A robot learning framework based on adaptive admittance control and generalizable motion modeling with neural network controller

Ning Wang[a,1], Chuize Chen[b,1], Chenguang Yang[a,*]

[a] *Bristol Robotics Laboratory, University of the West of England, Bristol BS16 1QY, United Kingdom*
[b] *Key Laboratory of Autonomous Systems and Networked Control, College of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China*

Abstract

Robot learning from demonstration (LfD) enables the robots to be fast programmed, which involves the teaching phase, the learning phase and the reproduction phase. This paper proposes a novel LfD framework considering the performance of the methods used in these phases. An adaptive admittance controller is developed to take into account the unknown human dynamics so that the human tutor can smoothly move the robot around in the teaching phase. The task model in this controller is formulated by the Gaussian mixture regression to extract the human motion characteristics. In the learning and reproduction phases, the dynamic movement primitive is employed to model a robotic motion that is generalizable. A neural-network-based controller is designed for the robot to track the trajectories generated from the motion model, and a radial basis function neural network is used to compensate for the effect caused by the dynamic environments. The experiments have been performed using a Baxter robot and the results have confirmed the validity of the proposed methods.

Keywords: Robot learning, adaptive admittance control, motion generation, neural network

*Corresponding author
Email address: cyang@ieee.org (Chenguang Yang)
[1] Ning Wang and Chuize Chen contributed equally to this work.

1. Introduction

Robot learning from demonstration (LfD) has recently drawn much attention due to its high efficiency in robot programming[1]. Robots can learn variable skills from human tutor to complete tasks in complex industrial environment[2]. Compared to conventional programming methods using a teaching pendant, LfD is an easier and more intuitive way for people who are unfamiliar with programming. Besides, human characteristics involved in the demonstrations are available for robots to furtuer improve the flexibility and compliance of motions.

The applicability of LfD frameworks can be assessed according to the criteria defined in [3], which includes learning fatigue, adaptability, generality, accuracy and so on. It is difficult to satisfy all the criteria simultaneously. Thus, we can focus on one of the criteria in each phase of LfD.

The entire process of LfD includes the teaching phase, the learning phase and the reproduction phase. In the teaching phase, the human tutor demonstrates how to perform a task and the motion of the robot or human will be recorded. The criterion that needs to be satisfied in this phase is learning fatigue. In order to reduce learning fatigue, demonstrating should occur in an intuitive and easy way [3]. There are many methods to accomplish demonstration, such as directly guiding the robot or using visual devices to capture and transmit the human motion. In this paper, the teaching phase is accomplished by directly moving the end-effector of the robot beacuse it is more quicker and less loss of motion information. The learning phase is usually ignored in traditional industrial environment and the motion is directly used for reproduction. This will cause massive repetition of demonstrations when the tasks are similar, for example, pick-and-place tasks with different place targets. If a demonstration can be generalized to adapt to similar situations, the teaching process will be more efficient. Thus, the learning phase that models generalizable motion is necessary. The reproduction phase involves the trajectory tracking; thus, the

tracking accuracy of the robot dynamics controller should be guaranteed.

Admittance control has been widely used in human-robot interaction, which can generate robot motions based on the human force[4, 5, 6]. Thus, in this paper we use the admittance control to achieve the human-guided teaching. The admittance control exploits the end-effector position controller to track the output of an admittance model. Most studies on admittance control have not considered the human factor, which is an important part of this control loop. The interaction force between robot and human can be used to recognize the human intent, and to further improve the interaction safety and user experience[7]. In [8], the human force was employed to compute the desired movement trajectory of the human, which is used in the performance index of the admittance model. In [9], the unknown human dynamics was considered in the control loop. The human transfer function and the admittance model were formulated as a Wiener filter, and a task model was used to estimate the human intent. The Kalman filter estimate law was used to tune the parameters of the admittance model. Howerer, the task model in this work is assumed as a certain linear system, which is unreasonable beacuse the estimated human motions should be different for each individual due to different motion habits. Thus, a task model that involves the human characteristics need to be developed.

Gaussian mixture regression (GMR) is an effective algorithm to encode the human characteristics based on the human demonstrations[10, 11]. In [12], the human motion was analyzed using Gaussian mixture model and a new motion that involves the human motion distribution was generated using GMR. This algorithm shows great feasibility to develop a task model that involves the human characteristics.

In the learning phase of LfD, the robotic motion caused by the human guiding will be modelling. Dynamic system (DS) has been widely used to achieve the generalization of the motion model[12, 13, 14]. Dynamic movement primitive (DMP) is a powerful method to model generalizable motion based on DS[15, 16, 17]. It exploits a spring-damper system to guarantee the stability of the model, and uses a nonlinear function to motivate the model to generate motion that

keeps the characteristics of origin motion. It can be used to effectively model a series of primitive templates that are decomposed from a demonstration[18]. In [19], the DMP was used to model striking motion in robot table tennis. The learned model was used to generate motions that has different targets to hit the ball. To achieve trajectory joining and insertion effectively, a method called linearly decayed DMP+ extended the origin DMP by using truncated kernels and removing the problem of vanishing exponential time decay[20]. In this paper, the DMP is also integrated in our framework to improve the efficiency of LfD. Through adjusting the goal parameter of the DMP, we can generate a group of similar motions so that unnecessary repetition of demonstrations can be reduced.

The generated motion is finally used in reproduction, the accuracy of which depends on the performance of trajectory tracking controller. The controller design methods can be classified into the model-based methods and the model-free methods. The model-based method has better tracking accuracy because the robot dynamics is considered. Howerer, the accurate robot model is difficult to obtain. The function approximation methods such as neural network (NN) have been used to solve this problem[21, 22]. In [23], the backpropagation (BP) NN was employed to approximate the unknown model of the vibration suppression device, which achieved better control result. Compared to BPNN, the radial basis function (RBF) NN has a faster learning procedure and is more suitable for controller design. In this paper, we use the RBF NN to approximate the robot dynamics so that the robot can complete the reproducted motion accurately without the knowledge of the robot manipulator dynamics.

The contributions of this paper are as follows:

1) An adaptive admittance controller is developed to takes into account the unknown human dynamics. The task model in this controller is formulated by the GMR to extract the human motion characteristics.

2) A complete LfD framework that considers the teaching phase, the learning phase and the reproduction phase is developed, as shown in Fig. 1. In the

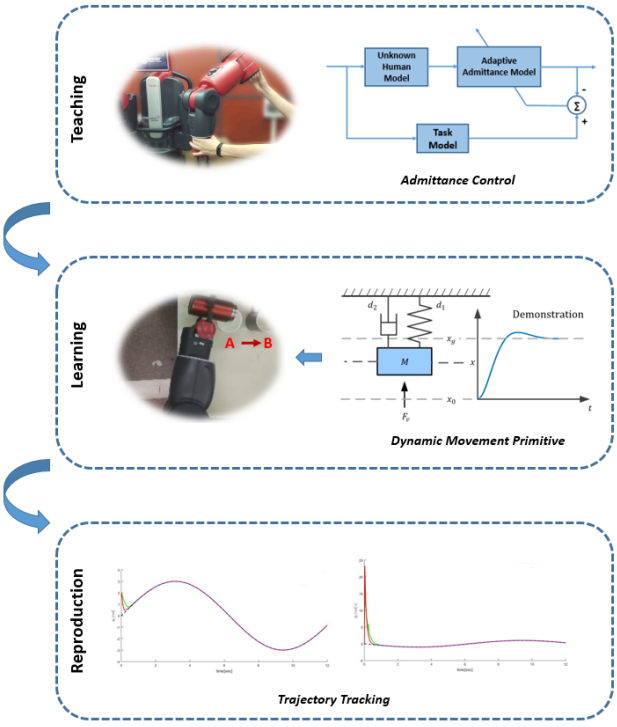


Figure 1: Overview of the proposed framework.

learning phase, the adaptive admittance controller described in 1) is employed so that the human tutor can smoothly guide the robot to accomplish the demonstration. In the learning phase, the DMP is used to model the robotic motion. The learned model can generalize the motion to adapt to different situations. In the reproduction phase, the RBF-NN-based trajectory track controller is developed to achieve accurate motion reproduction.

This paper is organized as follows. In section II, the methodology including the adaptive admittance control, the DMP and the NN-based controller will be inroducted. The experimental study is then presented in section III, Section IV finally concludes this paper.

2. Methodology

2.1. Adaptive Admittance Control with Demonstration-based Task Model

In this section, an adaptive task-specific admittance controller is developed. This adapts the parameters of the prescribed robot admittance model so that the robot system assists the human to achieve task-specific objectives. The task information is modelled by GMR so that the controller can adapt to the human tutor characteristics. After designing, the adaptive admittance controller will be used in the teaching phase for human tutor to demonstrate.

The prescribed admittance model is defined as follows:

$$M_m\ddot{x}_m + D_m\dot{x}_m + K_m x_m = f_h \tag{1}$$

where M_m is a prescribed mass matrix, D_m is a prescribed damping matrix, K_m is a prescribed spring constant matrix, and f_h is the human input force. The function of this admittance model is to generate the desired robot response $x_m(t)$, which serves as the human demonstration later.

The human-robot adaptive admittance control system with task model is shown in Fig. 2(a). The system input x_{tg} is the target of a point-to-point motion. In the teaching phase, we focus on the point-to-point motions beacuse they are very common type of motions in many tasks, and a more complex motion can be segmented into multiple point-to-point motions. Given a target x_{tg}, the human tutor will apply force on the end-effector of the robot and move it around. The intent of the human is to complete a motion that gets to this target. Using the admittance model, the desired robot response $x_m(t)$ from human input force is generated and then the robot tracks this generated trajectory to implement synchronous motion. In this process, it is expected that the interaction force between human and robot is relatively small so that the human can smoothly move the robot, which creates a better user experience. To achieve this objective, the parameters of the admittance model need to be tuned to adapt to the human tutor characteristics.

There are two factors need to be considered. First, we can estimate the human intent when a target is given. If the human motion is estimated, the

robot can be directly commanded to track this motion. However, the estimation relies on the interaction force and thus the admittance model is still necessary. Therefore, the second factor we should considered is the output of the admittance model. We need to adapt the parameters of the admittance model to minimize the error $\epsilon(t)$ between the desired human motion $x_d(t)$ and the actual output of the model.

The task model in Fig. 2(b) is employed to estimate the human motion. In [9], the task model is implemented as a linear system, and an exponential function is used to describe the human motion. However, this hypothesis is unreasonable beacuse the initial acceleration is not equal to zero for an exponential type trajectory. Beside, the estimated human motions should be different for each individual due to different motion habits. In this paper, the task model is developed using the GMR algorithm so that the human characteristics can be encoded properly. After the task model is learned, the recursive least square (RLS) algorithm [24] is used to adapt the parameters of the admittance model.

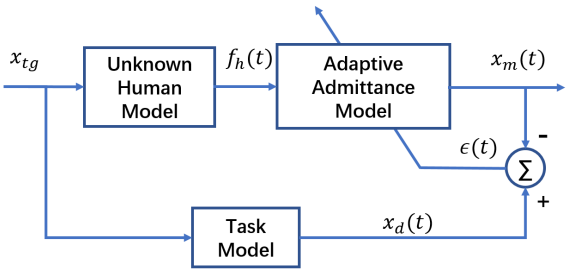
2.1.1. Demonstration-based task model

To develop the task model using GMR, human motion information is necessary for learning. To collect the motion data, the human tutor need to demonstrate the point-to-point motions with a fixed target, and then the recorded demonstration data is used for learning.

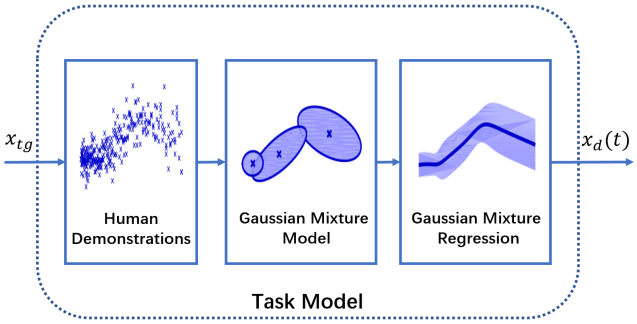
Assume that the demonstration data is represented as a dataset $O_b = \{o_1, ..., o_t, ..., o_{n_p}\}$ with $o_t = [o_{1t}, o_{2t}] \in R^2$, where $o_{1t} \in t$, $o_{2t} \in x_D(t)$, $x_D(t)$ is the recorded positions set of the demonstration, and n_p is the number of the data o_t. The distribution of O_b is first modeled by the GMM with finite Gaussian distributions, the probability density of which is:

$$p(O_b|\Theta) = \prod_{t=1}^{n_p} p(o_t|\Theta) = \prod_{t=1}^{n_p} \left(\sum_{i=1}^{n_g} \beta_i p(o_t|\theta_i) \right) \tag{2}$$

where $\Theta = (\beta_1, ..., \beta_{n_g}, \theta_1, ..., \theta_{n_g})$, $\beta_i \in R$ is the mixing weight with $\sum_{i=1}^{n_g} \beta_i = 1$, n_g is the number of the Gaussian distributions, and $\theta_i = (\rho_i, \kappa_i)$ is the



(a)



(b)

Figure 2: (a) Human-Robot Adaptive Admittance Control System. (b) Demonstration-base Task Model.

parameter of the i-th Gaussian distribution:

$$p(o_t|\theta_i) = \frac{\exp\left(-0.5(o_t-\rho_i)^T\kappa_i^{-1}(o_t-\rho_i)\right)}{2\pi\sqrt{|\kappa_i|}} \tag{3}$$

where $\rho_i \in R^2$ is the mean and $\kappa_i \in R^{2\times 2}$ is the covariance matrix:

$$\rho_i = \begin{bmatrix}\rho_{1i}\\ \rho_{2i}\end{bmatrix},\ \kappa_i = \begin{bmatrix}\kappa_{1i} & \kappa_{12i}\\ \kappa_{12i} & \kappa_{2i}\end{bmatrix} \tag{4}$$

The maximum likelihood estimation is employed to estimate the parameters of the GMM. the objective of which is to find $\hat{\Theta}$ that maximizes the log-likelihood function $\log(L(\Theta|O_b)) = \log(p(O_b|\Theta))$:

$$\hat{\Theta} = \arg\max_{\Theta} \log(L(\Theta|O_b)) \tag{5}$$

This problem can be solved by using the expectation-maximization (EM) algorithm[25].

Then the GMR is utilized to retrieve the task model output $x_d(t)$, which is defined as[26]:

$$x(t) = \sum_{i=1}^{n_g} \alpha_i(t)\eta_i(t) \tag{6}$$

with

$$\alpha_i(t) = \frac{\beta_i G(t|\rho_{1i}, \kappa_{1i})}{\sum_{i=1}^{n_g} \beta_i G(t|\rho_{1i}, \kappa_{1i})} \tag{7}$$

$$\eta_i(t) = \rho_{2i} + \frac{\kappa_{12i}}{\kappa_{1i}}(t - \rho_{1i}) \tag{8}$$

where $G(t|\rho_{1i}, \kappa_{1i})$ denotes the Gaussian function with the mean ρ_{1i} and the variance κ_{1i}.

2.1.2. Parameter adaptation of admittance model

The parameters of the admittance model can be adapted based on the task model output $x_d(t)$ and the human input force $f_h(t)$ by using the RLS algorithm.

Firstly the admittance model should be discretized. Define $x_m(k)$ as the value of x_m at time step k, and T_s as the sampling period. The velocity and the acceleration of $x_m(t)$ are defined as:

$$\dot{x}_m(k) = \frac{x_m(k) - x_m(k-1)}{T_s} \tag{9}$$

$$\ddot{x}_m(k) = \frac{x_m(k) - 2x_m(k-1) + x_m(k-2)}{T_s^2} \tag{10}$$

In this paper, the spring constant matrix K_m is assume as zero because the spring term will pull the motion back to the origin when the human input force is equal to zero. Then the discrete form of the admittance model (1) is:

$$h_0 x_m(k) + h_1 x_m(k-1) + h_2 x_m(k-2) = f_h \tag{11}$$

where

$$h_0 = \frac{M_m}{T_s^2} + \frac{D_m}{T_s} \tag{12}$$

$$h_1 = -(\frac{2M_m}{T_s} + \frac{D_m}{T_s}) \tag{13}$$

$$h_2 = \frac{M_m}{T_s^2} \tag{14}$$

To employ the RLS algorithm, the model (11) is rewritten as:

$$x_m(k) = H(k)^T Z(k) \tag{15}$$

where $H = [-h_0^{-1}h_1, -h_0^{-1}h_2, h_0^{-1}]^T$, and $Z(k) = [x_m(k-1), x_m(k-2), f_h]^T$. Define $\hat{H}(k)$ is the estimate of H. The RLS algorithm is used to minimize the error between the admittance model output and the task model output, which is defined as:

$$J = \sum_k ||\hat{H}(k)^T Z(k) - x_d(k)||^2 \tag{16}$$

The estimate update equations of RLS is:

$$\begin{aligned} R(k) &= I + Z^T(k)P(k-1)Z(k) && (17)\\ K(k) &= P(k-1)Z(k)R(k)^{-1} && (18)\\ \dot{P}(k) &= -K(k)Z^T(k)P(k-1) && (19)\\ \dot{\hat{H}}(k) &= K(k)[x_d(k) - \hat{H}(k-1)^T Z(k)] && (20) \end{aligned}$$

where $R(k)$ is an auxiliary variable, I is an identity matrix, $K(k)$ is the gain, $\hat{H}(k)$ is the estimated admittance model parameter matrix. The algorithm is initialized by setting $\hat{H}(0) = 0$ and a threshold for the error is set to ensure the convergence of the algorithm.

2.2. Generalizable Motion Modeling Using DMP

In the learning phase, the demonstration data in Cartesian space is used for motion modeling. The DMP model for Cartesian space motion is defined as follows:

$$\begin{aligned} \tau_s \dot{v} &= D_1(x_g - x) - D_2 v - D_1(x_g - x_0)s + D_1 f(s)\\ \tau_s \dot{x} &= v \end{aligned} \tag{21}$$

where $x \in R$ denotes the position variable in Cartesian space, x_0 is the start position, x_g is the targer, $v \in R$ is the velocity, $\dot{v} \in R$ is the acceleration, $D_1, D_2 \geq 0$ are the positive constants to be designed, $\tau_s > 0$ is the temporal-scaling factor, and $s \in R$ is defined as the state of the following DS called the

canonical system:

$$\tau_s \dot{s} = -\alpha_s s \tag{22}$$

where $\alpha_s > 0$ is the decay rate. Usually, the variable s is initialized as $s_0 = 1$. $f(s)$ is a continuous nonlinear function defined as follows:

$$f(s) = \sum_{i=1}^{n_s} \omega_i \phi_i(s) s \tag{23}$$

with

$$\phi_i(s) = \frac{\exp\left[-(s-a_i)^2/(2b_i)\right]}{\sum_{i=1}^{n} \exp\left[-(s-a_i)^2/(2b_i)\right]} \tag{24}$$

where $\phi_i(s)$ is the normalized Gaussian function, $a_i \in R$ is the mean and $b_i \in R$ is the variance, n is the number of the Gaussian components. $\omega_i \in R$ is the weight of the i-th Gaussian function.

The main part of the DMP model is a spring-damper system, which is perturbed by a nonlinear force as follows:

$$F_v = -D_1(x_g - x_0)s + D_1 f(s) \tag{25}$$

where $(x_g - x_0)$ serves as the spatial-scaling factor. According to (22), variable s in time domain is a exponential function, thus, s will converge to zero when $s_0 > 0$. Obviously, $f(s)$ and F_v will converge to zero, and variable x will converge to the goal x_g. The goal of the motion can be modulated by setting the value of x_g. Additionally, the duration of the motion is determined by the time-scaling factor τ_s. Therefore, we can generalize the motion in space and in time.

The model parameters ω_i are learned using the locally weighted regression (LWR)[27], which minimizes the following error:

$$\min \sum (f_{tg} - f(s))^2 \tag{26}$$

with

$$f_{tg} = \frac{\tau_s \dot{v} - D_1(x_g - x) + D_2 v}{x_g - x} \tag{27}$$

which can be computed based on a demonstrated trajectory.

2.3. Neural-based Controller for Motion Reproduction

Using the learned DMP model, the generalized motion in Cartesian space can be reproducted by adjusting τ_s and x_g. The motion trajectory is then transformed into an n-dimensional trajectory $q_d \in R^n$ in joint space using the inverse kinematics, and a neural-network-based controller is designed to track this trajectory. The RBF NN is utilized to estimate the dynamics uncertainties such as unknown nonlinearities and varying payloads.

2.3.1. RBF NN

RBF NN is an effective tool to approximate any continuous function $h : R^n \to R$ as follows:

$$h(x) = W^T S(x) + \varepsilon(x) \tag{28}$$

where $x \in R^n$ denotes the input vector, $W = [\omega_1, \omega_2, ..., \omega_N]^T \in R^N$ represents the ideal NN weight vector and N is the number of NN nodes. The approximation error $\varepsilon(x)$ is bounded. $S(x) = [s_1(x), s_2(x), ..., s_N(x)]^T$ is a nonlinear vector function, where $s_i(x)$ is defined as a radial basis function:

$$s_i(x) = \exp\left[-\frac{(x - c_i)^T (x - c_i)}{\chi_i^2}\right], i = 1, 2, ..., N \tag{29}$$

where $c_i = [c_{i1}, c_{i2}, ..., c_{in}]^T \in R^n$ denotes the centers of the Gaussian function and χ_i^2 is the variance. The ideal weight vector W is defined as follows:

$$W = \underset{\hat{W} \in R^N}{\arg\min} \left\{ \sup \left| h(x) - \hat{W}^T S(x) \right| \right\} \tag{30}$$

which minimizes the approximation error.

2.3.2. Controller Design

The dynamics of an n-link manipulator is described as follows[28]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \tag{31}$$

where $M(q)$ represents the inertia matrix, $C(q, \dot{q})$ denotes the Coriolis matrix, $G(q)$ is the gravity terms, and τ is the control torque.

Define the tracking error as $e_q = q - q_d$, the velocity error as $v = \dot{q}_d - \Lambda e_q$, where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_n)$, and the auxiliary variable $e_s = \dot{e}_q + \Lambda e_q$. The error dynamics can be written as:

$$M(q)\dot{e}_s + C(q,\dot{q})e_s + G(q) + M(q)\dot{v} + C(q,\dot{q})v = \tau \tag{32}$$

Design the control torque as:

$$\tau = \hat{G} + \hat{M}\dot{v} + \hat{C}v - Ke_s \tag{33}$$

where $\hat{G}(q)$, $\hat{M}(q)$ and $\hat{C}(q,\dot{q})$ are the estimates of $G(q)$, $M(q)$ and $C(q,\dot{q})$, respectively. Then the error dynamics is written as:

$$\begin{aligned} &M(q)\dot{e}_s + C(q,\dot{q})e_s + Ke_s \\ &\qquad = -(M - \dot{M})\dot{v} - (C - \hat{C})v - (G - \hat{G}) \end{aligned} \tag{34}$$

The RBF NN is then use to approximate the unknown robot dynamics:

$$\begin{aligned} M(q) &= W_M^T S_M(q) \\ C(q,\dot{q}) &= W_C^T S_C(q,\dot{q}) \\ G(q) &= W_G^T S_G(q) \end{aligned} \tag{35}$$

where W_M , W_C and W_G are the weight matrices; $S_M(q)$, $S_C(q,\dot{q})$ and $S_G(q)$ are the basis function matrices. Then the estimates of $M(q)$, $C(q,\dot{q})$ and $G(q)$ is written as:

$$\begin{aligned} \hat{M}(q) &= \hat{W}_M^T S_M(q) \\ \hat{C}(q,\dot{q}) &= \hat{W}_C^T S_C(q,\dot{q}) \\ \hat{G}(q) &= \hat{W}_G^T S_G(q) \end{aligned} \tag{36}$$

where $\hat{W}_M$, $\hat{W}_C$ and $\hat{W}_G$ are the estimates of W_M , W_C and W_G, respectively. Then the error dynamics is written as:

$$M\dot{e}_s + Ce_s + Ke_s = -\tilde{W}_M^T S_M \dot{v} - \tilde{W}_C^T S_C v - \tilde{W}_G^T S_G \tag{37}$$

where $\tilde{W}_M^T = W_M^T - \hat{W}_M^T$, $\tilde{W}_C^T = W_C^T - \hat{W}_C^T$, and $\tilde{W}_G^T = W_G^T - \hat{W}_G^T$.

Choose the Lyapunov function as:

$$\begin{aligned} V =& \frac{1}{2} e_s^T M e_s + \frac{1}{2} \mathrm{tr}(\tilde{W}_M^T \Gamma_M \tilde{W}_M) \\ & \frac{1}{2} \mathrm{tr}(\tilde{W}_C^T \Gamma_C \tilde{W}_C + \tilde{W}_G^T \Gamma_G \tilde{W}_G) \end{aligned} \tag{38}$$

where Γ_M, Γ_C and Γ_G are positive definite matrices. And the derivative of V is

$$\begin{aligned} \dot{V} =& - e_s^T K e_s \\ & - \mathrm{tr}[\tilde{W}_M^T (S_M \dot{v} e_s^T + \Gamma_M \dot{\hat{W}}_M)] \\ & - \mathrm{tr}[\tilde{W}_C^T (S_C v e_s^T + \Gamma_C \dot{\hat{W}}_C)] \\ & - \mathrm{tr}[\tilde{W}_G^T (S_G e_s^T + \Gamma_G \dot{\hat{W}}_G)] \end{aligned} \tag{39}$$

By designing the NN weight update law as follows:

$$\begin{aligned} \dot{\hat{W}}_M &= -\Gamma_M^{-1} S_M \dot{v} e_s^T \\ \dot{\hat{W}}_C &= -\Gamma_C^{-1} S_C v e_s^T \\ \dot{\hat{W}}_G &= -\Gamma_G^{-1} S_G e_s^T \end{aligned} \tag{40}$$

, we have:

$$\dot{V} = -e_s^T K e_s \leq 0 \tag{41}$$

Thus the tracking error e_q will converge to zero. Using the control law (33) and the NN weight update law (40), the robot can complete the reproducted motion accurately without the knowledge of the robot manipulator dynamics.

3. Experiment Study

The experiments are performed using a Baxter robot, which has two 7-DOF arms, as shown in Fig. 3. An ATI force sensor is attached on the end of the left arm to detect the human force. We verify the performance of the proposed framework by respectively testing the methods in each phase.

3.1. Parameter Adaptation of Admittance Model in Push-and-Pull Task

The push-and-pull experiment is conducted to test the performance of the adaptive admittance controller. The human tutor holds the left end-effector of

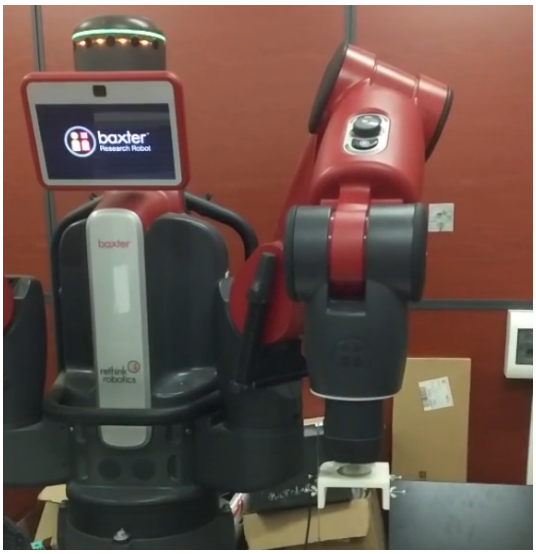


Figure 3: Experiment setup.

the robot and moves it between two specified points along the y-axis back and forth. Two specified points are set as $y_1 = 0$ (m) and $y_2 = 0.33$ (m). The period is set as 8s. Firstly, a admittance controller with fixed parameters is tested. The parameters is set as: $M_m = 10, D_m = 40, K_m = 0$. The human force in this process is recorded as shown in Fig. 6(a). Then this group of parameters is used to initialize the parameters in adaptive admittance controller with a sampling period $T_s = 0.02$s, which are transformed in RLS algorithm as: $H_1 = 1.92, H_2 = -0.92, H_3 = 0.0004$.

We conduct comparative experiments to verify the performance of the proposed task model. In the first experiment, a linear task model is employed, the output of which is exponential type function. In the second experiment, the human tutor first demonstrate this point-to-point motions four times and the demonstration is recorded to learn a task model using GMR. The number of the Gaussian components is set as 16. The learning result is shown in Fig. 4. Then this task model is used in the adaptive admittance controller. The output of the admittance model and the human force in both experiments are recorded as shown in Fig. 5 and in Fig. 6.

Table 1: PERFORMANCE COMPARISON OF THE PROPOSED METHODS

Task model	Linear	Demonstration-based
MSE	0.0314	0.0004
$Var(\Delta F)$	0.6898	0.4079

The pursued target of the proposed method is to provide a better interaction experience, which can be characterized as smaller human force. From the results we can find that the admittance controller with fixed parameters requires a relatively large human force in the interaction, while the adaptive admittance controller requires a relatively small force.

We use the mean square error (MSE) to quantify the output error of the admittance model:

$$\text{MSE} = \frac{1}{N}\sum_{k=1}^{N}(x_m(k) - x_d(k))^2$$

where N is the number of the samples, $x_m(k)$ is the admittance model output and $x_d(k)$ is the task model output. We also use the variance of the differences $Var(\Delta F)$ to quantify the smoothness of human force:

$$Var(\Delta F) = \frac{1}{N-1}\sum_{k=1}^{N-1}(f(k) - f(k+1) - E(\Delta F))^2$$

where $f(k)$ is the human force and $E(\Delta F)$ is the mean of the differences of human force. Smaller variance indicates better smoothness. The results are shown in TABLE 1, showing that the output error of the admittance model with demonstration-based task model is smaller than that of the admittance model with linear task model. And the smoothness of human force is better, which indicates better user experience for the human tutor.

3.2. Motion Generalization Using DMP in Pick-and-Place Task

In this experiment, the human tutor teaches the robot to complete a pick-and-place task. The human tutor holds the left end-effector of the robot and

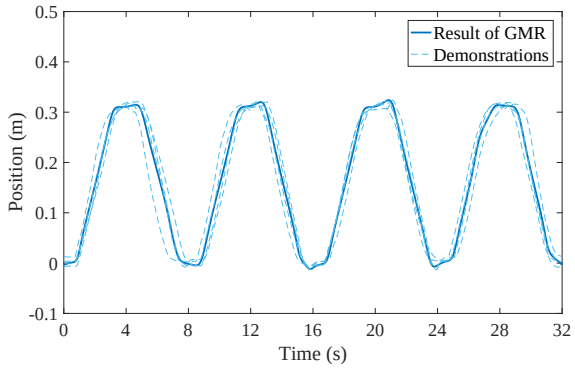


Figure 4: Learned task model using GMR.

moves it around to demonstrate this task. And then the recorded motion is used to train a DMP model. The number of the Gaussian function in DMP model is set as 10. The start position of the motion is $x = 0.7, y = 0.03, z = -0.14$ (m) and the target position is $x = 0.83, y = 0.32, z = -0.08$ (m). When the DMP model is learned, it is used to reproduct and generalize the motion. The target of the generalized motion is set as: $x = 0.90, y = 0.60, z = -0.08$ (m).

The demonstration, the reproduction, and the generalization are shown in Fig. 7. We can see that the target of the motion is successfully adjusted to a new position and the characteristics of motion is still kept. Thus we can use the DMP model to generate motions that can adapt to different similar task situations.

3.3. Trajectory Tracking Using NN-based Controller

We conduct comparative experiments to verify the performance of the NN-based controller on a joint of the Baxter. We set a desired trajectory for controller to track, which is defined as: $q = 0.5sin(2\pi t/4) + 1.0$. In the first experiment, a PD controller is used, the parameters is set as: $K_p = 15, K_d = 1.8$. In the second experiment, the proposed controller is employed. The NN node is set as 3^7 and the centers of the basis function are set uniformly distributed in the joint motion interval. The experiment is shown in Fig. 8 and in Fig. 9. We

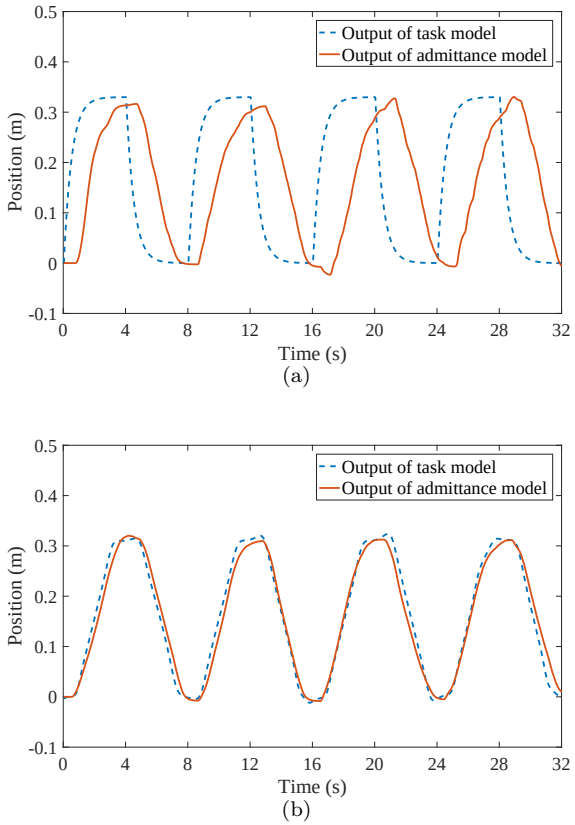


Figure 5: Model output: (a) Adaptive admittance controller with linear task model. (b) Adaptive admittance controller with Demonstration-based task model.

can see that the tracking error of the NN-based controller is smaller than that of the PD controller. Thus, the framework can enables the robot to accurately complete the motion in the reproduction phase.

4. Conclusion

In this paper, a novel robot learning framework based on adaptive admittance control and generalizable motion modeling is developed. This framework considers the performance of the methodology in each phase of LfD. A demonstration-based task model is developed using GMR to integrate the human characteristics into the adaptive admittance model. The DMP is used to

model generalizable motion in the learning phase and a RBF-NN-based controller is developed to track the reproducted motion accurately. In future work, we will develop a task model database to learning different human characteristics, so that the admittance model can adapt to different individuals.

References

[1] A. Billard, S. Calinon, R. Dillmann, S. Schaal, Robot programming by demonstration, in: Handbook of Robotics, Springer, 2008, pp. 1371–1394.

[2] S. Schaal, Is imitation learning the route to humanoid robots?, Trends Cognitive Sci. 3 (6) (1999) 233–242.

[3] W. K. H. Ko, Y. Wu, K. P. Tee, J. Buchli, Towards industrial robot learning from demonstration, in: Proceedings of the 3rd International Conference on Human-Agent Interaction, ACM, 2015, pp. 235–238.

[4] T. Tsuji, Y. Tanaka, Tracking control properties of human-robotic systems based on impedance control, IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans 35 (4) (2005) 523–535.

[5] C. Ott, R. Mukherjee, Y. Nakamura, Unified impedance and admittance control, in: Robotics and Automation (ICRA), 2010 IEEE International Conference on, IEEE, 2010, pp. 554–561.

[6] A. Morbi, M. Ahmadi, A. D. Chan, R. G. Langlois, Stability-guaranteed assist-as-needed controller for powered orthoses., IEEE Trans. Contr. Sys. Techn. 22 (2) (2014) 745–752.

[7] B. D. Argall, A. G. Billard, A survey of tactile human–robot interactions, Robotics and autonomous systems 58 (10) (2010) 1159–1176.

[8] I. Ranatunga, S. Cremer, D. O. Popa, F. L. Lewis, Intent aware adaptive admittance control for physical human-robot interaction, in: Robotics and Automation (ICRA), 2015 IEEE International Conference on, IEEE, 2015, pp. 5635–5640.

[9] I. Ranatunga, F. L. Lewis, D. O. Popa, S. M. Tousif, Adaptive admittance control for human-robot interaction using model reference design and adaptive inverse filtering., IEEE Trans. Contr. Sys. Techn. 25 (1) (2017) 278–285.

[10] S. Calinon, A. Billard, Statistical learning by imitation of competing constraints in joint space and task space, Adv. Robot. 23 (15) (2009) 2059–2076.

[11] S. Calinon, F. Guenter, A. Billard, On learning, representing, and generalizing a task in a humanoid robot, IEEE Trans. Syst., Man, Cybern. B, Cybern. 37 (2) (2007) 286–298.

[12] S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis, D. G. Caldwell, Statistical dynamical systems for skills acquisition in humanoids, in: Proc. IEEE-RAS Int. Conf. Humanoid Robots, 2012, pp. 323–329.

[13] S. M. Khansari-Zadeh, A. Billard, Learning stable nonlinear dynamical systems with gaussian mixture models, IEEE Trans. Robot. 27 (5) (2011) 943–957.

[14] J. Duan, Y. Ou, J. Hu, Z. Wang, S. Jin, C. Xu, Fast and stable learning of dynamical systems based on extreme learning machine, IEEE Trans. Syst., Man, Cybern. A, Syst. 49 (6) (2019) 1175–1185.

[15] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, S. Schaal, Dynamical movement primitives: learning attractor models for motor behaviors, Neural Comput. 25 (2) (2013) 328–373.

[16] F. Stulp, E. A. Theodorou, S. Schaal, Reinforcement learning with sequences of motion primitives for robust manipulation, IEEE Trans. Robot. 28 (6) (2012) 1360–1370.

[17] Y. Zhao, R. Xiong, L. Fang, X. Dai, Generating a style-adaptive trajectory from multiple demonstrations, Int. J. Adv. Robot. Syst. 11 (7) (2014) 103–111.

[18] Y. Wu, Y. Su, Y. Demiris, A morphable template framework for robot learning by demonstration: Integrating one-shot and incremental learning approaches, Robotics and Autonomous Systems 62 (10) (2014) 1517–1530.

[19] K. Mülling, J. Kober, O. Kroemer, J. Peters, Learning to select and generalize striking movements in robot table tennis, Int. J. Robot. Res. 32 (3) (2013) 263–279.

[20] Y. Wu, R. Wang, L. F. D'Haro, R. E. Banchs, K. P. Tee, Multi-modal robot apprenticeship: Imitation learning using linearly decayed dmp+ in a human-robot dialogue system, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2018, pp. 1–7.

[21] X. Ren, A. Rad, F. L. Lewis, Neural network-based compensation control of robot manipulators with unknown dynamics, in: American Control Conference, 2007. ACC'07, IEEE, 2007, pp. 13–18.

[22] C. Yang, T. Teng, B. Xu, Z. Li, J. Na, C.-Y. Su, Global adaptive tracking control of robot manipulators using neural networks with finite-time learning convergence, Int. J. Control Autom. Syst. 15 (4) (2017) 1916–1924.

[23] Z. Mao, F. Zhao, Structure optimization of a vibration-suppression device for underwater moored platforms using cfd and neural network, Complexity 2017 (2017) 21.

[24] M. H. Hayes, Statistical digital signal processing and modeling, John Wiley & Sons, 2009.

[25] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the em algorithm, J. R. Stat. Soc. 39 (1977) 1–38.

[26] H. G. Sung, Gaussian mixture regression and classification, Ph.D. thesis, Rice University (2004).

[27] C. G. Atkeson, A. W. Moore, S. Schaal, Locally weighted learning for control, in: Lazy learning, Springer, 1997, pp. 75–113.

[28] D. M. Dawson, C. T. Abdallah, F. L. Lewis, Robot manipulator control: theory and practice, CRC Press, 2003.

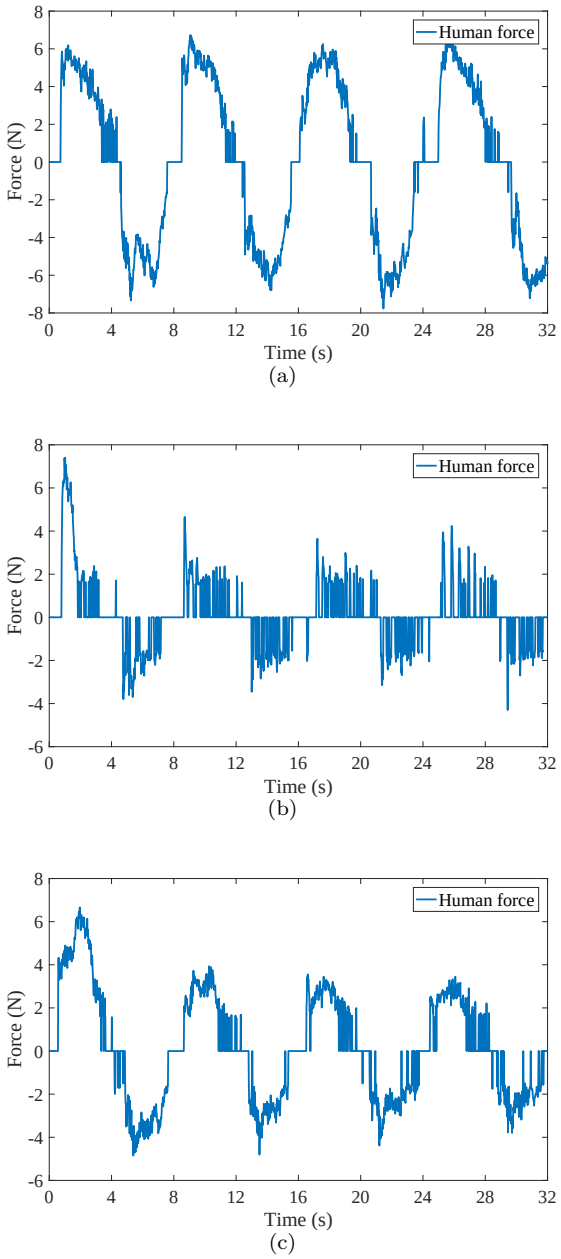


Figure 6: Human force: (a) Admittance controller with fixed parameters. (b) Adaptive admittance controller with linear task model. (c) Adaptive admittance controller with Demonstration-based task model.

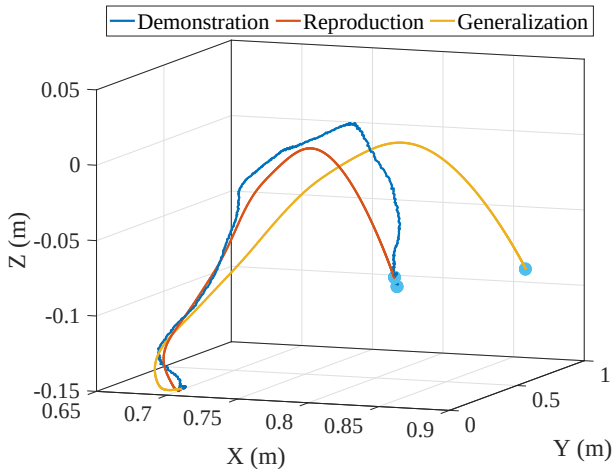


Figure 7: Learning results of DMP.

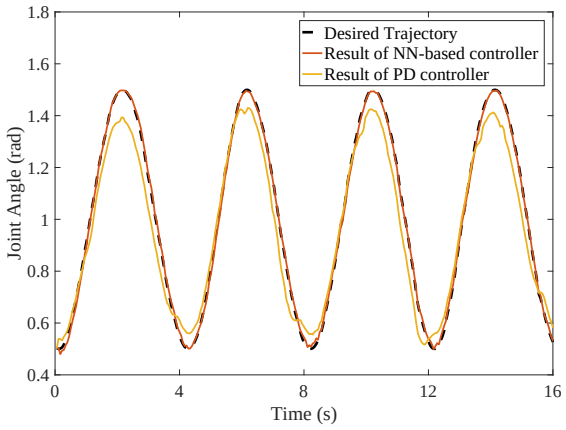


Figure 8: Trajectory tracking results of the NN-based controller and the PD controller.

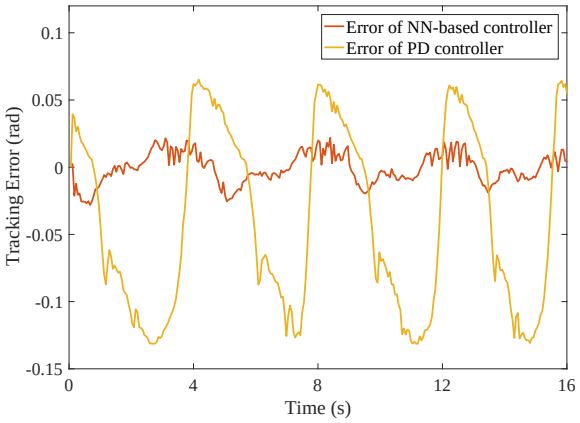


Figure 9: Trajectory tracking errors of the NN-based controller and the PD controller.