

A robotic shared control teleoperation method based on learning from demonstrations

International Journal of Advanced
Robotic Systems
July-August 2019: 1–13
© The Author(s) 2019
DOI: 10.1177/1729881419857428
journals.sagepub.com/home/arx



Bao Xi^{1,2} , Shuo Wang^{1,2,3}, Xuemei Ye¹, Yinghao Cai¹,
Tao Lu¹ and Rui Wang¹

Abstract

In teleoperation, the operator is often required to command the motion of the remote robot and monitor its behavior. However, such an interaction demands a heavy workload from a human operator when facing with complex tasks and dynamic environments. In this article, we propose a shared control method to assist the operator in the manipulation tasks to reduce the workload and improve the efficiency. We adopt a task-parameterized hidden semi-Markov model to learn a manipulation skill from several human demonstrations. We utilize the learned model to predict the manipulation target given the current observed robotic motion trajectory and subsequently estimate the desired robotic motion given the current input of the operator. The estimated robotic motion is then utilized to correct the input of the operator to provide manipulation assistance. In addition, a set of virtual reality devices are used to capture the operator's motion and display the vision feedback from the remote site. We evaluate our approach through two manipulation tasks with a dual-arm robot. The experimental results show the effectiveness of the proposed method.

Keywords

Dual arm, hidden semi-Markov model, goal prediction, operation assistance, virtual reality

Date received: 18 July 2018; accepted: 2 April 2019

Topic: AI in Robotics; Human Robot/Machine Interaction

Topic Editor: Andrey V Savkin

Associate Editor: Jawhar Ghommam

Introduction

Recently, advancements in robot learning theories, such as deep reinforcement learning^{1,2} and imitation learning,³ enable robots carry out many tasks independently, but human intervention is still indispensable for many complex tasks, such as handling hazardous materials, underwater and space exploration, minimally invasive surgery, and so on. In these cases, teleoperation that combines human manipulation capabilities and the robot is required. In conventional teleoperation systems, a human operator is directly required to control the remote robot in detail while monitoring its behavior, which results in a heavy workload.⁴ Some bimanual tasks, such as peg-in-hole, require two arms to act cooperatively. However, teleoperating a dual-arm robot to accomplish such cooperative tasks is a complicated skill.

In such scenarios, shared control methods can reduce the workload and improve operation efficiency by combining manual teleoperation with autonomous assistance.

¹ State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China

² University of Chinese Academy of Sciences, Beijing, China

³ CAS Center for Excellence in Brain Science and Intelligence Technology, Shanghai, China

Corresponding author:

Bao Xi, State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China; University of Chinese Academy of Sciences, Beijing 101408, China.

Email: xibao2016@ia.ac.cn



Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License

(<http://www.creativecommons.org/licenses/by/4.0/>) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (<https://us.sagepub.com/en-us/nam/open-access-at-sage>).

Intensive research studies in developmental psychology indicate that human actions are often goal-directed.^{5,6} Accordingly, in order to assist the operator in the manipulation tasks, the robot needs to first predict the manipulation goal and then provides assistance based on the predictions. For example, in reaching-and-grasping task, which object to reach and how to reach are essential for providing manipulation assistance. To do the predictions while performing the task, a robotic motion model is needed. Additionally, since the type of teleoperation tasks may vary frequently, the robotic motion model for a specific manipulation task is required to be built efficiently and rapidly. Among the robotic learning algorithms, learning from demonstrations (LfD) enables a robot to be programmed by simply showing it how to perform a desired task, which is efficient and intuitive.

In this work, we propose a shared control method based on task-parameterized hidden semi-Markov model (TP-HSMM).⁷ The task parameters here refer to the variables that describe the manipulation context, such as the positions of some task-related objects, which can be used to reshape the robotic movement. For bimanual tasks, we set the pose of a robotic arm as a task parameter to model the cooperation behaviors between two arms. The demonstrations used to train TP-HSMM can be obtained from direct teleoperation. During teleoperation, our approach evaluates the probability distribution over the possible targets given the current observed robotic motion trajectory based on the learned robotic motion model. The object with the highest probability is assumed to be the manipulation target. To provide operation assistance, many shared control works blend operator input with some autonomous policy, which selects assistance actions independent of the operator input.⁸ In this work, the desired robotic movement is estimated based on the current operator input and most likely target. The desired robotic movement is then used as an assistance action to blend with the operator input. Moreover, the virtual reality (VR) techniques are adopted in our teleoperation system to give the operator a better operation experience. Concretely, we utilize a pair of VR controllers to capture the motion of the operator and display the vision feedback from the remote site in the headset to improve the transparency of the teleoperation system.

The contributions of the proposed method are as follows: (i) we adopt TP-HSMM to learn a robotic manipulation skill so that our shared control method can adapt to new tasks rapidly. (ii) The assistance action is dependent on the current operator input and manipulation target. (iii) VR techniques are utilized as the human–robot interface to improve the transparency of the teleoperation system. We evaluate the approach on a reaching task and a bimanual peg-in-hole task. The experiments reveal that our method can provide the operator useful assistance and improve the operational efficiency.

The remainder of this article is organized as follows. The second section reviews the related work. The proposed

VR-based dual-arm teleoperation system, the goal prediction method, and manipulation assistance method based on TP-HSMM are described in the third section. The fourth section presents the experimental settings and results. Conclusions and future work are presented in the fifth section.

Related works

In recent years, the research studies on the field of robotic teleoperation have obtained fruitful achievements, such as adopting new human–machine interfaces and introducing autonomy to assist the operators. These advancements intend to provide the operators more natural and efficient means to accomplish teleoperation tasks.

The human–robot interface is a vital part in teleoperation, which directly impacts the operation experience of the user. The currently widely used motion capture sensors, such as exoskeletal mechanical devices,⁹ data glove with angle sensors,¹⁰ and inertial motion tracking sensors,¹¹ have the merits of accurate detection and stable performance but are invasive and hindrance to natural human motions. Vision-based techniques^{12–14} do not require operator to wear extra devices; however, the problems of object occlusion and loss of tracking result in bad user experience. VR interfaces offer intuitive means for mapping a user's motion actions to robot. In the work of Lipton et al.¹⁵ and Zhang et al.,¹⁶ VR devices were utilized to directly control a humanoid robot. In this work, we also built our teleoperation system using current consumer VR hardware, but we adopt a shared control strategy to provide assistance in manipulation tasks.

Recently, many shared control methods have been proposed to reduce the workload of the operators by improving the autonomy in teleoperation systems.^{17–20} Some researchers investigate to assist the operator with some force feedback, such as haptic guidance to a target position or desired trajectory.^{21,22} Our work adopts a predict-then-blend framework,¹⁷ which predicts the most likely goal of the operators and then provides assistance in the manipulation tasks for that goal.

There are a large number of works on predicting the goal of the user. Narayanan et al.¹⁸ proposed an approach to predict the short-term goal of the user based on the distance between the current and the goal configuration. This method is intuitive but very crude, since it does not take the history information into account. In the work of Javdani et al.¹⁷ and Dragan and Srinivasa,¹⁹ the prediction problem is formulated based on maximum entropy inverse reinforcement learning (MaxEnt IRL).^{23–25} In MaxEnt IRL, the user's input is assumed to be noisy that optimizes a goal-dependent cost function, which is relevant to the historic trajectory of the user. However, learning a robotic motion skill by IRL is time-consuming. In our work, we adopt an efficient LfD method, namely TP-HSMM, as the motion planner to generate goal-dependent trajectory. Moreover, in TP-HSMM, the state duration is modeled explicitly,

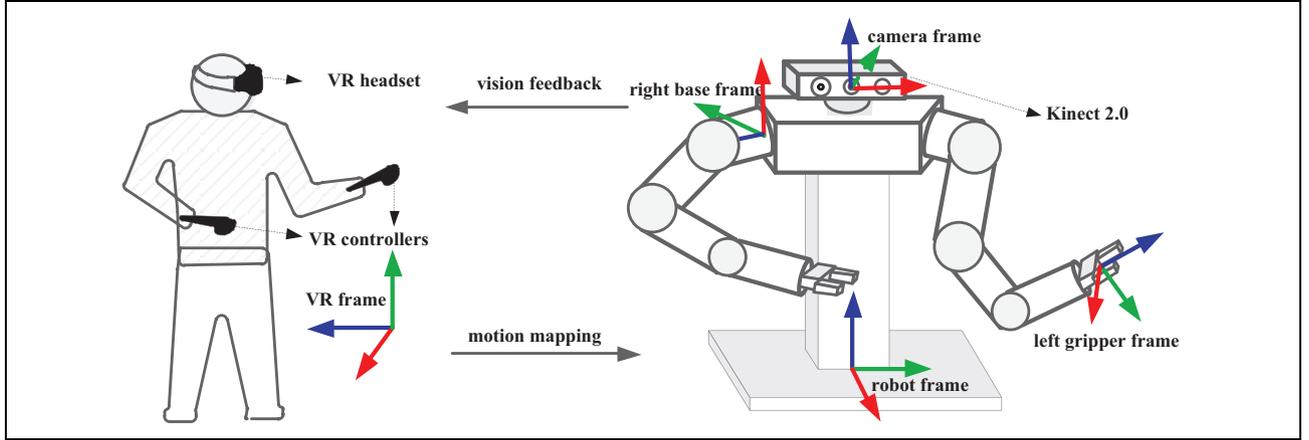


Figure 1. Teleoperation diagram. The VR controllers capture the motion of operator’s hands and the VR headset shows the vision feedback from the Kinect mounted on the robot. The coordination systems of VR devices, dual arm robot, and Kinect are used in motion mapping from operator to robot. VR: virtual reality.

which is beneficial for resisting temporal perturbations and representing the time constraint of the tasks.⁷

There are also a few works which address the prediction problem by utilizing other machine learning methods. Koppula and Saxena²⁶ proposed to use anticipatory temporal conditional random field to predict human motions. Martinez et al.²⁷ utilized a recurrent neural network to model human motion to do short-term prediction.

After predicting the goal, shared control teleoperation system provides the user operation assistances for that goal. There are many approaches to offer the assistances. El-Hussieny et al.²⁸ proposed a teleoperation system, which turns to autonomous mode if the confidence of the prediction is high. However, in some tasks, human intervention is indispensable, even though the prediction confidence is high. Rosenberg proposed a method called virtual fixtures²⁹ which can be seen as a ruler so that the user can teleoperate a robot to achieve some precise motions.^{30,31} Therefore, virtual fixture-based methods are usually used to guide the user along a predefined path. Our work is similar to Tanwani and Calinon.³² We advance it in two directions. First, we blend the desired robotic motion with the input of the operator so that they can complement each other, which can avoid manipulation failure led by bad estimation or improper input. Second, we model the cooperation behaviors between two arms so that the operator can obtain assistance in bimanual manipulation tasks.

Proposed approach

In this work, the technology of motion mapping between operator and robot is divided into position mapping and orientation mapping. The basic idea of position mapping is that the position displacements of the user’s hands correspond to those of the robot’s end-effectors. In order to make the teleoperation more intuitive, the orientations of the user’s hands are mapped to the robot directly.

We improve the autonomy in teleoperation system to assist the human operator in performing tasks. TP-HSMM is used to encode the human demonstrations observed from multiple task-parameterized frames. The parameters of HSMM are estimated using an expectation maximization (EM) algorithm.³³ The learned HSMM is used as a motion planner to evaluate probability distribution over the possible targets. To avoid improper assistance, a prediction confidence is defined based on the entropy of the evaluated probability distribution. When the confidence is higher than a predefined threshold, the teleoperation system switches to assist mode from direct mode. To assist operator, we estimate the intended robotic movement and blend it with the input of the operator to teleoperate the remote robot.

Motion mapping in direct teleoperation

We build the direct teleoperation system based on a set of VR devices, a dual-arm robot, and a Microsoft Kinect V2.0. During teleoperation, we map the poses of the operator’s hands from the VR frame into the robot frame to get the target poses of the robotic arms. And then the target poses are sent to the robot through network.

Since the dual-arm robot is right-and-left symmetrical, we shall discuss only the right side for the sake of brevity. Some of the used frames of reference in this work are illustrated in Figure 1. The pose of the right robotic arm in the right base frame at each time step is denoted as $\Phi^r = (\mathbf{p}^r, \mathbf{O}^r)$, where $\mathbf{p}^r \in \mathbb{R}^3$ and $\mathbf{O}^r \in \mathbb{R}^{3 \times 3}$ represent the position and the orientation of the right end-effector, respectively. Similarly, the pose of user’s right hand in the VR frame is denoted as $\Phi^h = (\mathbf{p}^h, \mathbf{O}^h)$. The initial pose of the right robotic arm and user’s hand is denoted as $\Phi_0^r = (\mathbf{p}_0^r, \mathbf{O}_0^r)$ and $\Phi_0^h = (\mathbf{p}_0^h, \mathbf{O}_0^h)$, respectively. The superscript r and h represent robot and operator’s hand, respectively.

As is illustrated in Figure 1, in the VR frame, the x -axis is pointed right (red arrow), the y -axis is pointed upward (green arrow), and the z -axis is pointed backward (blue arrow). In the robot frame, the x -axis is pointed forward; the y -axis and the z -axis are pointed left and upward, respectively. In teleoperation systems, the translation between these two reference frames is not a concern. The rotation matrix between the robot frame and the VR frame is

$$\mathbf{R}_V^r = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (1)$$

Then, we convert the orientation of the operator's hand from the VR frame to the robot frame by \mathbf{R}_V^r

$$\mathbf{O}_r^h = \mathbf{R}_V^r \cdot \mathbf{O}^h \cdot (\mathbf{R}_V^r)^{-1} \quad (2)$$

where \mathbf{O}_r^h denotes the rotation of the hand in the robot frame.

The displacement of the operator's hand is transformed to the robot frame as

$$\mathbf{d}_r^h = \mathbf{R}_V^r \cdot (\mathbf{p}^h - \mathbf{p}_0^h) \quad (3)$$

Finally, the target position and orientation of right end-effector in the right base frame is

$$\mathbf{p}_b^r = \mathbf{p}_0^r + \gamma(\mathbf{R}_r^b)^{-1} \cdot \mathbf{d}_r^h \quad (4)$$

$$\mathbf{O}_b^r = (\mathbf{R}_r^b)^{-1} \cdot \mathbf{O}_r^h \quad (5)$$

where \mathbf{R}_r^b denotes the rotation from the robot frame to the right base frame and γ is a scale coefficient so that the positional displacement is appropriately scaled. Then, we send the target pose $(\mathbf{p}_b^r, \mathbf{O}_b^r)$ to the controller of the right robotic arm via network. The target pose of the left robotic arm is calculated in a similar way.

Task-parameterized HSMM

Hidden markov model (HMM) is composed of several discrete states, which emit observations drawn from a set of probability density functions. We adopt a multivariate Gaussian distribution to represent the emission probability distribution of each state. In LfD, HMM segments the demonstrations into a set of movement primitives, which correspond to the hidden states.

The parameter of an HMM with K hidden states is described by $\Theta = \left\{ \{a_{i,j}\}_{j=1}^K, \Pi_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i \right\}_{i=1}^K$, where $a_{i,j}$ represents the probability to transit from state q_i to state q_j , Π_i is the initial state prior, and $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}$ are the center and covariance matrix of the Gaussian emission probability of the i th state. In robotic applications, HMM models the duration of each state roughly as a geometric distribution by a state self-transition probability. HSMM extends HMM

by modeling the state duration explicitly as a state-dependent stochastic variable. Here, the duration is modeled by a univariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_i^D, \boldsymbol{\Sigma}_i^D)$. Therefore, the parameters of an HSMM are described by

$$\Theta = \left\{ \{a_{i,j}\}_{j=1}^K, \Pi_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \boldsymbol{\mu}_i^D, \boldsymbol{\Sigma}_i^D \right\}_{i=1}^K.$$

Task parameters here are the positions and orientations of the reference frames, which are attached on the task-relevant objects. The task-parameterized models learn to describe a movement from different frames of reference. Therefore, a TP-HSMM with K states and P frames of reference are

$$\Theta = \left\{ \{a_{i,j}\}_{j=1}^K, \Pi_i, \left\{ \boldsymbol{\mu}_i^{(j)}, \boldsymbol{\Sigma}_i^{(j)} \right\}_{j=1}^P, \boldsymbol{\mu}_i^D, \boldsymbol{\Sigma}_i^D \right\}_{i=1}^K.$$

The demonstrations used to train TP-HSMM can be obtained by direct teleoperation or kinesthetic teaching.

In order to learn the relation between the robot's movement $\left\{ \boldsymbol{\xi}_t^I \right\}_{t=1}^T$ and the environment, we represent the observation

as $\boldsymbol{\xi}_t^{(j)} = \left[\boldsymbol{\xi}_t^{IT} \boldsymbol{\xi}_t^{OjT} \right]^T$, where $\boldsymbol{\xi}_t^{IT}$ and $\boldsymbol{\xi}_t^{Oj}$ are observed from right base frame and j th task frame, respectively. The right base frame is located at the base of the right robotic arm as shown in Figure 1. To get the observation, $\boldsymbol{\xi}_t^{(j)}$, we denote the task parameters with P reference frames as $\{\mathbf{A}_j, \mathbf{b}_j\}_{j=1}^P$

$$\mathbf{A}_j = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_j \end{bmatrix}, \quad \mathbf{b}_j = \begin{bmatrix} \mathbf{0} \\ \mathbf{p}_j \end{bmatrix} \quad (6)$$

where $\mathbf{I} \in \mathbb{R}^3$, $\mathbf{R}_j \in \mathbb{R}^{3 \times 3}$, and $\mathbf{p}_j \in \mathbb{R}^3$. \mathbf{R}_j and \mathbf{p}_j denote the relative rotation matrix and translation vector between the j th reference frame and the right base frame, and \mathbf{I} is a identity matrix. Then, the observation $\boldsymbol{\xi}_t^{(j)}$ is obtained by

$$\boldsymbol{\xi}_t^{(j)} = \mathbf{A}_j^{-1}(\boldsymbol{\xi}_t - \mathbf{b}_j) \quad (7)$$

where $\boldsymbol{\xi}_t = \left[\boldsymbol{\xi}_t^{IT} \boldsymbol{\xi}_t^{OT} \right]^T$. The observation forms a third-order tensor data set $\left\{ \boldsymbol{\xi}_t^{(j)} \right\}_{t,j=1}^{T,P}$, where T is the length of the demonstration. The parameters of TP-HSMM are obtained using EM algorithm.

Target prediction

In this section, we describe our approach to predict the manipulation target based on the learned TP-HSMM. In what follows, S denotes the initial position of the robot's end-effector, C denotes the current position, G denotes a potential target, and \mathcal{G} denotes a set of possible targets in the manipulation environment.

To adapt to a new environment, the parameters of the i th state of the learned TP-HSMM in the j th reference frame $\left\{ \boldsymbol{\mu}_i^{(j)}, \boldsymbol{\Sigma}_i^{(j)} \right\}$ are transformed to the right base frame according to the task parameter of the j th reference frame $\{\tilde{\mathbf{A}}_j, \tilde{\mathbf{b}}_j\}$

$$\mathcal{N}\left(\tilde{\boldsymbol{\mu}}_i^{(j)}, \tilde{\boldsymbol{\Sigma}}_i^{(j)}\right) = \mathcal{N}\left(\tilde{\mathbf{A}}_j \boldsymbol{\mu}_i^{(j)} + \tilde{\mathbf{b}}, \tilde{\mathbf{A}}_j \boldsymbol{\Sigma}_i^{(j)} \tilde{\mathbf{A}}_j^T\right) \quad (8)$$

The center and covariance $\tilde{\boldsymbol{\mu}}_i^{(j)}$ and $\tilde{\boldsymbol{\Sigma}}_i^{(j)}$ can be represented in block form

$$\tilde{\boldsymbol{\mu}}_i^{(j)} = \begin{bmatrix} \tilde{\boldsymbol{\mu}}_i^{\mathcal{I}j} \\ \tilde{\boldsymbol{\mu}}_i^{\mathcal{O}j} \end{bmatrix}, \tilde{\boldsymbol{\Sigma}}_i^{(j)} = \begin{bmatrix} \tilde{\boldsymbol{\Sigma}}_i^{\mathcal{I}j} & \tilde{\boldsymbol{\Sigma}}_i^{\mathcal{I}\mathcal{O}j} \\ \tilde{\boldsymbol{\Sigma}}_i^{\mathcal{O}\mathcal{I}j} & \tilde{\boldsymbol{\Sigma}}_i^{\mathcal{O}j} \end{bmatrix} \quad (9)$$

According to the block form of \mathbf{A}_j and \mathbf{b}_j described in the last section, the top-left block of \mathbf{A}_j is always \mathbf{I} , and the corresponding block of \mathbf{b}_j is $\mathbf{0}$. Therefore, the component $\{\tilde{\boldsymbol{\mu}}_i^{\mathcal{I}j}, \tilde{\boldsymbol{\Sigma}}_i^{\mathcal{I}j}\}$ is the same for different j values. For the sake of brevity, we present $\{\tilde{\boldsymbol{\mu}}_i^{\mathcal{I}j}, \tilde{\boldsymbol{\Sigma}}_i^{\mathcal{I}j}\}$ as $\{\tilde{\boldsymbol{\mu}}_i^{\mathcal{I}}, \tilde{\boldsymbol{\Sigma}}_i^{\mathcal{I}}\}$.

Then, we can synthesize the transformed model from the P task-parameterized frames and the right base frame by computing the product of Gaussians

$$\mathcal{N}(\tilde{\boldsymbol{\mu}}_i, \tilde{\boldsymbol{\Sigma}}_i) \propto \mathcal{N}(\tilde{\boldsymbol{\mu}}_i^{\mathcal{I}}, \tilde{\boldsymbol{\Sigma}}_i^{\mathcal{I}}) \prod_{j=1}^P \mathcal{N}(\tilde{\boldsymbol{\mu}}_i^{\mathcal{O}j}, \tilde{\boldsymbol{\Sigma}}_i^{\mathcal{O}j}) \quad (10)$$

where

$$\tilde{\boldsymbol{\Sigma}}_i = \left[\left(\tilde{\boldsymbol{\Sigma}}_i^{\mathcal{I}} \right)^{-1} + \sum_{j=1}^P \left(\tilde{\boldsymbol{\Sigma}}_i^{\mathcal{O}j} \right)^{-1} \right]^{-1} \quad (11)$$

$$\tilde{\boldsymbol{\mu}}_i = \tilde{\boldsymbol{\Sigma}}_i \left\{ \left(\tilde{\boldsymbol{\Sigma}}_i^{\mathcal{I}} \right)^{-1} \tilde{\boldsymbol{\mu}}_i^{\mathcal{I}} + \sum_{j=1}^P \left[\left(\tilde{\boldsymbol{\Sigma}}_i^{\mathcal{O}j} \right)^{-1} \tilde{\boldsymbol{\mu}}_i^{\mathcal{O}j} \right] \right\}$$

This product model can plan the motion of the robot given the current observation $\boldsymbol{\xi}_t^{\mathcal{I}}$. Firstly, it has to recognize the most likely state at present by

$$s_0 = \arg \max_i \frac{\pi_i \mathcal{N}\left(\boldsymbol{\xi}_t^{\mathcal{I}} | \tilde{\boldsymbol{\mu}}_i, \tilde{\boldsymbol{\Sigma}}_i\right)}{\sum_{k=1}^K \pi_k \mathcal{N}\left(\boldsymbol{\xi}_t^{\mathcal{I}} | \tilde{\boldsymbol{\mu}}_k, \tilde{\boldsymbol{\Sigma}}_k\right)} \quad (12)$$

Secondly, a state sequence is generated by the means of forward variable $\alpha_{t,i} = P(i, \boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_t | \boldsymbol{\theta})$

$$s_t = \arg \max_i \alpha_{t,i} \quad (13)$$

The state sequence is then used to retrieve a stepwise reference trajectory $\mathcal{N}(\hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\Sigma}}_t)$

$$\hat{\boldsymbol{\mu}}_t = \tilde{\boldsymbol{\mu}}_{s_t}^{\mathcal{O}}, \hat{\boldsymbol{\Sigma}}_t = \tilde{\boldsymbol{\Sigma}}_{s_t}^{\mathcal{O}} \quad (14)$$

The reference trajectory is tracked by a finite horizon discrete-time linear quadratic regulator to generate an executable and smooth trajectory.

In this work, the observation sequence of robotic motion from initial position S to current position C is denoted as $\boldsymbol{\xi}_{S \rightarrow C}$. The problem of predicting the target G^* given $\boldsymbol{\xi}_{S \rightarrow C}$ is formulated as maximizing a posterior probability

$$G^* = \arg \max_{G \in \mathcal{G}} P(G | \boldsymbol{\xi}_{S \rightarrow C}, \boldsymbol{\theta}) \quad (15)$$

where $\boldsymbol{\theta}$ is the parameters of TP-HSMM. Rewriting equation (15) according to the Bayes' theorem

$$G^* = \arg \max_{G \in \mathcal{G}} P(G | \boldsymbol{\xi}_{S \rightarrow C}, \boldsymbol{\theta}) \\ \propto \arg \max_{G \in \mathcal{G}} P(\boldsymbol{\xi}_{S \rightarrow C} | G, \boldsymbol{\theta}) P(G) \quad (16)$$

The learned TP-HSMM is utilized to plan the robotic motion from S to G , denoted as $\bar{\boldsymbol{\xi}}_{S \rightarrow G}$. Assume the closest point to C from $\bar{\boldsymbol{\xi}}_{S \rightarrow G}$ is D . Dynamic time wrap is then adopted to compare the observation sequence $\boldsymbol{\xi}_{S \rightarrow C}$ and the planned sequence $\bar{\boldsymbol{\xi}}_{S \rightarrow D}$ to get the distance $D(\boldsymbol{\xi}_{S \rightarrow C}, \bar{\boldsymbol{\xi}}_{S \rightarrow D})$, which is a target-dependent measurement. Then, the similarity between $\boldsymbol{\xi}_{S \rightarrow C}$ and $\bar{\boldsymbol{\xi}}_{S \rightarrow D}$ is defined as

$$C_G(\boldsymbol{\xi}_{S \rightarrow C}) = -D(\boldsymbol{\xi}_{S \rightarrow C}, \bar{\boldsymbol{\xi}}_{S \rightarrow D}) \quad (17)$$

Based on the principle of maximum entropy, we induce the likelihood in equation (16) by the similarity

$$P(\boldsymbol{\xi}_{S \rightarrow C} | G, \boldsymbol{\theta}) \approx \frac{e^{\beta C_G(\boldsymbol{\xi}_{S \rightarrow C})}}{\sum_{g \in \mathcal{G}} e^{\beta C_g(\boldsymbol{\xi}_{S \rightarrow C})}} \quad (18)$$

where β is an adjustable coefficient that increases the importance of similarity on the distribution. Substituting equation (18) into equation (16), the prediction becomes

$$G^* = \arg \max_{G \in \mathcal{G}} \frac{e^{\beta C_G(\boldsymbol{\xi}_{S \rightarrow C})}}{\sum_{g \in \mathcal{G}} e^{\beta C_g(\boldsymbol{\xi}_{S \rightarrow C})}} P(G) \quad (19)$$

In the preceding, S is the initial position, in fact, it can be any point in the observation sequence. Before making predictions, the robot is teleoperated directly at the beginning to get some observations $\boldsymbol{\xi}_{S \rightarrow C}$. The prior probability of each possible target $P(G)$ reflects the relevant environment information, such as task type, object affordance, and so on.

Many previous works studied the impact of autonomy on the performance of teleoperation and analyzed various influencing factors. Among all of the factors, prediction correctness has a substantial effect on the operation performance and user experience. Thus, the assistance should take into account how good the prediction is. In this work, a prediction confidence is defined based on the entropy of the prediction to measure the quality or correctness of the prediction

$$\text{conf} = \sum_{G \in \mathcal{G}} P(\boldsymbol{\xi}_{S \rightarrow C} | G, \boldsymbol{\theta}) \log P(\boldsymbol{\xi}_{S \rightarrow C} | G, \boldsymbol{\theta}) \quad (20)$$

At the beginning of the operation, since there is little information for the robot to predict the user's intent, the prediction confidence is usually low. As the manipulator expresses apparent tendency toward a target in the course of teleoperation, the prediction confidence increases. In order to avoid bad assistance based on the incorrect

predictions, we define a switching threshold to decide when to assist the operator

$$\text{control mode} = \begin{cases} \text{direct mode,} & \text{conf} < \mathcal{T} \\ \text{assist mode,} & \text{conf} \geq \mathcal{T} \end{cases} \quad (21)$$

where \mathcal{T} is the switching threshold. When the confidence is lower than the switching threshold, the operator controls the robot directly, otherwise the system switches from direct mode to assist mode.

Operation assistance

In the direct mode, the position and orientation of the operator hand at time t are mapped to the robot as $(\mathbf{p}_b^r, \mathbf{O}_b^r)$. Then, \mathbf{p}_b^r is directly sent to the robot as the target position of the robot end-effector \mathbf{M}_t

$$\mathbf{M}_t = \mathbf{p}_b^r \quad (22)$$

In assist mode, we first obtain the task parameters based on the predicted target. Then, our approach corrects \mathbf{p}_b^r using the information contained in the TP-HSMM. Concretely, given the current operator input, \mathbf{p}_b^r , we first approximate the conditional probability distribution of the robot end-effector position from the perspective of each task-parameterized frame as

$$\mathcal{P}(\xi_t^{Oj} | \mathbf{p}_b^r) \approx \mathcal{N}(\tilde{\boldsymbol{\mu}}_t^{Oj}, \tilde{\boldsymbol{\Sigma}}_t^{Oj}) \quad (23)$$

The center and covariance of conditional probability distribution $\mathcal{P}(\xi_t^{Oj} | \mathbf{p}_b^r)$ are estimated based on the block matrix form of $\tilde{\boldsymbol{\mu}}_i^{(j)}$ and $\tilde{\boldsymbol{\Sigma}}_i^{(j)}$ in equation (9)

$$\tilde{\boldsymbol{\mu}}_t^{Oj} = \sum_{i=1}^K h_i(\mathbf{p}_b^r) \hat{\boldsymbol{\mu}}_i^{Oj}(\mathbf{p}_b^r) \quad (24)$$

$$\tilde{\boldsymbol{\Sigma}}_t^{Oj} = \sum_{i=1}^K h_i(\mathbf{p}_b^r) \left(\hat{\boldsymbol{\Sigma}}_i^{Oj} + \hat{\boldsymbol{\mu}}_i^{Oj}(\mathbf{p}_b^r) \hat{\boldsymbol{\mu}}_i^{Oj}(\mathbf{p}_b^r)^T \right) - \tilde{\boldsymbol{\mu}}_t^{Oj} \tilde{\boldsymbol{\mu}}_t^{Oj^T} \quad (25)$$

with

$$h_i(\mathbf{p}_b^r) = \frac{\pi_i \mathcal{N}(\mathbf{p}_b^r | \tilde{\boldsymbol{\mu}}_i^{Ij}, \tilde{\boldsymbol{\Sigma}}_i^{Ij})}{\sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{p}_b^r | \tilde{\boldsymbol{\mu}}_k^{Ij}, \tilde{\boldsymbol{\Sigma}}_k^{Ij})} \quad (26)$$

$$\hat{\boldsymbol{\mu}}_i^{Oj}(\mathbf{p}_b^r) = \tilde{\boldsymbol{\mu}}_i^{Oj} + \tilde{\boldsymbol{\Sigma}}_i^{Oj} \tilde{\boldsymbol{\Sigma}}_i^{Ij^{-1}} (\mathbf{p}_b^r - \tilde{\boldsymbol{\mu}}_i^{Ij})$$

$$\hat{\boldsymbol{\Sigma}}_i^{Oj} = \tilde{\boldsymbol{\Sigma}}_i^{Oj} - \tilde{\boldsymbol{\Sigma}}_i^{Oj} \tilde{\boldsymbol{\Sigma}}_i^{Ij^{-1}} \tilde{\boldsymbol{\Sigma}}_i^{Ij}$$

Synthesizing the information from all task-parameterized frames using the product of Gaussians, we can get the conditional probability distribution of the end-effector's position at time t

Algorithm 1. Shared control teleoperation.

Input: The learned task parameterized HSMM; the input of operator $(\mathbf{p}_b^r, \mathbf{O}_b^r)$.

Output:

- 1: **for** every $G \in \mathcal{G}$ **do**
- 2: set new task parameters using Eq. 8;
- 3: plan the robotic motion from S to G , $\bar{\xi}_{S \rightarrow G}$;
- 4: calculate the posterior probability using Eq. 18.
- 5: **end for**
- 6: predict the target G^* using Eq. 19.
- 7: calculate the confidence *conf* using Eq. 20.
- 8: **if** *conf* $< \mathcal{T}$ **then**
- 9: get the target pose of robot using Eq. 22.
- 10: **else**
- 11: set the task parameters according to the target G^* ;
- 12: estimate the intended robotic movement using Eq. 28;
- 13: correct the input of the operator using Eq. 29.
- 14: **end if**
- 15: send \mathbf{M}_t to the robot.

$$\mathcal{N}(\hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\Sigma}}_t) \propto \prod_{j=1}^P \mathcal{N}(\tilde{\boldsymbol{\mu}}_t^{Oj}, \tilde{\boldsymbol{\Sigma}}_t^{Oj}) \quad (27)$$

Evaluating the product of Gaussians yields

$$\hat{\boldsymbol{\Sigma}}_t = \left[\sum_{j=1}^P (\tilde{\boldsymbol{\Sigma}}_t^{Oj})^{-1} \right]^{-1} \quad (28)$$

$$\hat{\boldsymbol{\mu}}_t = \hat{\boldsymbol{\Sigma}}_t \sum_{j=1}^P \left[(\tilde{\boldsymbol{\Sigma}}_t^{Oj})^{-1} \tilde{\boldsymbol{\mu}}_t^{Oj} \right]$$

The mean $\hat{\boldsymbol{\mu}}_t$ denotes the most likely desired position of the end-effector given the operator input \mathbf{p}_b^r . In this work, we utilize $\hat{\boldsymbol{\mu}}_t$ to correct \mathbf{p}_b^r to yield the target position of the robot end-effector

$$\mathbf{M}_t = \alpha \hat{\boldsymbol{\mu}}_t + (1 - \alpha) \mathbf{p}_b^r \quad (29)$$

where $\alpha \in (0, 1)$ is a combination coefficient. By regulating the value of α , we can adjust the level of correction to the input of the operator. If α is close to 0, \mathbf{M}_t is largely decided by \mathbf{p}_b^r . When α increases to 1, \mathbf{M}_t is mainly determined by the estimated movement, $\hat{\boldsymbol{\mu}}_t$. The value of α cannot be set too large as human interference is important in teleoperation. Especially when the predicted goal is wrong or the estimated $\hat{\boldsymbol{\mu}}_t$ is improper, the operator will not be able to correct the movement of the robot if α is close to 1.

Our approach for shared control teleoperation is summarized in Algorithm 1. The TP-HSMM is used to predict the manipulation target and estimate the intended position of the robotic end-effector. When the confidence of the prediction is higher than a switching threshold, the operation assistance is provided by correcting the input of the operator.

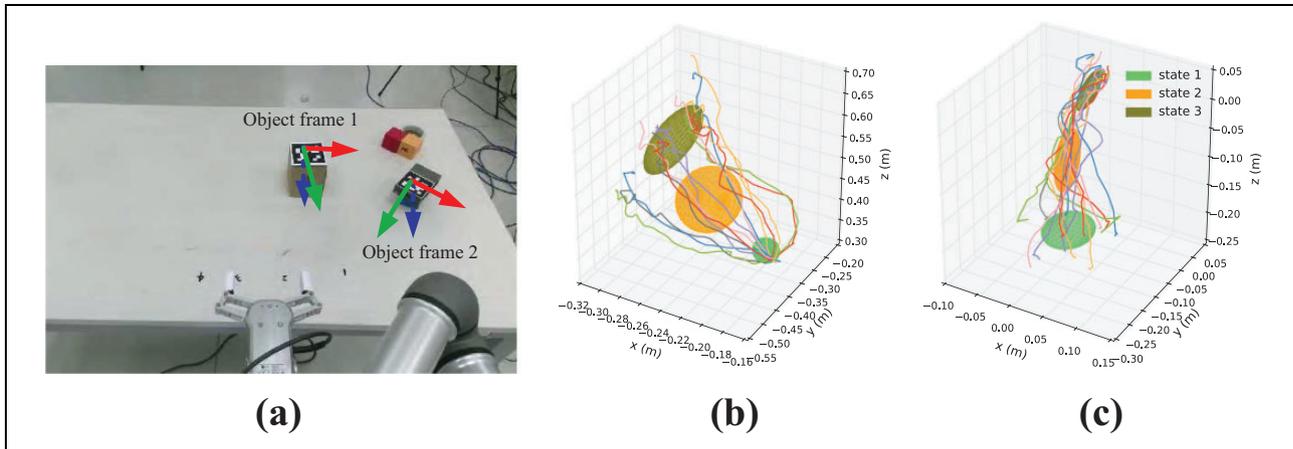


Figure 2. Manipulation environment, demonstrations, and the learned HSMM. (a) The manipulation environment captured by the Kinect fixed on the robot and two object frames. Demonstrations and models in (b) right base frame and (c) object frame. The lines are the recorded demonstrations, and the ellipsoids represent the Gaussian hidden states of the learned HSMM. HSMM: hidden semi-Markov model.

Experiments

In this section, two experiments are conducted to validate the proposed shared control teleoperation method. In the first experiment, we evaluate the performance of predicting the manipulation target in a working environment that contains more than one candidate objects. In the second experiment, we apply the proposed method in a bimanual manipulation scenario to show how the cooperation behaviors can be learned and used to assist the operator during teleoperation.

Experimental setup

In the following experiments, the AprilTags are used to help to detect the poses of the interested objects in the manipulation environment. The experimental platform consisted of a Microsoft Kinect 2.0 sensor, a dual-arm robot, and a set of HTC Vive VR devices. The Kinect is placed on the robot whose view is as shown in Figure 2(a). The dual-arm robot is composed of two six-DoF Universal Robot 3 (UR3), and each UR3 is equipped with a two-finger gripper as shown in Figure 5(a). UR3 can be controlled by torque and position mode, and all its joints contribute to the transformational and rotational movements of its end effector. We operate each UR3 in position control mode by sending the target pose of its end-effector (x, y, z, rx, ry, rz) continuously to its controller via network, and UR3 moves its end effector to the latest target pose as quickly as possible. In our system, HTC Vive controllers are used to measure the position and orientation of the user’s hands, and triggers of the controllers are used to command the opening and closing of the two-finger grippers. The vision feedback of the remote working environment captured by the Kinect is displayed in the HTC Vive headset.

Goal prediction

Reaching is a standard benchmark in robotics because many tasks can be represented by one or more reaching tasks. In order to assist the user in these tasks, the robot has to know which object the user is intended to reach. In this experiment, more than one object were placed in the manipulation environment (see Figure 2(a)), and the operator teleoperated the robot to pick one of the two objects. During the operation, the intended goal object was predicted in real time. This scenario allows us to demonstrate the ability of the proposed method to predict the manipulation target.

This experiment consists of a learning phase and a prediction phase. In the learning phase, the user teleoperated the robot directly to reach an object from initial pose and recorded the pose of the end-effector as human demonstrations. There were 15 demonstrations (see Figure 2) recorded for different locations of the target object.

In this experiment, one task-parameterized frame ($P = 1$) is utilized to describe the manipulation context, namely the object frame, which is located at the center of the AprilTag marker attached on the object as shown in Figure 2(a). The demonstrations were used to train a TP-HSMM with $K = 3$ states using EM algorithm. The states of the resulting TP-HSMM in the right base frame and the object frame are displayed in Figure 2(b) and (c). The ellipsoids represent the states of HSMM, whose locations and shapes are determined by the mean and covariance of the corresponding Gaussian distributions. The models in these two reference frames describe the reaching task from different perspectives. In the right base frame, since all demonstrations start from a same initial position and end at different object positions, the variance of the demonstrations gets bigger and bigger, which is reflected by the size of the ellipsoids (see Figure 2(b)). The object frame is located

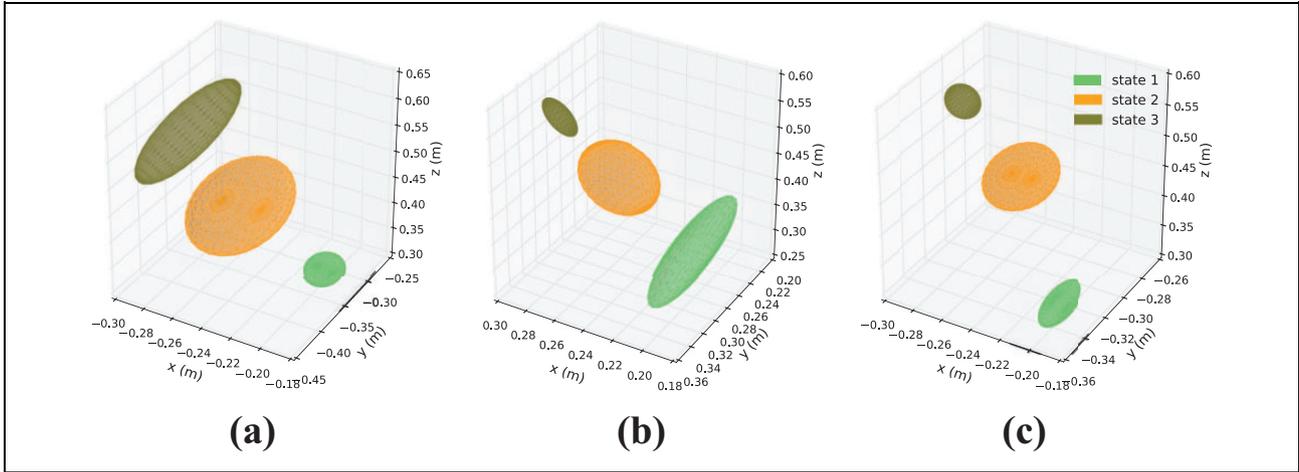


Figure 3. The adaptation of the learned model to new environment. (a) The base frame model is the HSMM model in the right base frame. (b) The transformed model is obtained by transforming the learned model in object frame to the robot base frame according to the current pose of the object. (c) The product model is the product of the base frame model and the transformed model. HSMM: hidden semi-Markov model.

on the object, as shown in Figure 2(a). When the demonstrations are transformed from the right base frame to the object frame, they converge to the origins of the object frame at the end phase. Since the location of the object is different in each demonstration, the initial position of the robotic end-effector is different in the object frame. Therefore, in the object frame, the starting phase of the demonstrations shows great variance. As a result, the size of the ellipsoid corresponding to state 3 is small and that corresponding to state 1 is large as shown in Figure 2(c).

Given the current pose of the object, the model in the object frame (Figure 2(b)) is transformed to the right base frame to get the transformed model (Figure 3(b)) using equation (8). Therefore, the transformed model contains the information about the location of the object while the base frame model (Figure 3(a)) contains the information about the initial position of the robotic end-effector. The information of the two models can be used to complement each other. An all-round description of the task is obtained by computing the product of the base frame model and the transformed model. The product of two Gaussian is still a Gaussian, and the product is closer to the one with smaller variance. Since the variance of the first state of the base frame model is small and that of the transformed model is large, the first state of the product model is closer to that of the base frame model. Analogously, the third state of the product model is closer to that of the transformed model. Therefore, the product model reflects the synthesizing of the information from different frames of reference. On the other hand, the product model is the motion trajectory distribution of the reaching task. In a specific reaching task, since the start and end position of the motion is fixed, the variances of the trajectory segments corresponding to the start and end phase are small. However, the middle part of the trajectory is not

fixed, so the variance is high. Therefore, as shown in Figure 3(c), the size of the first and third ellipsoid is small while that of the second one is large.

The product model is used as a motion planner to evaluate probability distribution over the possible targets. The dotted lines in the left column of Figure 4 are the planned trajectories to the two objects at different locations. At every time step, the probability to each object is estimated based on the similarity between the planned trajectory and the recorded actual trajectory. The β in equation (18) is set at 50 to evaluate the probabilities. In the first two cases (see Figure 4(a) and (d)), since the operator teleoperated the robot to reach one of the objects directly, the estimated probabilities to that object increased rapidly and the prediction confidence also increased monotonically. In the third case, the robot moved toward object 2 at the beginning and then turned to object 1 as shown in Figure 4(g). Since the user changed mind during the operation, the estimated probability and the prediction confidence exhibit fluctuations apparently. Therefore, the probabilities to each objects and the prediction confidence in these three cases can reflect the intent of the user clearly.

Bimanual peg-in-hole

Bimanual teleoperation tasks involve up to two robotic arms at the same time, thus increasing the operational difficulty. In this example, the user teleoperated the dual arm robot to insert a peg held by the right end-effector into a slot held by the left end-effector as shown in Figure 5(a). Due to the heterogeneity between human and the dual arm robot, this kind of manipulation task is often difficult for the operator.

In bimanual manipulation tasks, the relative motions can be used to specify the desired motions of both robotic arms.

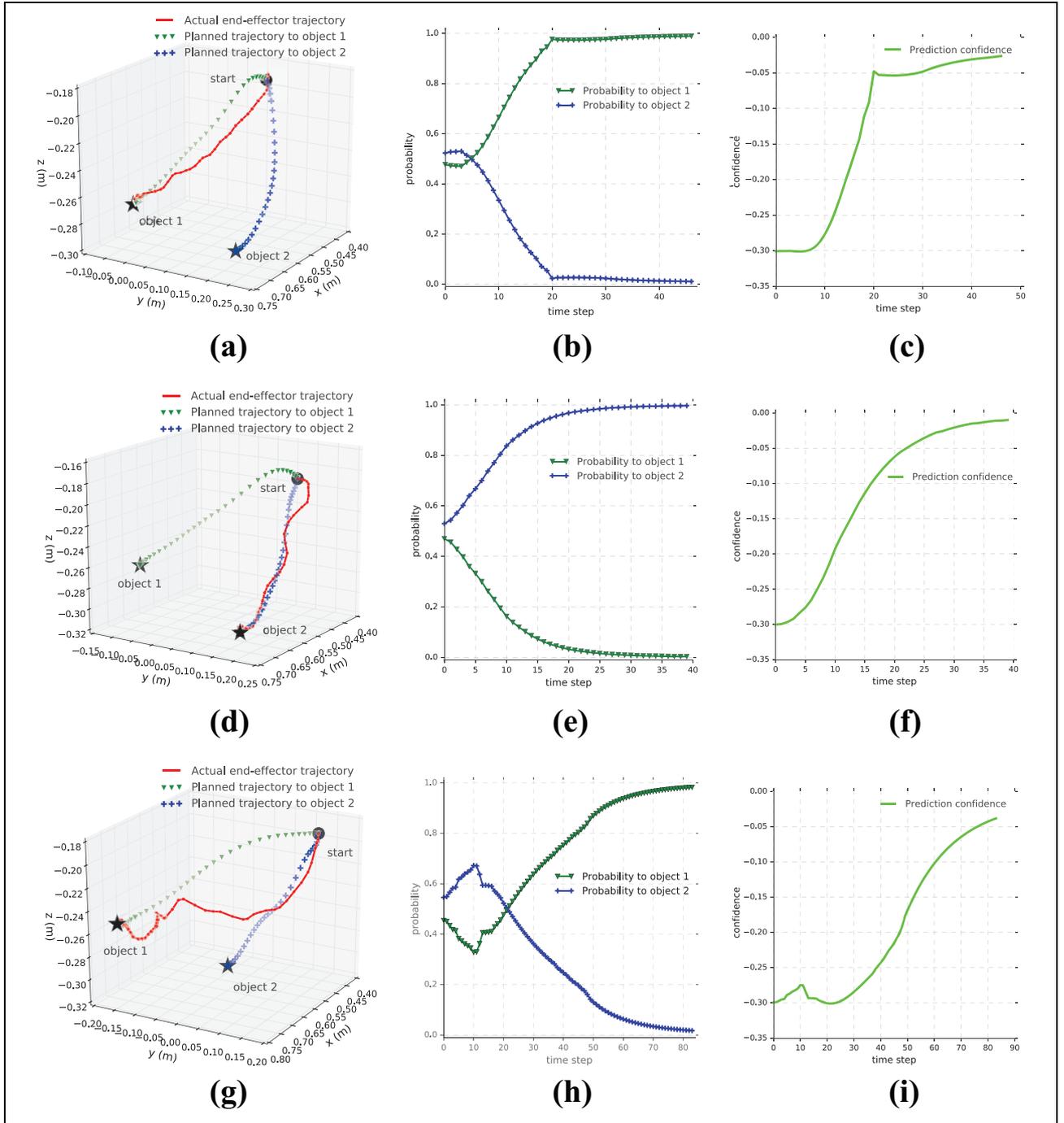


Figure 4. Goal prediction in three different cases (three rows): left column illustrates the actual trajectories and the planned trajectories of the end-effector to each of the two objects in the manipulation environment; the middle column illustrates the estimated probabilities move to one of the objects during operation; the right column gives the confidence of the predictions. The black stars represent the positions of different objects and the black circles represent the start position of the robot’s end-effector. (a), (d), and (g) The actual and planned trajectories; (b), (e), and (h) the probabilities to different objects; and (c), (f), and (i) the prediction confidence.

In order to describe the relative motion between the two robotic arms in the task, we attached a task-parameterized frame on the left end-effector called left end-effector frame. A task-parameterized model with $K = 4$ states and $P = 1$ task-parameterized frame is used to encode the demonstrations.

The demonstrations and the learned models in the robot base frame and in the left end-effector frame are illustrated in Figure 5(b) and (c). The red ellipsoids represent the fourth state of HSM, which correspond to the insertion phase of the peg-in-hole task. Since the position of the left end-effector is different between demonstrations, the

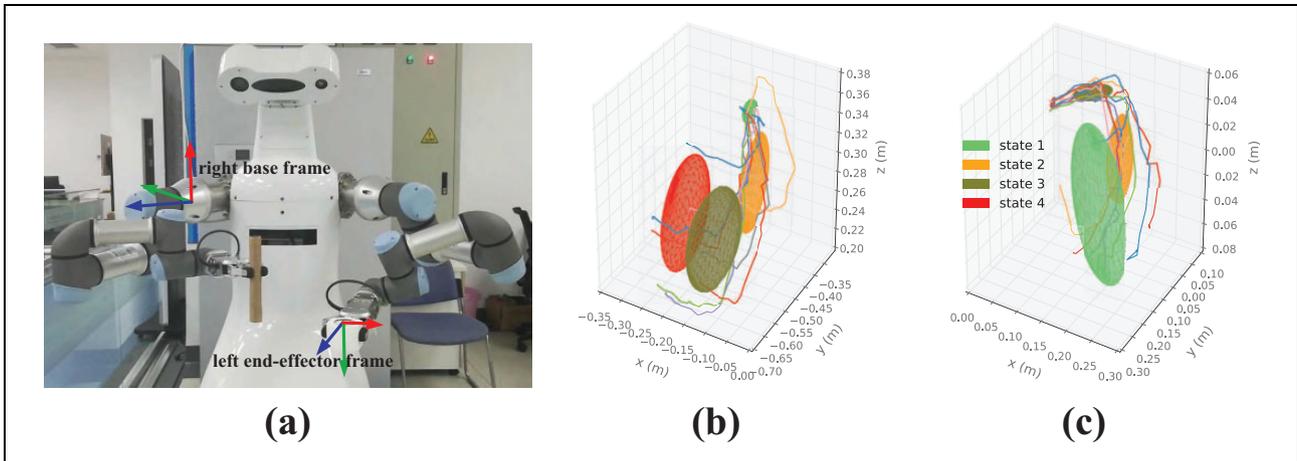


Figure 5. The dual arm robot, demonstrations, and learned HSMM. The ellipsoids represent the Gaussian hidden states of the learned HSMM: (a) the dual arm robot and some frames of reference used in the experiment; (b) demonstrations and models in the base frame; and (c) demonstrations and models in left end-effector frame. HSMM: hidden semi-Markov model.

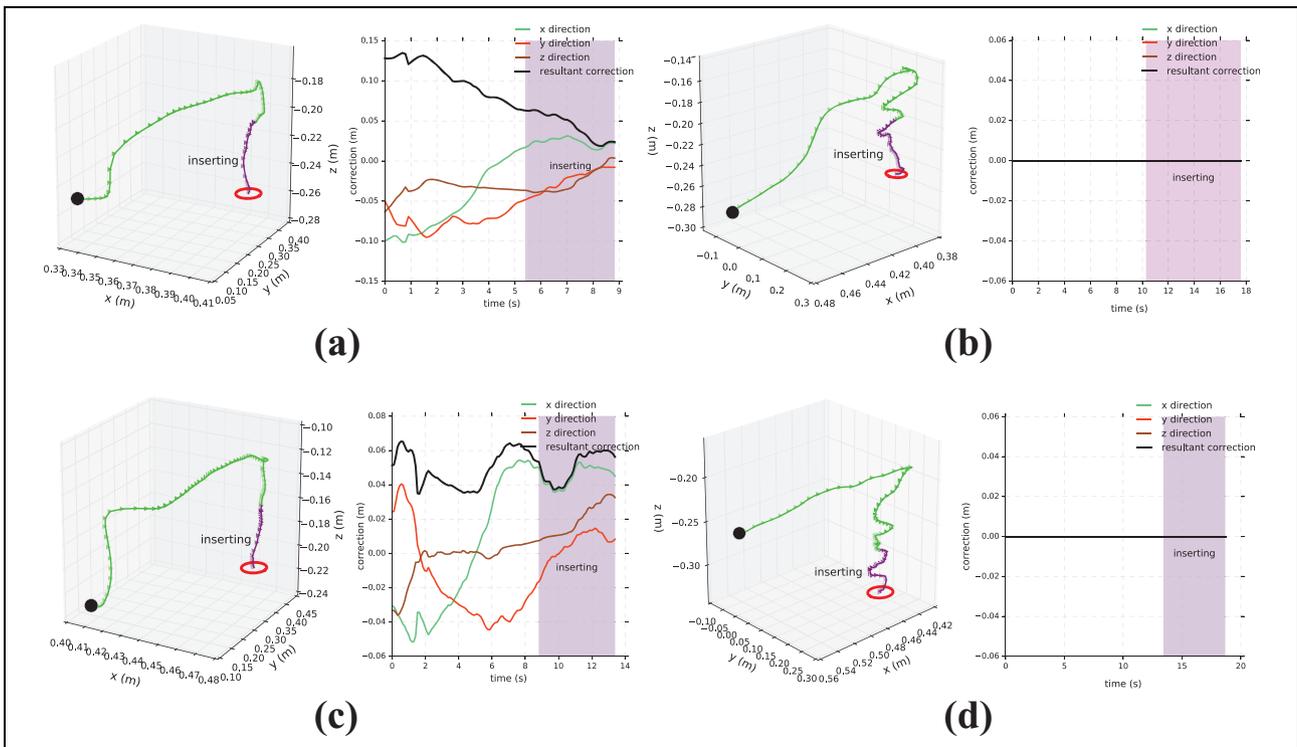


Figure 6. The trajectories and corrections during teleoperation. The red circles represent the position of the slot and the purple part of the trajectories indicate less than 10 cm to the slot. Plots in the second and fourth columns show the corrections to the input of the operator, in which the purple rectangle regions correspond to the purple parts of the trajectories. Trajectory and corrections in (a) and (c) assist mode and (b) and (d) direct mode.

trajectories in the robot base frame show great variance in the insertion phase. Conversely, in the left gripper frame, all trajectories finally gather at the position of the peg. Therefore, the red ellipsoids are large in the robot base frame but rather small in the left end-effector frame.

During teleoperation, the position and orientation of the task-parameterized frame are updated at each time step.

Then, learned model is transformed by equation (8) according to the new task parameter.

We compare the shared control teleoperation with direct teleoperation by analyzing the corresponding trajectories of the right end-effector. The operators teleoperated the robot to do the bimanual peg-in-hole task in the direct mode and the assist mode 10 times, respectively. In assist mode, the

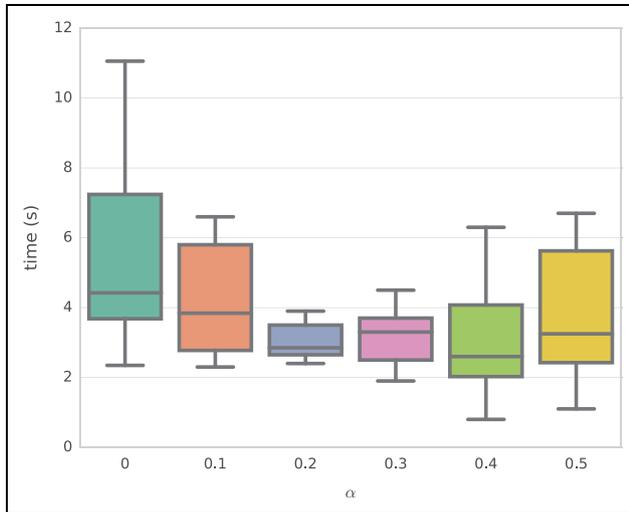


Figure 7. Duration time of the insertion phase for different α values.

value of α was empirically set to 0.2. The learned model assisted the operator in controlling the position of the right end-effector while the orientation was controlled by the operator directly. In the direct mode, the operator controls the robot directly without corrections as shown in Figure 6(b) and (d). The purple part of the trajectories indicates that the operator is continuously adjusting the position of the peg in order to insert into the slot. In assist mode, the purple part is smoother which illustrates that the operator did less adjustment during operation. The corrections in the assist mode are illustrated in Figure 6(a) and (c). The purple rectangle regions correspond to the purple part of the trajectories.

To investigate the influence of α in shared control, we set the value of α to different values and teleoperate the robot to do the peg-in-hole task 10 times for each value. The duration time of the insertion phase for different values of α is illustrated in Figure 7. We compare the performance of teleoperation from the perspectives of mean and variance of the duration time of the insertion phase. When $\alpha = 0$, the system works in direct teleoperation mode. As shown in Figure 7, the mean and variance of the duration time in direct mode are higher than those in shared control, where $\alpha \neq 0$. Therefore, the assistance can improve the manipulation efficiency in teleoperation tasks. The high variance indicates the performance is effected by the manipulation context greatly, while the low variance indicates the performance is stable. When the value of α is close to 0, the manipulation assistance is weak and the performance is close to direct control. As the value of α gets bigger, the influence of the operator becomes weaker. If the estimation of desired position is good, the duration time of larger α is usually shorter than that of the smaller α . On the other hand, if the estimation is bad, the duration time of larger α becomes longer. For example, in Figure 7, the minimum duration time of large α is shorter than that of

the small α , while the maximum duration time of the large α is longer than that of the small α . Therefore, when α gets larger, even though the mean duration time might become lower, the variance usually becomes higher. Taking both the mean and variance into account, the value 0.2 performs good.

Conclusion

In this article, a shared control method based on LfD is proposed to assist operator in manipulation tasks. The VR devices are used to capture the motion of the human operator and display the vision feedback from the remote site. A motion mapping approach is developed to map the operator motion to the robot. We adopt a predict-then-blend framework to correct the operator input during operation. A TP-HSMM is utilized to encode human demonstrations so that the model can adapt to new situations. The learned model is used as a motion planner to predict the manipulation target based on the principle of maximum entropy and Bayes' theorem. Then, the desired motion of the robot is estimated given the current operator input and task parameters. We utilize the estimated desired robot motion to correct operator input. Our method can apply to bimanual teleoperation tasks to reduce the operator workload. We set the pose of left end-effector as a task parameter to describe the relative behavior of the two robotic arm in bimanual tasks. The experimental results show that the proposed method can predict the manipulation target accurately and provide useful assistance to the operator to reduce the manipulation workload and improve the operational efficiency. In this work, only the input position of the operator is corrected. We plan to investigate a way to assist the operator in controlling the orientation of the end-effector in future work.

Declaration of conflicting interests

The author(s) declared no potential conflict of interest with respect to the research, authorship and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported in part by the National Natural Science Foundation of China under grant 61773378, grant U1713222, and grant 61703401; in part by the Early Career Development Award of SKLMCCS; and partly by the National Key R & D Program of China (grant 2017YFB1300202) and Science and Technology on Space Intelligent Control Laboratory for National Defense, no. KGJZDSYS-2018-09.

ORCID iD

Bao Xi  <https://orcid.org/0000-0003-1495-8802>

References

1. Kai A, Deisenroth MP, Brundage M, et al. Deep reinforcement learning: a brief survey. *IEEE Signal Proc Mag* 2017; 34(6): 26–38.
2. Levine S, Finn C, Darrell T, et al. End-to-end training of deep visuomotor policies. *J Mach Learn Res* 2016; 17(1): 1334–1373.
3. Hussein A, Gaber MM, Elyan E, et al. Imitation learning: a survey of learning methods. *ACM Comput Surv (CSUR)* 2017; 50(2): 21.
4. Yang C, Wang X, Long C, et al. Neural-learning-based tele-robot control with guaranteed performance. *IEEE Trans Cybern* 2016; 47(10): 3148–3159.
5. Meltzoff ADJ. What imitation tells us about social cognition: a rapprochement between developmental psychology and cognitive neuroscience. *Philos Trans Royal Soc Lond. Ser B: Biol Sci* 2003; 358(1431): 491–500.
6. Csibra G. Teleological and referential understanding of action in infancy. *Philos Trans Royal Soc Lond. Ser B Biol Sci* 2003; 358(1431): 447–458.
7. Calinon S, Pistillo A, and Caldwell DG. Encoding the time and space constraints of a task in explicit-duration hidden Markov model. In: *IEEE/RSJ international conference on intelligent robots and systems*, San Francisco, CA, USA, 25–30 September 2011, pp. 3413–3418.
8. Trautman P. Assistive planning in complex, dynamic environments: a probabilistic approach. In: *IEEE international conference on systems, man, and cybernetics*, Kowloon, 2015, pp. 3072–3078.
9. Rebelo J, Sednaoui T, Exter EBD, et al. Bilateral robot teleoperation: a wearable arm exoskeleton featuring an intuitive user interface. *IEEE Robot Autom Mag* 2014; 21(4): 62–69.
10. Park S, Jung Y, and Bae J. A tele-operation interface with a motion capture system and a haptic glove. In: *International conference on ubiquitous robots and ambient intelligence*, Xi'an, 2016, pp. 544–549.
11. Kobayashi F, Kitabayashi K, Nakamoto H, et al. Hand/arm robot teleoperation by inertial motion capture. In: *2013 Second international conference on robot, vision and signal processing*, 2013, pp. 234–237.
12. Kofman J, Wu X, Luu TJ, et al. Teleoperation of a robot manipulator using a vision-based human-robot interface. *IEEE Trans Ind Elect* 2005; 52(5): 1206–1219.
13. Du G and Zhang P. Markerless human-robot interface for dual robot manipulators using Kinect sensor. *Robotics Comput Integr Manuf* 2014; 30(2): 150–159.
14. Xi B, Liu N, Lu T, et al. Robot teleoperation system based on SVDD. In: *Chinese automation congress (CAC)*, Jinan, 2017, pp. 2830–2835.
15. Lipton JI, Fay AJ, and Rus D. Baxter's Homunculus: virtual reality spaces for teleoperation in manufacturing. *IEEE Robot Autom Lett* 2018; 3(1): 179–186.
16. Zhang T, McCarthy Z, Jow O, et al. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, 2018, pp. 1–8.
17. Javdani S, Srinivasa SS, and Bagnell JA. Shared autonomy via hindsight optimization. *Comput Sci* 2015.
18. Narayanan VK, Spalanzani A, and Babel M. A semi-autonomous framework for human-aware and user intention driven wheelchair mobility assistance. In: *IEEE/RSJ international conference on intelligent robots and systems*, Daejeon, 2016, pp. 4700–4707.
19. Dragan AD and Srinivasa SS. A policy-blending formalism for shared control. *Int J Robot Res* 2013; 32(7): 790–805.
20. Schultz C, Gaurav S, Monfort M, et al. Goal-predictive robotic teleoperation from noisy sensors. In: *IEEE international conference on robotics and automation (ICRA)*, Singapore, 2017, pp. 5377–5383. DOI: 10.1109/ICRA.2017.7989633.
21. Oosterhout JV, Wildenbeest JGW, Boessenkool H, et al. Haptic shared control in tele-manipulation: effects of inaccuracies in guidance on task execution. *IEEE Trans Haptics* 2015; 8(2): 164–175.
22. Boessenkool H, Heemskerk CJM, Heemskerk CJM, et al. A task-specific analysis of the benefit of haptic shared control during telemanipulation. *IEEE Trans Haptics* 2013; 6(1): 2–12.
23. Ziebart BD, Maas A, Bagnell JA, et al. Maximum entropy inverse reinforcement learning. In: *Proceedings of the twenty-third AAAI conference on artificial intelligence* (ed A Cohn), 2008, pp. 1433–1438. AAAI Press.
24. Ziebart B, Dey A, Bagnell JA, et al. Probabilistic pointing target prediction via inverse optimal control. In: *ACM conference on intelligent user interfaces*, 2012, pp. 1–10. USA: ACM.
25. Monfort M, Liu A, and Ziebart BD. Intent prediction and trajectory forecasting via predictive inverse linear-quadratic regulation. In: *AAAI conference on artificial intelligence*, 2015, pp. 3672–3678.
26. Koppula HS and Saxena A. Anticipating human activities using object affordances for reactive robotic response. *IEEE Comput Soc* 2016; 38(1): 14–29.
27. Martinez J, Black MJ, and Romero J. On human motion prediction using recurrent neural networks. In: *IEEE conference on computer vision and pattern recognition (CVPR)*, Honolulu, 2017, pp. 4674–4683.
28. El-Hussieny H, Assal SFM, Abouelsoud AA, et al. A novel intention prediction strategy for a shared control tele-manipulation system in unknown environments. In: *IEEE international conference on mechatronics (ICM)*, Nagoya, 2015, pp. 204–209. DOI: 10.1109/ICMECH.2015.7083975.
29. Rosenberg LB. Virtual fixtures: perceptual tools for telerobotic manipulation. In: *IEEE virtual reality international symposium*, 1993, Seattle, USA, pp. 76–82.

30. Bohren J, Paxton C, Howarth R, et al. Semi-autonomous telerobotic assembly over high-latency networks. In: *ACM/IEEE international conference on human-robot interaction*, Christchurch, 2016, pp.149–156.
31. Raiola G, Lamy X, and Stulp F. Co-manipulation with multiple probabilistic virtual guides. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Hamburg, 2015, pp. 7–13.
32. Tanwani AK and Calinon S. A generative model for intention recognition and manipulation assistance in teleoperation. In: *IEEE/RSJ international conference on intelligent robots and systems*, Vancouver, BC, 2017, pp. 43–50.
33. Rabiner LR. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc IEEE* 1989; 77(2): 257–286.