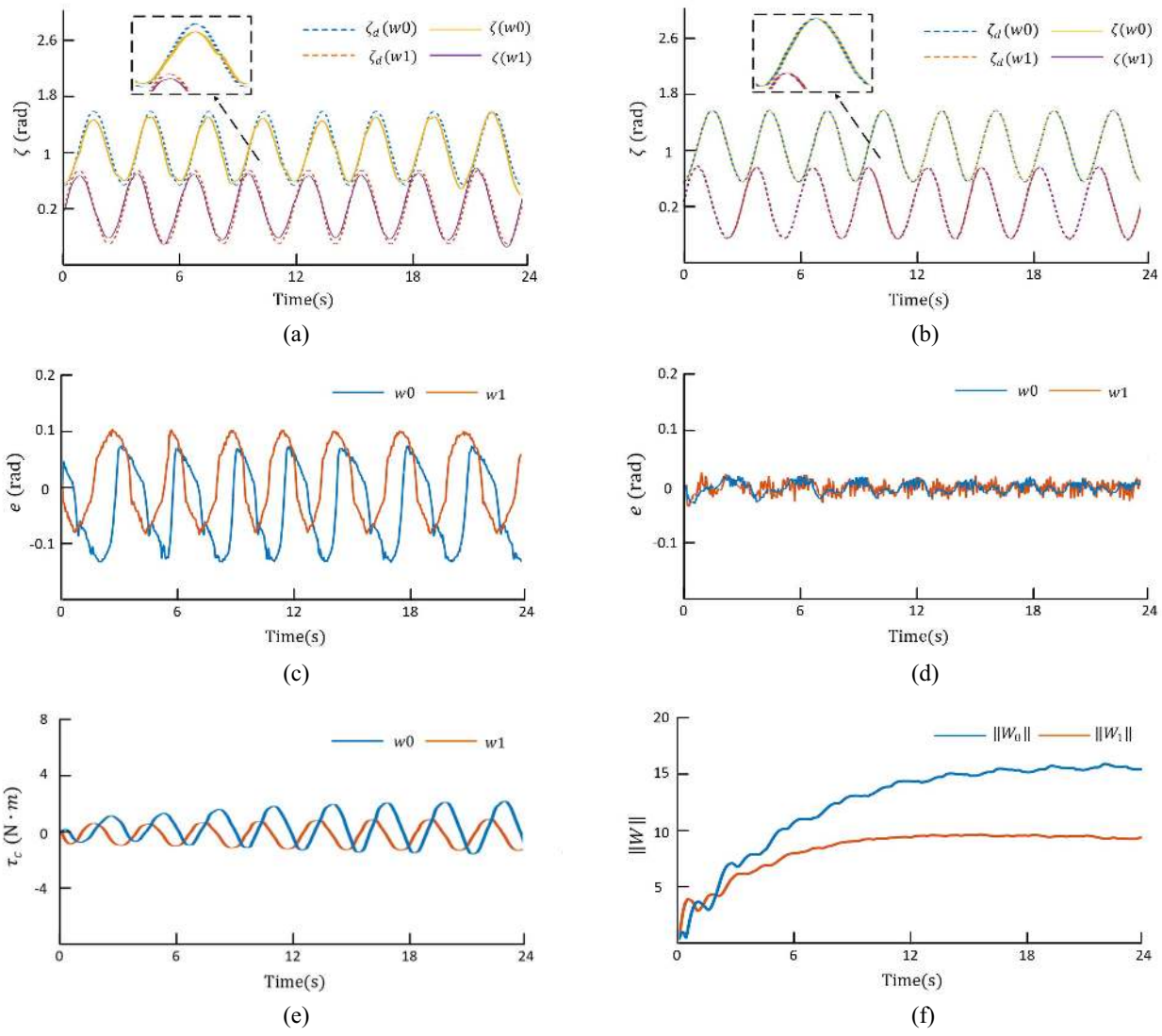


Fig. 11. Results of the comparison experiments that are used to verify the NN-based controller. Desired trajectories and the actual trajectories of the joints (a) $w0$ and $w1$ without NN learning and (b) $w0$ and $w1$ with NN learning. The tracking errors of the joints (c) $w0$ and $w1$ without neural learning and (d) $w0$ and $w1$ with neural learning. (e) Compensation torques generated by the neural network. (f) Norm of each column of the NN weight matrix W.

controller with the neural learning enables the robot to track the given trajectories more accurately. The tracking errors are reduced into the interval $[-0.035, 0.035]$ with the compensation torques generated by the neural network as is shown in Fig. 11(d) and (e), where the effect caused by the unknown dynamics and the dynamic environment is compensated for. Fig. 11(f) shows the norm of each column of the NN weight matrix. The value of each norm converges in 18 s.

V. Conclusion

This paper has proposed a novel robot LfD framework, considering the performance of both a motion model and dynamics controller. The DMP was chosen as the basic model in our method, the generalization ability of which was employed. The FGMM was utilized to fuse multiple demonstrations to cognize more features from human motions, which showed better nonlinearity fitting performance than the GMM. The regression algorithm for FGMM was also developed to replace the GMR. Besides, the CMAC-NN was integrated into the controller to cognize the dynamic environment and to compensate for the unknown dynamics, whereby the robot was able to track the trajectories generated from the motion model more accurately. The effectiveness of the proposed methods has been verified through several experiments that were performed on the Baxter robot. In future work, we will combine our framework with the reinforcement learning technology to enable the robot to learn motions through trial and error, rather than learning only from the human demonstrations.

References

[1] S. Schaal, "Learning from demonstration," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, pp. 1040–1046.

[2] Z. Li *et al.*, "Asymmetric bimanual control of dual-arm exoskeletons for human-cooperative manipulations," *IEEE Trans. Robot.*, vol. 34, no. 1, pp. 264–271, Feb. 2018.

[3] K. Dautenhahn *et al.*, "What is a robot companion—Friend, assistant or butler?" in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2005, pp. 1192–1197.

[4] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz, "Cortical control of a prosthetic arm for self-feeding," *Nature*, vol. 453, no. 7198, pp. 1098–1101, 2008.
[5] S. Zhao *et al.*, "Brain–machine interfacing-based teleoperation of multiple coordinated mobile robots," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 5161–5170, Jun. 2017.
[6] H. Liu, Y. Wu, F. Sun, B. Fang, and D. Guo, "Weakly paired multimodal fusion for object recognition," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 2, pp. 784–795, Apr. 2018.
[7] H. Liu, F. Sun, B. Fang, and S. Lu, "Multimodal measurements fusion for surface material categorization," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 2, pp. 246–256, Feb. 2018.
[8] J. Hu, Z. Yang, Z. Wang, X. Wu, and Y. Ou, "Neural learning of stable dynamical systems based on extreme learning machine," in *Proc. IEEE Int. Conf. Inf. Autom.*, 2015, pp. 306–311.
[9] C. Yang, K. Huang, H. Cheng, Y. Li, and C.-Y. Su, "Haptic identification by ELM-controlled uncertain manipulator," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 8, pp. 2398–2409, Aug. 2017.
[10] J. Duan *et al.*, "Fast and stable learning of dynamical systems based on extreme learning machine," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8002637/, doi: 10.1109/TSMC.2017.2705279.
[11] M. Wang, C. Wang, P. Shi, and X. Liu, "Dynamic learning from neural control for strict-feedback systems with guaranteed predefined performance," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2564–2576, Dec. 2016.
[12] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 763–768.
[13] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 2587–2592.
[14] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends Cogn. Sci.*, vol. 3, no. 6, pp. 233–242, 1999.
[15] D. Sternad and S. Schaal, "Segmentation of endpoint trajectories does not imply segmented control," *Exp. Brain Res.*, vol. 124, no. 1, pp. 118–136, 1999.
[16] Z. Li *et al.*, "Reinforcement learning of manipulation and grasping using dynamical movement primitives for a humanoidlike mobile manipulator," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 1, pp. 121–131, Feb. 2018.
[17] A. Coates, P. Abbeel, and A. Y. Ng, "Learning for control from multiple demonstrations," in *Proc. IEEE Int. Conf. Mach. Learn.*, 2008, pp. 144–151.
[18] S. Ekvall and D. Kragic, "Learning task models from multiple human demonstrations," in *Proc. IEEE Int. Symp. Robot Human Interact. Commun.*, 2006, pp. 358–363.
[19] T. Matsubara, S.-H. Hyon, and J. Morimoto, "Learning parametric dynamic movement primitives from multiple demonstrations," *Neural Netw.*, vol. 24, no. 5, pp. 493–500, 2011.
[20] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with Gaussian mixture models," *IEEE Trans. Robot.*, vol. 27, no. 5, pp. 943–957, Oct. 2011.
[21] T. Cederborg, M. Li, A. Baranes, and P.-Y. Oudeyer, "Incremental local online Gaussian mixture regression for imitation learning of multiple tasks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 267–274.
[22] S. Calinon, *Robot Programming by Demonstration: A Probabilistic Approach*. Lausanne, Switzerland: EPFL Press, 2009.
[23] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 2, pp. 286–298, Apr. 2007.
[24] H. G. Sung, "Gaussian mixture regression and classification," Ph.D. dissertation, Dept. Stat., Rice Univ., Houston, TX, USA, 2004.
[25] Z. Ju and H. Liu, "Fuzzy Gaussian mixture models," *Pattern Recognit.*, vol. 45, no. 3, pp. 1146–1158, 2012.
[26] B. Zhang, C. Zhang, and X. Yi, "Active curve axis Gaussian mixture models," *Pattern Recognit.*, vol. 38, no. 12, pp. 2351–2362, 2005.
[27] F. Ficuciello, R. Carloni, L. C. Visser, and S. Stramigioli, "Port-Hamiltonian modeling for soft-finger manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 4281–4286.
[28] C. Yang, X. Wang, Z. Li, Y. Li, and C.-Y. Su, "Teleoperation control based on combination of wave variable and neural networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 8, pp. 2125–2136, Aug. 2017.
[29] Z. Li, Z. Huang, W. He, and C.-Y. Su, "Adaptive impedance control for an upper limb robotic exoskeleton using biological signals," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 1664–1674, Feb. 2017.
[30] C. Yang *et al.*, "Global adaptive tracking control of robot manipulators using neural networks with finite-time learning convergence," *Int. J. Control Autom. Syst.*, vol. 15, no. 4, pp. 1916–1924, 2017.
[31] L. Cheng, Z.-G. Hou, M. Tan, and W.-J. Zhang, "Tracking control of a closed-chain five-bar robot with two degrees of freedom by integration of an approximation-based approach and mechanical design," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 5, pp. 1470–1479, Oct. 2012.
[32] F. Ke, Z. Li, H. Xiao, and X. Zhang, "Visual servoing of constrained mobile robots based on model predictive control," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1428–1438, Jul. 2016.
[33] C.-M. Lin and Y.-F. Peng, "Adaptive CMAC-based supervisory control for uncertain nonlinear systems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 2, pp. 1248–1260, Apr. 2004.
[34] S. Commuri, S. Jagannathan, and F. L. Lewis, "CMAC neural network control of robot manipulators," *J. Robot. Syst.*, vol. 14, no. 6, pp. 465–482, 1997.
[35] J. S. Albus, "A new approach to manipulator control: The cerebellar model articulation controller (CMAC)," *Trans. ASME J. Dyn. Syst. Meas. Control*, vol. 97, no. 3, pp. 220–227, 1975.
[36] M. Darka, "The control of a manipulator using cerebellar model articulation controllers," M.S. thesis, Dept. Mech. Eng., İzmir Inst. Technol., Urla, Turkey, 2003.
[37] Z. Li *et al.*, "Trajectory-tracking control of mobile robot systems incorporating neural-dynamic optimized model predictive approach," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 6, pp. 740–749, Jun. 2016.
[38] M. Wang, H. Ye, and Z. Chen, "Neural learning control of flexible joint manipulator with predefined tracking performance and application to Baxter robot," *Complexity*, vol. 2017, pp. 1–14, Oct. 2017. [Online]. Available: https://www.hindawi.com/journals/complexity/2017/7683785/
[39] M. Wang and A. Yang, "Dynamic learning from adaptive neural control of robot manipulators with prescribed performance," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 8, pp. 2244–2255, Aug. 2017.
[40] Z. Li, Y. Kang, Z. Xiao, and W. Song, "Human–robot coordination control of robotic exoskeletons by skill transfers," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 5171–5181, Jun. 2017.
[41] C. Yang *et al.*, "Interface design of a physical human–robot interaction system for human impedance adaptive skill transfer," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 1, pp. 329–340, Jan. 2018.
[42] Z. Ju and H. Liu, "A unified fuzzy framework for human-hand motion recognition," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 5, pp. 901–913, Oct. 2011.
[43] W. He and Y. Dong, "Adaptive fuzzy neural network control for a constrained robot using impedance learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1174–1186, Apr. 2018.
[44] B. Xian, D. M. Dawson, M. S. de Queiroz, and J. Chen, "A continuous asymptotic tracking control strategy for uncertain nonlinear systems," *IEEE Trans. Autom. Control*, vol. 49, no. 7, pp. 1206–1211, Jul. 2004.

Chenguang Yang (M'10–SM'16) received the B.Eng. degree in measurement and control from Northwestern Polytechnical University, Xi'an, China, in 2005, and the Ph.D. degree in control engineering from the National University of Singapore, Singapore, in 2010.

He received post-doctoral training at Imperial College London, London, U.K. His current research interest includes robotics and automation.

Dr. Yang was a recipient of the Best Paper Award from the IEEE TRANSACTIONS ON ROBOTICS and a number of international conferences.

Chuize Chen received the B.Eng. degree in automation from the South China University of Technology, Guangzhou, China, in 2017, where he is currently pursuing the M.S. degree.

His current research interests include human–robot interaction and machine learning and robotics.

Ning Wang (M'10) received the B.Eng. degree in measurement and control technologies and devices from the College of Automation, Northwestern Polytechnical University, Xi'an, China, in 2005, and the M.Phil. and Ph.D. degrees in electronic engineering from the Department of Electronic Engineering, Chinese University of Hong Kong, Hong Kong, in 2007 and 2011, respectively.

She was a Post-Doctoral Fellow with the Department of Computer Science and Engineering, Chinese University of Hong Kong, from 2011 to 2013. Her current research interests include signal processing and machine learning, with applications in robust speaker recognition, biomedical pattern recognition, intelligent data analysis, and human–robot interaction.

Zhaojie Ju (M'08–SM'16) received the B.S. degree in automatic control and the M.S. degree in intelligent robotics from the Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree in intelligent robotics with the University of Portsmouth, Portsmouth, U.K.

He is currently a Reader in machine learning and robotics with the School of Computing, University of Portsmouth. His current research interests include machine intelligence, pattern recognition and their applications on human motion analysis, human–robot interaction and collaboration, and robot skill learning. He has authored or co-authored over 110 publications in journals, book chapters, and conference proceedings.

Dr. Ju was a recipient of the four Best Paper Awards and one Best AE Award in ICRA2018. He is an Associate Editor of the IEEE TRANSACTIONS ON CYBERNETICS and the *International Journal of Humanoid Robotics*.

Jian Fu received the B.S. degree in industrial and electrical automation from the Wuhan University of Science and Technology, Wuhan, China, in 1994, the M.S. degree in computer applications from the Huazhong University of Science and Technology, Wuhan, in 1999, and the Ph.D. degree in control theory and control engineering from the University of Science and Technology Beijing, Beijing, China, in 2006.

He is currently an Associate Professor with the School of Automation, Wuhan University of Technology. His current research interests include human–robot collaboration, coexisting-cooperative-cognitive robots (Tri-Co robots), and machine learning.

Min Wang (M'09) received the B.Sc. and M.Sc. degrees from the Department of Mathematics, Bohai University, Jinzhou, China, in 2003 and 2006, respectively, and the Ph.D. degree from the Institute of Complexity Science, Qingdao University, Qingdao, China, in 2009.

Since 2017, she has been an Academic Visitor with the Department of Information Systems and Computing, Brunel University London, Uxbridge, U.K. She is currently an Associate Professor with the School of Automation Science and Engineering, South China University of Technology, Guangzhou, China. She has authored or co-authored nearly 30 papers in top international journals. Her current research interests include nonlinear systems, intelligent control, robot control, and dynamic learning.

Dr. Wang was a recipient of the Excellent Doctoral Dissertations Award of Shandong Province in 2010, the Science and Technology New Star of Zhujiang, Guangzhou, in 2014, and the Youth Talent of Guangdong Tezhi Plan in 2016. She is a very active reviewer for many international journals.