



## Research Article

# Learning Force-Relevant Skills from Human Demonstration

Xiao Gao ,<sup>1</sup> Jie Ling ,<sup>1</sup> Xiaohui Xiao ,<sup>1,2</sup> and Miao Li ,<sup>1,3</sup>

<sup>1</sup>School of Power and Mechanical Engineering, Wuhan University, China

<sup>2</sup>National Key Laboratory of Human Factors Engineering, China Astronauts Research and Training Center, Beijing, China

<sup>3</sup>The Institute of technological Sciences, Wuhan University, China

Correspondence should be addressed to Xiaohui Xiao; [xhxiao@whu.edu.cn](mailto:xhxiao@whu.edu.cn) and Miao Li; [miao.li@whu.edu.cn](mailto:miao.li@whu.edu.cn)

Received 11 August 2018; Revised 17 November 2018; Accepted 30 December 2018; Published 3 February 2019

Guest Editor: Junpei Zhong

Copyright © 2019 Xiao Gao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many human manipulation skills are force relevant, such as opening a bottle cap and assembling furniture. However, it is still a difficult task to endow a robot with these skills, which largely is due to the complexity of the representation and planning of these skills. This paper presents a learning-based approach of transferring force-relevant skills from human demonstration to a robot. First, the force-relevant skill is encapsulated as a statistical model where the key parameters are learned from the demonstrated data (motion, force). Second, based on the learned skill model, a task planner is devised which specifies the motion and/or the force profile for a given manipulation task. Finally, the learned skill model is further integrated with an adaptive controller that offers task-consistent force adaptation during online executions. The effectiveness of the proposed approach is validated with two experiments, i.e., an object polishing task and a peg-in-hole assembly.

## 1. Introduction

Manipulation skill is one of the most important capabilities that a robot is expected to have. During the past decades, a large number of studies have been done on robot manipulation in free space in terms of planning and control. However, in many scenarios, a robot is required to physically interact with humans or its environment, such as the human-robot cooperation in household or in the industrial setting like polishing, deburring, assembly, etc. Apparently, the information about the interacting force between the robot and its environment is of great importance in these tasks. Therefore, in order to successfully accomplish these manipulation tasks for robots, it is necessary to endow them with force-relevant skills.

To this end, during the past decades, various control algorithms have been proposed and ported to different robot platforms to accomplish a large variety of interaction tasks [1]. These algorithms can be roughly divided into two groups, named passive interaction control and active interaction control. The first group uses the mechanical design to empower robots with passive compliance that can roughly accommodate some force-motion relation during interaction. The most

notable example is the remote center of compliant (RCC) device used in industrial assembly [2], especially for the peg-in-hole case. The RCC adapts its motion passively to the unexpected forces during insertion process. In general, these compliant mechanisms are specifically designed for some interaction tasks and they can only provide a limited range of compliance. The other group encompasses active force control approaches that modulate the interacting force explicitly according to some task status variables. Typical exemplar algorithms include hybrid force/position control [3] and the well-known impedance control [4]. The active force control approach offers more flexibility to accommodate the interaction between the robots and the environment. The flexibility is usually achieved by an explicit specification of the motion and the force profile for the given interaction task.

In order to ensure the stability of the force controller, a typical active force control approach requires an accurate dynamic model of the contact interactions [1], which is not a trivial task, especially when the environment to be interacted with is unknown or varying over time. In these cases, a small uncertainty in position may lead to an extremely large reaction force that could be unfavorable for the robots and the environment. To solve this problem, usually a careful

parameter tuning is demanded for interaction tasks such as polishing and assembly. Besides the controller itself, the specification for the trajectory and force profile can greatly influence the performance of the interaction task. However, this planning process could be considerably time-consuming for the non-robotics-experts to apply a robot manipulator to a new interaction task. Moreover, the profile of the trajectory and the force during execution may need to adapt to the task requirements or to the variations in the environments. Hence, sensory feedback such as force information should be taken into account to monitor the status of the task completion and to adapt the trajectory accordingly.

Recently, learning-based approaches have been extensively applied for manipulation tasks, mainly including reinforcement learning [5–7] and Learning from Demonstration (LfD), or called imitation learning [8–10]. LfD can benefit from human guided demonstrations or simulations, which in general requires less training data and thus less time to train and deploy. This merit is practically important to teach a new manipulation skill to a robot, especially in industrial settings where deployment is usually time-limited. Therefore, we restrict the rest of this review to LfD for manipulation tasks only.

Depending on the tasks at hand, many researchers have used LfD for position-based manipulation tasks. In these tasks, the skills to be learned are usually encoded in the trajectory level in terms of position and velocity profile [11–13]. More recently, for interaction tasks, the profile of stiffness and impedance has been taken into account for skill learning to encapsulate the relation between forces and positions [14–18]. For example, in [19], EMG signal of a human arm was introduced to encode position and stiffness features. However, these works only *implicitly* capture the force characteristics with respect to the positions, which means that the precise value of the applied force is not so critical.

In this work, we mainly focus on *force-relevant* skills where the force profiles should be encoded *explicitly* [20]. However, it is not trivial to explicitly demonstrate and encode the applied force skills for a given task. The interaction force information is hard to be measured directly. Moreover, the correlation between the interaction force and the motion is also task-dependent and difficult to specify. Lin et al. presented a motion and force learning framework for grasping tasks [21]. The motion and force were modeled using temporal information with Gaussian Mixture Model (GMM) based machine learning approaches. Kormushev et al. adopted kinesthetic teaching and haptic input for demonstration of two manipulation tasks, namely, an ironing task and a door-opening task [22]. Time was considered as an additional input variable in these papers, which may lead to large time discrepancies to handle. In [23], a Hidden Markov Model (HMM) was adopted to encode force-based manipulation skill for a ball-in-box task. A haptic device was also exploited for teleoperation and for improving the teacher's demonstrations. As for more dexterous tasks, such as opening a bottle cap or inserting a bulb into the socket where multiple fingers (or manipulators) were involved, the learned skill was usually demonstrated and learned in the object-level. In [14],

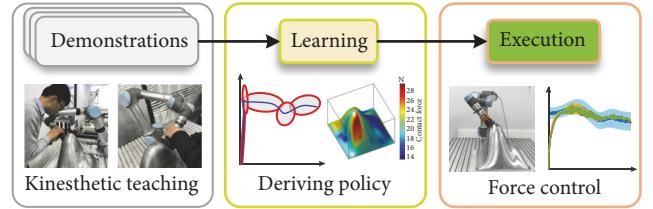


FIGURE 1: Framework of learning force-relevant skills. With multiple human demonstrations, a probabilistic task representation model is learned, from which a trajectory and force profile for the task are derived.

the force applied on the object was measured with a high resolution tactile sensor. A data glove mounted with tactile sensing was used for direct demonstration in [24]. Based on the similarity of varying demonstrations, the learned force-based skills were modularized and can be further combined for more complicated tasks. For other fine manipulation tasks such as assembly and surface-surface alignment, kinesthetic teaching with manual corrections was used to capture the important spatial relationships [25] or to encode the force-velocity correlations [26]. In these researches, the relationship and distributions of position and force were used to guide the design of the task planner but not to adjust parameters of the force controller.

In this paper, we present a learning approach from demonstration framework for force-relevant skills as shown in Figure 1. Inspired by kinesthetic teaching approaches as in [27, 28], we propose a demonstration method to allow the demonstrators to teach interaction tasks in a natural way. The demonstrated motion and force information are recorded simultaneously and encoded as a joint probability distribution without using temporal input. A task planner and an adaptive control policy are derived from this joint model to enable online task executions. Our work differs from these works in the sense that the learned model is used not only for fast planner generation but also for designing adaptive force control policy and the learned model. The main contributions of this paper are twofold: (1) A systematic framework is proposed in Figure 2 to learn the force-relevant skill as a statistical model, which essentially encapsulates the correlation between the interaction force and the motion; (2) based on the learned skill model, a task planner is devised to specify the desired motion and/or the force profile for a given manipulation task, and an adaptive force controller is designed for online executions.

This paper is organized as follows: in Section 1, the background and some related works regarding learning-based approaches for interaction manipulation tasks are summarized. In Section 2, the representation of the interaction manipulation tasks is formulated. In Section 3, methods for learning of force-relevant skills from human demonstration are presented. Experimental results on polishing and assembly tasks are demonstrated and discussed in Section 4, along

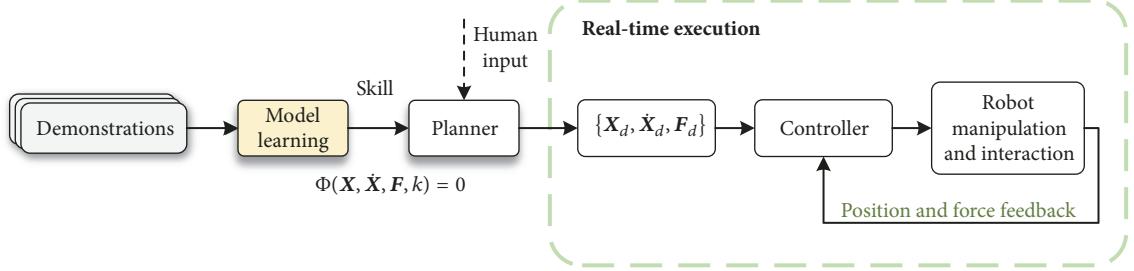


FIGURE 2: Framework of learning force-relevant skills.

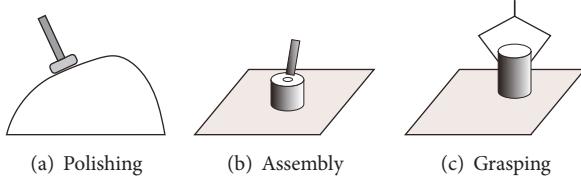


FIGURE 3: Typical examples of force-relevant skills, including polishing, assembly, and grasping.

with discussion about limitations. Finally, conclusions and future work are presented in Section 5.

## 2. Representation of Force-Relevant Skills

In this section, we will first introduce the representation of force-relevant skills, followed by several typical examples.

**2.1. Force-Relevant Skill Representation.** Contact interaction tasks of a robot manipulator require compliant behavior, including interaction force and end-effector position adjustment, which can be described as follows:

$$\Omega : \{\mathbf{X}, \dot{\mathbf{X}}, \mathbf{F}, k\} \quad (1)$$

where  $\Omega$  is a specific task.  $\mathbf{X}$  and  $\dot{\mathbf{X}}$  are the desired position and velocity in task space.  $\mathbf{F}$  is interaction force/torque (wrench) vector.  $k$  means the task constraint.

Skill acquisition is to find the internal correlation of these parameters. Skill of task  $\Omega$  can be represented as

$$\Phi(\mathbf{X}, \dot{\mathbf{X}}, \mathbf{F}, k) = 0 \quad (2)$$

Parameters and their internal correlation can be learned by learning algorithms presented in Section 3.

**2.2. Typical Examples.** Typical examples of force-relevant skills in robot manipulation include polishing, grinding, grasping and assembling in Figure 3. In these application scenarios, interaction force determines the quality of task execution.

Although most grinding and polishing operations are done manually or automated by robots in pure motion control with high accuracy and speed, force control is still necessary to obtain higher machining quality [20].

Grinding and polishing tasks have same characteristics. They require a robot manipulator equipped with a machining head as an end-effector to move along position trajectories attached to the workpiece, while contact forces are exerted on the normal direction. In this case, hybrid force/position control is often adopted to control contact force and movement trajectory on orthogonal subspaces. The task constraint  $k$  is to keep the position of polishing point along the tangential direction of the surface and limit the contact force within a proper range. Contact force is one of the most important technological parameters for the machining quality, such as dimensional tolerance, tolerance of form and position and surface roughness. This kind of skills could be represented as a mapping function  $f() : \{\mathbf{X}, \dot{\mathbf{X}}\} \xrightarrow{k} \mathbf{F}$ . For a certain position  $\mathbf{X}$  and velocity  $\dot{\mathbf{X}}$ , contact force  $\mathbf{F}$  is calculated as a conditional distribution. When there is a new polishing path by human kinesthetic teaching or human input, the force and velocity profiles are generated. However, precise force control is not necessary. It is enough to control the contact force within a proper range for the requirements of these tasks.

As for grasping and assembling tasks, force/torque data imply information interaction state. Human can complete these tasks only by haptic feedback without visual feedback. The position, orientation, and velocity of the task end-effector are corrected by force/torque feedback. In peg-in-hole assembly, the constraint  $k$  is to restrain the xy-axis force and torque to zero and xy-axis linear velocity and angular velocity to zero. Therefore, skills are adjustment strategies as a mapping function  $g() : \mathbf{F} \xrightarrow{k} \{\mathbf{X}, \dot{\mathbf{X}}\}$ . A controller is designed by adjusting position  $\mathbf{X}$  and velocity  $\dot{\mathbf{X}}$  based on current interaction force/torque under constraint condition  $k$ , and tasks could be carried out.

## 3. Learning of Force-Relevant Skills

**3.1. Human Demonstration.** In LfD, choosing an appropriate technique to perform human demonstration and record data is vitally important. Methods can be classified into two categories: manipulating robots via kinesthetic teaching and executing the task directly by a demonstrator, as shown in Figure 4.

Kinesthetic teaching is the major technique for directly transferring human experience to robot. It allows a human to touch and handle the robot's body with hands. The method of *hands on tools* is achieved by using sensed force data to

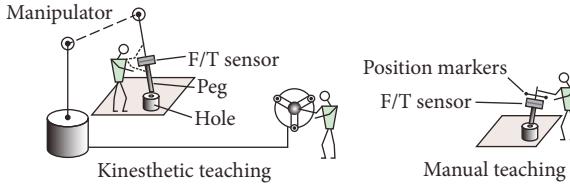


FIGURE 4: Different demonstration methods. Left: kinesthetic teaching with human hands on tools or robot arms, including teleoperation with a remote control device. Right: human direct teaching with recorded sensor data.

guide the robot. The method of *hands on arms* is based on torque-controlled backdrivable joints. So manipulation tasks can be accomplished under experienced human guidance. The trajectory of end-effector movement and contact force are recorded simultaneously by the robot's encoders and F/T sensor. Another common way is by teleoperation. A remote control device, such as a joystick or data glove, is handled as a master to control the robot by motion mapping. So interaction force is unknown for human hands. It is often used as kinematic demonstration for trajectory record in LfD.

The other approach for human demonstration is to perform tasks directly with human hands, rather than guiding the robot [26]. A force/torque sensor and 3D vision trackers are equipped on the end-effector to record the interaction force and hands' movement respectively. This is the most direct demonstration approach. Delicate force and position control strategy can be realized. However, the recorded data must be transferred to robots. Many differences between demonstrators and robots lead to correspondence issues for direct mapping [29], which are still challenging tasks for effective transfers. Another disadvantage of this technique is that expensive sensors for 3D vision position tracking and force/torque measurement are required.

For specific manipulation tasks, we may adopt different demonstration methods for better data recording.

**3.2. Skill Learning.** With a set of demonstrations under human guidance, machine learning is adopted in terms of encoding and reproduction of a skill. In this subsection, GMM is used to encode demonstration data as a probabilistic model. Then, Gaussian Mixture Regression (GMR) is used to predict the desired skills [30].

Considering a framework of skill representation in (1), we define a dataset  $\xi = [\mathbf{F}^T, \mathbf{X}^T]^T$  or  $\xi = [\mathbf{X}^T, \mathbf{F}^T]^T$  (determined by different skills), where  $\mathbf{F} \in \mathbb{R}^{D_F \times N}$  and  $\mathbf{X} \in \mathbb{R}^{D_X \times N}$  are the interaction force and trajectory, respectively. The dataset is encoded with GMM, a mixture of Gaussian distributions in  $K$  components. The probability of a datapoint  $\xi_0 \in \mathbb{R}^D$  in dataset  $\xi = [\xi^i, \xi^o]$  under the GMM is

$$\begin{aligned} p(\xi_0) &= \sum_{k=1}^K \pi_k \mathcal{N}(\xi_0; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ &= \sum_{k=1}^K \pi_k \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}_k|}} e^{-(1/2)(\xi_0 - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\xi_0 - \boldsymbol{\mu}_k)} \end{aligned} \quad (3)$$

where  $\pi_k \in [0, 1]$  are prior probabilities;  $\sum_{k=1}^K \pi_k = 1$ ,  $\boldsymbol{\mu}_k \in \mathbb{R}^D$ ,  $\boldsymbol{\Sigma}_k \in \mathbb{R}^{D_i \times D_o}$  represent means and covariance matrices of the Gaussian in the GMM. Then we define the means and covariance matrices of input and output components as

$$\begin{aligned} \boldsymbol{\mu}_k &= \begin{bmatrix} \boldsymbol{\mu}_k^i \\ \boldsymbol{\mu}_k^o \end{bmatrix}, \\ \boldsymbol{\Sigma}_k &= \begin{bmatrix} \boldsymbol{\Sigma}_k^i & \boldsymbol{\Sigma}_k^{io} \\ \boldsymbol{\Sigma}_k^{oi} & \boldsymbol{\Sigma}_k^o \end{bmatrix} \end{aligned} \quad (4)$$

For a given input variable  $\xi^i$ , the conditional probability distribution of the output  $\xi^o$  can be written:

$$p(\xi^o | \xi^i) = \sum_{k=1}^K p(k | \xi^i) \mathcal{N}(\hat{\xi}_k, \hat{\boldsymbol{\Sigma}}_k) \quad (5a)$$

where

$$\hat{\xi}_k = \boldsymbol{\mu}_k^o + \boldsymbol{\Sigma}_k^{oi} (\boldsymbol{\Sigma}_k^i)^{-1} (\xi^i - \boldsymbol{\mu}_k^i) \quad (5b)$$

$$\hat{\boldsymbol{\Sigma}}_k = \boldsymbol{\Sigma}_k^o - \boldsymbol{\Sigma}_k^{oi} (\boldsymbol{\Sigma}_k^i)^{-1} \boldsymbol{\Sigma}_k^{io} \quad (5c)$$

The weighting function  $p(k | \xi^i)$  represents the probability that  $k$ -th Gaussian component is responsible for  $\xi^i$

$$\begin{aligned} p(k | \xi^i) &= \frac{p(k) p(\xi^i | k)}{\sum_{j=1}^K p(j) p(\xi^i | j)} \\ &= \frac{\pi_k \mathcal{N}(\xi^i; \boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i)}{\sum_{j=1}^K \pi_j \mathcal{N}(\xi^i; \boldsymbol{\mu}_j^i, \boldsymbol{\Sigma}_j^i)} \end{aligned} \quad (6)$$

GMR is achieved by calculating the conditional expectation of  $\xi^o$ , given  $\xi^i$ , in (5a):

$$E\{p(\xi^o | \xi^i)\} = \sum_{k=1}^K p(k | \xi^i) \hat{\xi}_k \quad (7)$$

GMM/GMR is then described by a set of parameters  $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ , which are estimated iteratively by the Expectation Maximization (EM) algorithm. The hyperparameter  $K$ , namely, the number of Gaussian components, is selected using Bayesian Information Criterion (BIC).

### 3.3. Task Execution

**3.3.1. Adaptive Hybrid Force/Position Control.** A task planner is to specify the motion and force profile and plan compliant motion commands, shown in Figure 2. The motion and force profile could be generated by learned skills or given from human input.

In scenarios of force-relevant tasks, position and force control is required to accomplish compliant behavior. A popular approach is to adopt hybrid force/position control scheme, which separates the task of position and force control into two orthogonal subspaces. Aiming at position/velocity

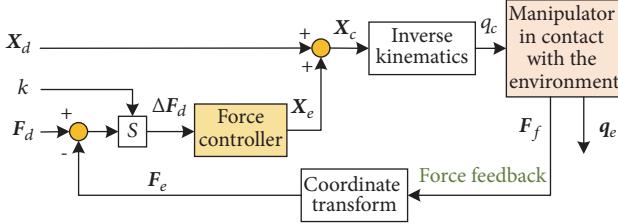


FIGURE 5: Scheme of hybrid force/position control for force-relevant tasks.  $F_d$  refers to the desired contact force.

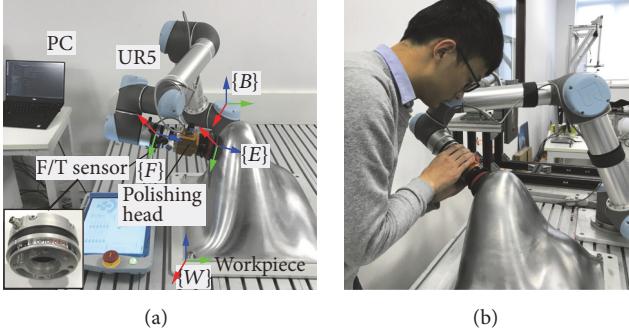


FIGURE 6: Polishing workstation. (a) Setup of the polishing workstation. The workpiece is manufactured as a complex surface of the aluminium alloy with an overall dimension 400 mm × 400 mm × 300 mm. (b) Kinesthetic teaching via a demonstrator.

controlled robot manipulators, we establish an adaptive force/position controller based on planned trajectories and contact force distribution from the learned model, shown in Figure 5. The control law  $X_c \in \mathbb{R}^{6 \times 1}$  is calculated combining the desired position  $X_d$  and the adjustment  $X_F$  from force controller. We take an adaptive PI controller as the force controller. All coordinates are marked in Figure 6(a), including the robot base link  $\{B\}$ , the force sensor  $\{F\}$ , end-effector  $\{E\}$ , and the workpiece  $\{W\}$ . The sensory wrench  $F_f$  is in coordinate system  $\{F\}$  and the transformation from  $\{F\}$  to  $\{E\}$  is represented as a rotation matrix  ${}^F_R \in \mathbb{R}^{3 \times 3}$  and a translation vector  ${}^F_P \in \mathbb{R}^3$ . With inertia force ignored, the contact force  $F_e$  in  $\{E\}$  is calculated as

$$F_e = {}_F^E T F_f \quad (8a)$$

with

$${}_F^E T = \begin{bmatrix} {}^E R & \mathbf{0}_{3 \times 3} \\ P_t {}^E R & {}^E R \end{bmatrix} \quad (8b)$$

where  ${}_F^E T \in \mathbb{R}^{6 \times 6}$  is the force transform matrix.  $P_t$  is the skew-symmetric matrix of the vector  ${}^E P$ .

The whole controlled motion command  $X_c$  is given by

$$X_c = X_d + X_e \quad (9a)$$

$$X_e = K_p^f \Delta F_d + K_i^f \int \Delta F_d dt, \quad K_p^f = \frac{K_p^{f_0}}{\sigma_{F_d}} \quad (9b)$$

$$\Delta F_d = S(F_d - F_e) \quad (9c)$$

where  $X_e \in \mathbb{R}^{6 \times 1}$  is the adjustment to tracking the desired force  $F_d \in \mathbb{R}^{6 \times 1}$ .  $K_p^{f_0}, K_i^f \in \mathbb{R}_{6 \times 6}$  are constant parameters of the PI controller.  $\sigma_{F_d}$  represents the standard deviation of the force distribution.  $S$  is the diagonal selection matrix under constraint  $k$ . As precise force control is difficult for force-relevant tasks in complex environment, keeping the interaction force in a proper range is enough for the task requirements. We design this adaptive PI control for force tracking. A smaller value of  $\sigma_{F_d}$  implies more accurate force is desired. Thus a bigger proportional parameter  $K_p^f$  is calculated as (9b) for rapidly tracking the desired force.

**3.3.2. Task Performance Evaluation.** Force-relevant skills have their specific requirements between different tasks. For machining process, such as polishing and burring, controlling contact force in a proper range is required for machining quality. A natural approach to evaluate tasks is to measure the machining quality, which is hard to detect and highly relevant to the demonstrator and processing technology. As precise force control is not necessary, tracking performance could be partly evaluated by the proportion of effective time of force in the proper range.

Considering interaction tasks of more complex manipulation, such as assembly and grasping, we can define performance evaluation criteria on account of characteristics of task executions. Task success rate and completion time are directly visualized parameters. The norm of contact force/torque and average of energy consumption refer to inner interaction quality. Excessive contact force/torque may damage the end-effector or workpiece. More energy consumption is against extensive use of automation, which can be calculated as follows:

$$W_e = \sum_{i=1}^n \int_0^T U_i I_i dt \quad (10)$$

where  $n$  refers to joint numbers,  $U_i, I_i$  for the voltage and current of  $i$ -th joint and  $T$  is the task completion time.

## 4. Experiments

In this section, we set up two kinds of force-relevant tasks, a surface polishing task and a peg-in-hole assembly, to validate the proposed framework. The experiments were conducted with UR robot manipulators and an OptoForce six-axis F/T sensor. UR robots were controlled and programmed by using the Robot Operating System (ROS) through the UR driver [31], with a control rate of 125 Hz.

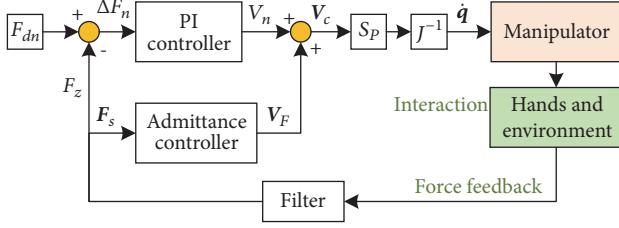


FIGURE 7: Block diagram of control system for polishing kinesthetic teaching.  $F_s \in \mathbb{R}^6$  is the sensed force and is used for teaching.

**4.1. Polishing.** We present our experimental results of surface polishing task, which requires the end-effector to exert a prescribed force normal on a given surface and follow a predefined motion trajectory attached to the surface. The experiment setup is shown in Figure 6.

**4.1.1. Polishing Demonstration and Data Collection.** For polishing tasks, we adopt the method of kinesthetic teaching, which allows a human to touch and handle the robot's body directly. Although the UR has *force mode* with external force estimation, it cannot make a natural kinesthetic teaching due to the imprecise force estimation. Traditional demonstration methods are trajectory-based, which may cause separation between the polishing head and the workpiece. Therefore, we design a kinesthetic teaching method combining a PI tracking controller and an admittance teaching controller, as shown in Figure 7. The robot is velocity controlled by

$$\dot{\mathbf{q}} = J^{-1} S_p \mathbf{V}_c \quad (11a)$$

$$\mathbf{V}_c = [0 \ 0 \ V_n \ 0 \ 0 \ 0]^T + \mathbf{V}_F \quad (11b)$$

with

$$V_n = K_p^D \Delta F_n + K_i^D \int \Delta F_n dt, \quad (11c)$$

$$M_a^D \dot{\mathbf{V}}_F + D_a^D \mathbf{V}_F = \mathbf{F}_s \quad (11d)$$

where  $\dot{\mathbf{q}} \in \mathbb{R}^6$  is the vector of joint velocities of robot manipulator and  $J \in \mathbb{R}^{6 \times 6}$  is the Jacobian matrix of the robot from  $\{E\}$  to  $\{B\}$ .  $S_p \in \mathbb{R}^{6 \times 6}$  is the projection matrix for selecting a specified demonstration plane.  $\mathbf{V}_c \in \mathbb{R}^6$  is the desired robot Cartesian velocity and  $\mathbf{F}_s \in \mathbb{R}^6$  for sensed wrench in  $\{E\}$ .  $F_z$  is the z-axis component of  $\mathbf{F}_s$ . A Butterworth low-pass filter is adopted to smooth wrench signal.  $V_n \in \mathbb{R}$  is the normal adjustment in  $\{E\}$  in order to keep a constant contact force  $F_{dn}$  in  $\{E\}$ .  $\mathbf{V}_F \in \mathbb{R}^6$  is generated by admittance controller.  $K_p^D, K_i^D \in \mathbb{R}$  are the proportional integral parameters of the PI controller.  $M_a^D, D_a^D \in \mathbb{R}_{6 \times 6}$  are the mass-damper parameters of the admittance controller, which are set by trials.

As the manipulator moves slowly, the effect of inertia force is ignored. The force analysis of end-effector is  $\mathbf{F}_e = \mathbf{F}_s + \mathbf{F}_h$ , where  $\mathbf{F}_h \in \mathbb{R}^6$  is the force that a human exerts on the end-effector and  $\mathbf{F}_e \in \mathbb{R}^6$  for reactive force from environment. The robot movement data are recorded versus time during



(a) Demonstration



(b) Replay

FIGURE 8: Demonstration of polishing. (a) Kinesthetic teaching for a trajectory. (b) Replay the teaching movement and record position and force data.

demonstration. Then the robot replays previous movement trajectory without human guidance. The force analysis of the end-effector is as  $\hat{\mathbf{F}}_e = \hat{\mathbf{F}}_s = \mathbf{F}_e$ . So we obtained the force profiles with human experience. Position and force were recorded as demonstration information.

Kinesthetic teaching is adopted based on control law of (11a), (11b), (11c), and (11d) and Figure 7. The parameter values for controllers are  $K_p^D = 2.5 \times 10^{-4}$ ,  $K_i^D = 1.9 \times 10^{-4}$ ,  $M_a^D = I_{6 \times 6} \times 10^4$ ,  $D_a^D = 5I_{6 \times 6} \times 10^3$ . To learn the profiles of interaction force from human polishing on the workpiece in Figure 6, we chose polishing path including 36 segments, which were generated by the projection matrix  $S_p$  in 36 vertical planes of equal spaced angle  $10^\circ$ . We demonstrated polishing trajectories with human guiding the polishing head as illustrated in Figure 8(a). Joint motion data were recorded. Then the robot replayed previous movement, shown in Figure 8(b), while normal contact force/torque  $F_p \in \mathbb{R}$ , position  $\mathbf{x}_p \in \mathbb{R}^3$ , and velocity  $\dot{\mathbf{x}}_p \in \mathbb{R}^3$  of the center point of polishing disc were recorded simultaneously at 125 Hz. Each path was demonstrated 5 times.

**4.1.2. Model Learning.** Before learning the model, the dataset was preprocessed by the DTW algorithm to align 5 demonstrations over time in same path. We acquired a training dataset  $\{\mathbf{x}_p^i, \dot{\mathbf{x}}_p^i, F_p^i\}_{i=1}^N$  composed of  $N$  datapoints. A task completion rate was defined to describe same path performing as  $\alpha_i = i/N$ ,  $\alpha_i \in [0, 1]$ . We chose  $\{\alpha_i\}_{i=1}^N$  for GMM input and  $\{F_p^i\}_{i=1}^N$  for output components in (5a). GMM and GMR were adopted to learn the internal correlation. The learning result of mapping  $\alpha \rightarrow \hat{\mathbf{F}}$  was shown in Figure 9. The standard deviation  $\sigma_{\hat{\mathbf{F}}}$  of contact force  $\hat{\mathbf{F}}$  conditional distribution was recorded as a function of  $\alpha$ . Noticed that there was a one-to-one correspondence between the task completion rate  $\alpha_i$  and position  $\mathbf{x}_p$  for same path demonstrations, we could obtain the mapping function  $f(x) = [\hat{v}, \hat{\mathbf{F}}, \sigma_{\hat{\mathbf{F}}}]$ ,  $\hat{v} \in \mathbb{R}^{1 \times N}$

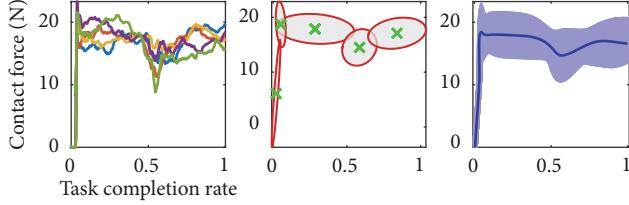


FIGURE 9: The GMM and GMR learning for one path polishing. Left: the demonstrations for the polishing path are repeated 5 times. Middle: GMM is trained to approximate the joint probability distribution  $p(\alpha, \hat{F})$  by EM algorithm using BIC for 5 Gaussian components. The green cross marks refer to the mean of each Gaussian model. Right: GMR policy is derived based on the GMM; the blue region is the the  $3\sigma$  confidence interval of the contact force distribution.

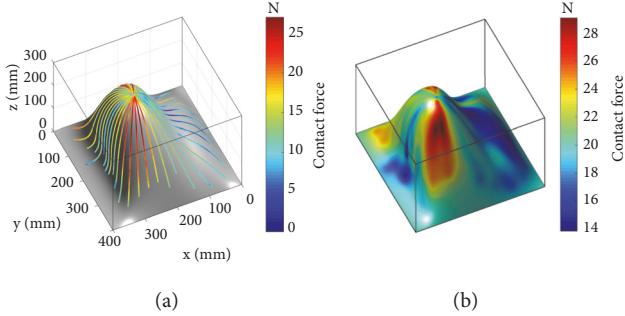


FIGURE 10: Mapping policy of  $x_p \rightarrow \hat{F}$ . (a) GMR result for 36 groups of mapping. The colors of these lines attached to the surface refer corresponding value of contact force  $\hat{F}$  along with the position  $x_p$ . (b) GP regression result for the whole surface mapping from position to contact force.

for the polishing speed. In the same way, there were 36 groups of mapping function  $F() : x \rightarrow \{\hat{v}_a, \hat{F}_a, \sigma_{\hat{F}_a}\}_{n=1}^{36}$ . The mapping policy of  $x \rightarrow \hat{F}_a$  was shown in Figure 10(a) for 36 trajectories. To get mapping policy of the whole surface, we performed GP regression [32] to solve this problem, as shown in Figure 10(b).

**4.1.3. Polishing Autonomous Executions.** In the task execution phase, we chose an arbitrary polishing trajectory in Figure 12(a) by an interactive path generation method, which was developed on the Visualization Toolkit (VTK) with the 3D model of the workpiece. It is also available for kinesthetic teaching to generate polishing path. Based on the position of this trajectory, parameters of  $\{\hat{v}_e, \hat{F}_d, \sigma_{\hat{F}_d}\}$  were predicted by the previous learned model. An adaptive hybrid force/position control law in Figure 5 was adopted to track this trajectory and generated contact force. The parameters of force controller were set as  $K_p^f = \text{diag}(1.6I_{3 \times 3} \times 10^{-4}, \mathbf{0}_{3 \times 3})$ ,  $K_d^f = \text{diag}(3I_{3 \times 3} \times 10^{-4}, \mathbf{0}_{3 \times 3})$ . The constraint selection matrix  $S^b = \text{diag}(0, 0, 1, 0, 0, 0)$  referred to normal constraint relative to the workpiece. The snapshots were illustrated in Figure 11, with force tracking result shown in Figure 12(b).



FIGURE 11: The snapshots for polishing execution with 10s interval.

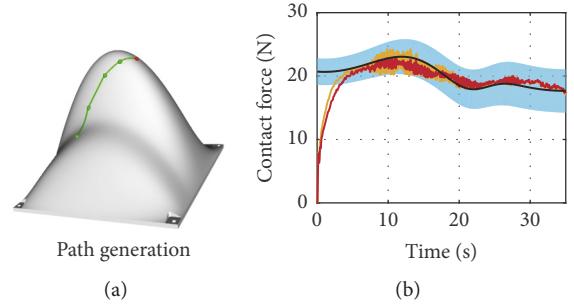


FIGURE 12: Trajectory generation and force tracking with adaptive hybrid force/position control. (a) Interactive trajectory generation for the surface polishing. (b) Contact force tracking using adaptive PI controller with 2 trials in different values of  $K_p^f$ . The blue region is the  $3\sigma$  interval of the contact force distribution and the black line is the desired force. The RMSE of the force tracking error is 1.58N for red curve.

We could evaluate the task performance by the force tracking result in Figure 12(b). The performance of task executions is evaluated by the proportion of real contact force in the confidence interval shown as the blue region. It accounts for 90.3% time proportion in the whole execution. The tracking result shows an excellent performance relative to the reference force distribution.

**4.2. Peg-In-Hole Assembly.** To evaluate the proposed framework, several experiments of peg-in-hole assembly were carried out on UR-3 arm. We designed a series of pegs and holes of stainless steel with different fit tolerances and dimensions, shown in Figure 13(a) and Table 1. Holes are with the same depth of 30 mm. Each peg is individually fixed on an elastic mechanism, which is a passive compliant device of rectilinear motion and mounted on the OptoForce F/T sensor as Figure 13(b).

**4.2.1. Collaborative Insertions for Data Collection.** Peg 1 and hole 1 were chosen for assembly demonstration group. We proposed a procedure for peg-in-hole assembly, which included three phases as Figures 13(b)–13(d). Firstly the peg moved towards the hole surface and made a constant contact force, with uniform distributed deviations of position  $\Delta x \in [-6 \text{ mm}, 6 \text{ mm}]$  and rotations in  $[-20^\circ, 20^\circ]$  along  $x$  and  $y$  axis. Then an Archimedean spiral movement was adopted to search the hole until the  $z$  axis value of the force sensor decreased suddenly or  $x$  and  $y$  axis force value was above a certain threshold. Finally, a demonstrator pressed the button

TABLE 1: Holes and pegs with different fit tolerances and dimensions.

	Hole 1	Hole 2	Hole 3	Hole 4	Hole 5	Hole 6	Peg 1	Peg 2	Peg 3
Tolerance	H7	F7	E7	D7	H7	H7	h6	h6	h6
Dimension mm(SI)	$\phi 10_0^{+0.015}$	$\phi 10_{-0.013}^{+0.028}$	$\phi 10_{-0.025}^{+0.04}$	$\phi 10_{-0.04}^{+0.055}$	$\phi 8_0^{+0.015}$	$\phi 12_0^{+0.018}$	$\phi 10_{-0.009}^0$	$\phi 8_{-0.009}^0$	$\phi 12_{-0.011}^0$

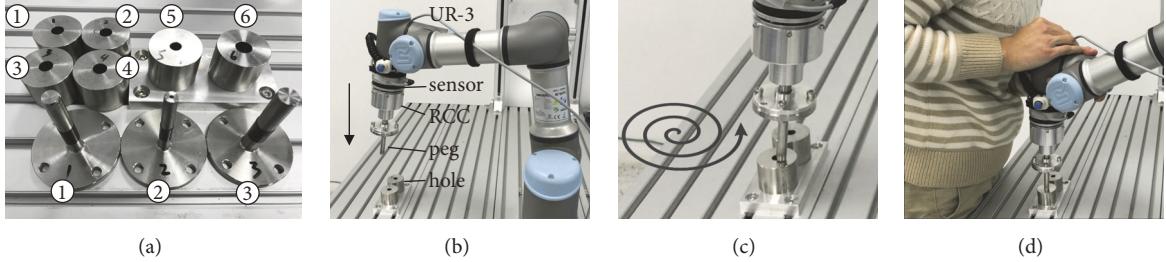


FIGURE 13: Experiment setup and demonstration phase by collaborative insertions. (a) The three pegs and six holes. (b) The peg is moving towards the hole. (c) Searching the hole by an Archimedean spiral movement. (d) Collaborative insertions.

of the end-effector to enable a free-drive mode of the UR arm, which was implemented by joint torque estimation and allowed the demonstrator to guide the peg into the hole, as shown in Figure 13(d).

In assembly tasks, the demonstrator's hands adjusted the position and orientation of the peg based on wrench feedback. We recorded sensed torque  $M = [M_x, M_y]^T$  along with the angular velocity  $\omega = [\omega_x, \omega_y]^T$  of peg with a sampling rate of 125 Hz during 20 groups of collaborative insertions.  $M_x, M_y \in \mathbb{R}^{1 \times N}$  and  $\omega_x, \omega_y \in \mathbb{R}^{1 \times N}$  were the sensed torque and angular velocity in Cartesian space, respectively.

**4.2.2. Adjustment Policy Learning.** GMM was trained with  $M \in \mathbb{R}^{2 \times N}$  input and  $\omega \in \mathbb{R}^{2 \times N}$  output in (5a) by EM algorithm. BIC was used for the optimal Gaussian components. Figure 15 showed the GMM distributions of  $\{M_x, \omega_x\}$ . The conditional distribution of the angular velocity was  $\widehat{\omega} \sim p(\omega | M)$  from (5a). Thus for a certain torque  $M$ , the angular velocity could be calculated for the robot movement. Using (7), we could acquire the GMR technique:  $\widehat{\omega} = E\{p(\omega | M)\}$ .

**4.2.3. Peg-In-Hole Autonomous Executions.** Adjustment for the position  $x$  and orientation  $R \in \mathbb{R}^{3 \times 3}$  of the peg is the key to insertion. The control scheme of Figure 5 is also adopted with F/T feedback  $F_c = [F_a, M_a]^T$ ,  $F_a \in \mathbb{R}^{3 \times 1}$  and  $M_a \in \mathbb{R}^{3 \times 1}$ . In addition, a torque controller is specially designed for orientation adjustment. The orientation  $R$  of the peg is calculated by the angular velocity  $\omega_a = [\omega_x, \omega_y, \omega_z]^T$  as

$$R(t + \Delta t) = (\Delta t S(\omega_a) + I_{3 \times 3}) R(t) \quad (12a)$$

with the skew-symmetric matrix

$$S(\omega_a) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (12b)$$

where  $\Delta t$  is the control time step 0.008s and  $I_{3 \times 3}$  for a unit matrix. To compare our approach of LfD with random searching that did not use human experience, we design four kinds of torque controllers as follows:

- (i) A: Angular velocity obeyed an uniform distribution  $\widehat{\omega} \sim \mathcal{U}(\mathbf{a}, \mathbf{b})$ , where  $\mathbf{a}$  is chosen to  $\mathbf{a} = \mathbf{0}_{2 \times 1}$  and  $\mathbf{b} = [\sigma_x, \sigma_y]^T$  is selected from the covariance matrix in the GMM.
- (ii) B: Gaussian distribution  $\widehat{\omega} \sim \mathcal{N}(\boldsymbol{\mu}_B, \boldsymbol{\Sigma}_B)$  with constant value of  $\boldsymbol{\mu}_B = [0, 0]^T$  and  $\boldsymbol{\Sigma}_B = [0.1, 0.1]^T$ .
- (iii) C: GMM for  $\widehat{\omega} \in p(\omega | M)$  based on sensed torque.
- (iv) D: GMR for  $\widehat{\omega} = E\{p(\omega | M)\}$ .

The autonomous executions are performed including four phases:

- (i) Moving towards the hole surface as Figure 14(a).
- (ii) Searching the hole as Figures 14(b) and 14(c) with constant contact force along  $z$  axis.
- (iii) Macro orientation adjustment. The torque controller is chosen as  $\widehat{\omega} = -3\Sigma_B \text{sign}(M)$  to minish the orientation error in a large scale until the contact torque is less than a threshold value  $M_0 = 0.1 \text{Nm}$ .
- (iv) Micro adjustment based four kinds of torque controllers in Figures 14(e)–14(g). A decreasing function  $\gamma(\alpha) = (1 - \alpha^2)e^{-\alpha^2}$  is designed to reduce the contact force based on the task completion rate  $\alpha$ . Then the reference angular velocity is written as  $\tilde{\omega} = |\gamma(\alpha)\widehat{\omega}| \text{sign}(M)$  with (12a) and (12b) adjusting orientation  $R$ .

It is noticed that the force controller is working in the third and fourth phases to achieve a compliant interaction. And in the fourth phase, the reference angular velocity  $\tilde{\omega}$  is the same positive or negative value with the sensed torque  $M$ . The position  $x$  and orientation  $R$  are controlled by the sensed force  $F_a$  and torque  $M_a$  respectively. Desired contact force is

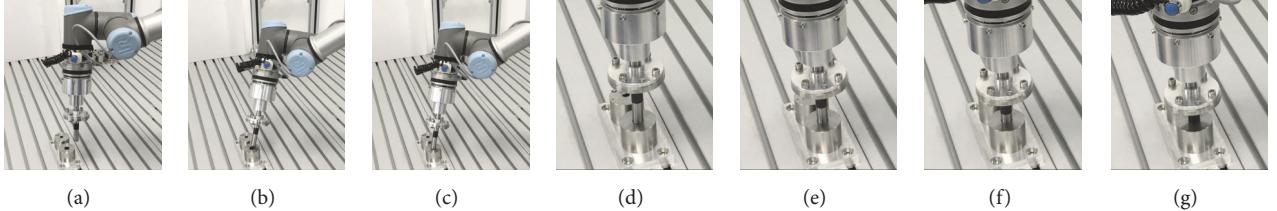


FIGURE 14: The snapshots for peg-in-hole execution. (a) Moving towards the hole. (b) and (c) Searching hole. (d)–(g) Adjustment based on sensed wrench.

TABLE 2: Result comparison of insertions for different controller.

	Rate of success (%)	Avg. completion time (s)	Norm of contact force mean (std.dev) (N)	Avg. energy consumption (J)
A	64	20.1	8.5 (4.5)	1847
B	56	30.1	7.0 (4.6)	2433
C	88	15.2	11.1 (6.2)	1483
D	40	38.5	4.1 (3.5)	2204

set constantly as  $F_d = [0, 0, F_z^0]^T$  to push the peg into the hole. The force controller is realized by a PI controller to diminish the contact force along  $xy$  axis and keep a constant force in the vertical direction, while torque controller is designed by four kinds of adjustment strategies. The parameters of the PI controller are set as  $K_p^a = \text{diag}(5, 5, 8) \times 10^{-5}$ ,  $k_i^a = \text{diag}(2, 2, 3) \times 10^{-5}$ ,  $S^a = \text{diag}(1, 1, 1, 1, 1, 1)$ .

Two sets of experiments were carried out to evaluate our approach. The first one was conducted for comparing the performance of the same peg and hole. The second one was set for validating the generalization capabilities of the proposed approach based on GMM learning with different fit tolerances and dimensions, using peg 1 for hole 1 to 4 and pegs 2, 3 for holes 5, 6 correspondingly. Each group was performed on 25 trials for equally spaced orientation at  $[-20^\circ, 20^\circ]$  and position error at  $[-6 \text{ mm}, 6 \text{ mm}]$ . Holes 1-4 share the same dimension and different tolerances, and peg 1 - hole 1, peg 2 - hole 5, peg 3 - hole 6 are with different dimensions and same tolerance fit.

The performance of peg-in-hole assembly can be evaluated by several aspects, the rate of success, the average of the task completion time, which means the adaption and efficiency of task executions, the norm of the contact force, and the average of energy consumption, which determines the compliant interaction quality.

The result for the first set of experiments is shown in Table 2 with hole 1 and peg 1. A phase diagram of autonomous execution is presented in Figure 15. The controller C is sampling from the conditional distribution  $\hat{\omega} \in p(\omega | M)$  based on learned model, while controllers A and B are sampling from static distributions. The controller C achieve a higher rate of success and a lower completion time than others, while the controller C is along with higher contact forces. It is acceptable for the improvement of assembly efficiency. The lower average of energy consumption is calculated by (10) and mainly due to the less completion time. The controller

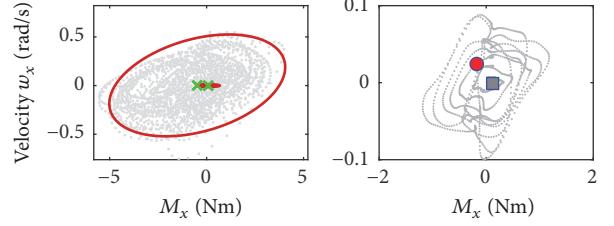


FIGURE 15: The GMM learning result and once peg-in-hole execution. Left: GMM for  $\{M_x, \omega_x\}$  in  $x$  axis. The gray sample points are 20 groups of data of collaborative insertions for the peg 1 and hole 1. The red ellipse refers to the  $3\sigma$  confidence interval of GMM. Green crosses are the center points of GMM. Right: phase diagram for once auto execution with contact torque and angular velocity in  $x$  axis, with the circle block for starting point and the square block for ending point.

D uses GMR for the real-time angular velocity calculating. It has the lowest rate of success and highest time. The main reason is that the fitting clearance of peg 1 and hole 1 is about 0.01 mm. Such small clearance can easily cause the assembly getting stuck. However, the controllers A, B, and C generate the angular velocity by random numbers, which may get rid of getting stuck in static conditions. Random disturbances contribute to precision assembling. Another reason may be on account of the inaccuracy of the sensed torque or errors in the learned model.

For the second set of experiments, the six groups of experiments show similar results in Table 3, all of which indicate a better performance relative to the group of the first set. Clearly, the approach learned from collaborative insertions of peg 1 and hole 1 can be well transferred to situations of different dimensions and tolerance fit.

**4.3. Discussion.** Two typical force-relevant experiments were set up to validate our learning framework successfully. Polishing autonomous executions were conducted according to

TABLE 3: Result comparison of insertions for different tolerance fit or dimensions.

Hole	Rate of success (%)	Avg. completion time (s)	Norm of contact force mean (std.dev) (N)	Avg. energy consumption (J)
1	88	15.2	11.0 (6.2)	1483
2	76	14.5	11.2 (6.3)	1413
3	80	15.4	10.8 (6.5)	1568
4	88	12.5	11.7 (6.4)	1336
5	88	14.1	11.1 (5.7)	1464
6	76	14.5	11.7 (6.1)	1531

learned model, with high force and position tracking performance. In peg-in-hole assembly experiments, adaptation and generalization capabilities were achieved for GMM learning.

While we have demonstrated the effectiveness with two experiments, there are still some limitations with current learning approach. Demonstrations for the polishing task of complex surface are a little tedious and time-consuming as a result of kinesthetic teaching and the replay. The back-drive teaching of assembly is also less natural than human direct teaching. In learning phase, we mainly use GMM to learn the correlation between the interaction force and the motion. Other learning algorithms may need to be compared with GMM. For the peg-in-hole assembly, execution time is still over time for industrial application. Parameters of controllers need to be improved in detail.

## 5. Conclusion

In this paper, a novel framework for learning force-relevant skills from human demonstration is proposed. The motion and force profiles during human demonstrations are recorded and learned as a statistical model to encode the force-relevant skill. Upon the learned model, a task planner is devised to offer the initial task policy. Furthermore, an adaptive force controller is proposed to adapt the motion of the robots according to the sensed force and the initial task policy. The proposed approach is demonstrated with two experiments (namely polishing and assembly) to showcase its effectiveness. In the future, we are planning to take more task constraints and sensor modalities into account.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 51705371), the Natural Science Foundation of Jiangsu Province (Grant No. BK20180235), and the Foundation of National Key

Laboratory of Human Factors Engineering (Grant No. 6142222180311).

## Supplementary Materials

The supplementary material is a video of the experiments, including an object polishing task and a peg-in-hole assembly task. Each task consists of three phases: demonstrations, learning, and executions. (*Supplementary Materials*)

## References

- [1] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*, Springer, 2016.
- [2] D. E. Whitney, “Quasi-static assembly of compliantly supported rigid parts,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 104, no. 1, pp. 65–77, 1982.
- [3] M. T. Mason, “Compliance and Force Control for Computer Controlled Manipulators,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.
- [4] N. Hogan, “Impedance control: an approach to manipulation: Part II—implementation,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 8–16, 1985.
- [5] S. Gu, E. Holly, T. Lillicrap, and S. Levine, “Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates,” in *Proceedings of the IEEE International Conference on Robotics and Automation, (ICRA)*, pp. 3389–3396, IEEE, 2017.
- [6] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, “Learning Hand-Eye Coordination for Robotic Grasping with Large-Scale Data Collection,” in *International Symposium on Experimental Robotics*, vol. 1 of *Springer Proceedings in Advanced Robotics*, pp. 173–184, Springer, 2016.
- [7] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare et al., “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [8] M. Kalakrishnan, L. Righetti, P. Pastor, and S. Schaal, “Learning force control policies for compliant manipulation,” in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems: Celebrating 50 Years of Robotics (IROS)*, pp. 4639–4644, IEEE, 2011.
- [9] A. X. Lee, H. Lu, A. Gupta, S. Levine, and P. Abbeel, “Learning force-based manipulation of deformable objects from multiple demonstrations,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 177–184, IEEE, 2015.
- [10] M. Denisa, A. Gams, A. Ude, and T. Petric, “Learning compliant movement primitives through demonstration and statistical

- generalization,” *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 5, pp. 2581–2594, 2016.
- [11] S. M. Khansari-Zadeh and A. Billard, “Learning stable nonlinear dynamical systems with Gaussian mixture models,” *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.
- [12] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, “Robot programming by demonstration,” in *Springer handbook of robotics*, pp. 1371–1394, Springer, 2008.
- [13] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, “Dynamical movement primitives: Learning attractor models for motor behaviors,” *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.
- [14] M. Li, H. Yin, K. Tahara, and A. Billard, “Learning object-level impedance control for robust grasping and dexterous manipulation,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6784–6791, IEEE, 2014.
- [15] M. Li, Y. Bekiroglu, D. Krasic, and A. Billard, “Learning of grasp adaptation through experience and tactile sensing,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3339–3346, IEEE, 2014.
- [16] K. Kronander and A. Billard, “Online learning of varying stiffness through physical human-robot interaction,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1842–1849, IEEE, 2012.
- [17] M. Li, K. Tahara, and A. Billard, “Learning task manifolds for constrained object manipulation,” *Autonomous Robots*, vol. 42, no. 1, pp. 159–174, 2018.
- [18] C. Yang, C. Zeng, P. Liang, Z. Li, R. Li, and C.-Y. Su, “Interface design of a physical human-robot interaction system for human impedance adaptive skill transfer,” *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 329–340, 2018.
- [19] C. Yang, C. Zeng, C. Fang, W. He, and Z. Li, “A DMPs-based Framework for Robot Learning and Generalization of Human-like Variable Impedance Skills,” *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1193–1203, 2018.
- [20] B. Siciliano and L. Villani, *Robot Force Control*, vol. 540, Springer Science & Business Media, 2012.
- [21] Y. Lin, S. Ren, M. Clevenger, and Y. Sun, “Learning grasping force from demonstration,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1526–1531, IEEE.
- [22] P. Kormushev, S. Calinon, and D. G. Caldwell, “Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input,” *Advanced Robotics*, vol. 25, no. 5, pp. 581–603, 2011.
- [23] L. Rozo, P. Jiménez, and C. Torras, “A robot learning from demonstration framework to perform force-based manipulation tasks,” *Intelligent Service Robotics*, vol. 6, no. 1, pp. 33–51, 2013.
- [24] B. Huang, M. Li, R. L. De Souza, J. J. Bryson, and A. Billard, “A modular approach to learning manipulation strategies from human demonstration,” *Autonomous Robots*, vol. 40, no. 5, pp. 903–927, 2016.
- [25] D. Vogt, S. Stepputtis, S. Grehl, B. Jung, and H. Ben Amor, “A system for learning continuous human-robot interactions from human-human demonstrations,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2882–2889, IEEE, 2017.
- [26] M. Khansari, E. Klingbeil, and O. Khatib, “Adaptive human-inspired compliant contact primitives to perform surface-surface contact under uncertainty,” *International Journal of Robotics Research*, vol. 35, no. 13, pp. 1651–1675, 2016.
- [27] K. Kronander, E. Burdet, and A. Billard, “Task transfer via collaborative manipulation for insertion assembly,” in *Workshop on Human-Robot Interaction for Industrial Manufacturing Robotics, Science and Systems*, Citeseer, 2014.
- [28] X. Li, G. Chi, S. Vidas, and C. C. Cheah, “Human-Guided Robotic Comanipulation: Two Illustrative Scenarios,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1751–1763, 2016.
- [29] K. Dautenhahn and C. L. Nehaniv, “The correspondence problem,” in *Imitation in Animals and Artifacts*, The MIT Press, 2002.
- [30] S. Calinon, *Robot programming by demonstration*, EPFL Press, 2009.
- [31] T. T. Andersen, *Optimizing the universal robots ros driver*, Technical University of Denmark, Department of Electrical Engineering, 2015.
- [32] C. E. Rasmussen and H. Nickisch, “Gaussian processes for machine learning (GPML) toolbox,” *Journal of Machine Learning Research*, vol. 11, pp. 3011–3015, 2010.

