Autonomous Robots
https://doi.org/10.1007/s10514-019-09853-4

Motion encoding with asynchronous trajectories of repetitive teleoperation tasks and its extension to human-agent shared teleoperation

Affan Pervez[1] · Hiba Latifee[2] · Jee-Hwan Ryu[2] · Dongheui Lee[1,3]

Received: 15 December 2018 / Accepted: 22 April 2019

Abstract
Teleoperating a robot for complex and intricate tasks demands a high mental workload from a human operator. Deploying multiple operators can mitigate this problem, but it can be also a costly solution. Learning from Demonstrations can reduce the human operator's burden by learning repetitive teleoperation tasks. Yet, the demonstrations via teleoperation tend to be inconsistent compared to other modalities of human demonstrations. In order to handle less consistent and asynchronous demonstrations effectively, this paper proposes a learning scheme based on Dynamic Movement Primitives. In particular, a new Expectation Maximization algorithm which synchronizes and encodes demonstrations with high temporal and spatial variances is proposed. Furthermore, we discuss two shared teleoperation architectures, where, instead of multiple human operators, a learned artificial agent and a human operator share authority over a task while teleoperating cooperatively. The agent controls the more mundane and repetitive motion in the task whereas human takes charge of the more critical and uncertain motion. The proposed algorithm together with the two shared teleoperation architectures (human-synchronized and agent-synchronized shared teleoperation) has been tested and validated through simulation and experiments on 3 Degrees-of-Freedom Phantom-to-Phantom teleoperation. Conclusively, the both proposed shared teleoperation architectures have shown superior performance when compared with the human-only teleoperation for a peg-in-hole task.

Keywords Dynamic movement primitives · Learning from demonstrations · Teleoperation · Human-agent shared teleoperation · Cooperative teleoperation · Human-synchronized · Agent-synchronized · Haptic feedback

1 Introduction

Teleoperation allows for an operator to remotely control a robot in situations where it is not possible to place a human with the robot. From the human's stand-point, teleoperation requires a heavy mental workload and, for repetitive tasks, it can be tedious and in turn, could lead to errors. To mitigate these problems, multiple operators can be deployed to execute the task. In such a setting for complex teleoperation tasks, the control authority of a slave robot is distributed among the human operators. This substantially decreases the cognitive workload of each operator and hence, has shown to ensure a reliable execution of the task (Usmani et al. 2015). The issue of dynamically divid-

This work is partially supported by Helmholtz Association and the Industrial Strategic Technology Development Program (10069072) funded by the MOTIE.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10514-019-09853-4) contains supplementary material, which is available to authorized users.

✉ Affan Pervez
affan.pervez@tum.de

Hiba Latifee
hibalatifee@koreatech.ac.kr

Jee-Hwan Ryu
jhryu@koreatech.ac.kr

Dongheui Lee
dhlee@tum.de

[1] Department of Electrical and Computer Engineering, Technical University of Munich (TUM), Munich, Germany

[2] Department of Mechanical Engineering, Korea University of Technology and Education, Cheonan, South Korea

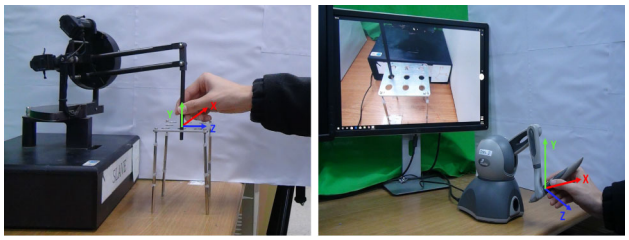
[3] Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Cologne, Germany

Published online: 02 May 2019

ing the control authority among multiple operators with multiple Field-of-Views (FOVs) is addressed in Gromov et al. (2012). That way, an operator can always take over partial or full authority of the task, whenever and wherever a need is. This distribution of the control authority of the slave robot among many operators, based on the information available to them, is absolutely pivotal for cooperative teleoperation systems. But usually having multiple human operators can be very costly. Also, in a setting with multiple operators, a fixed distribution of control cannot handle a camera failure or an obstruction in FOV of one operator.

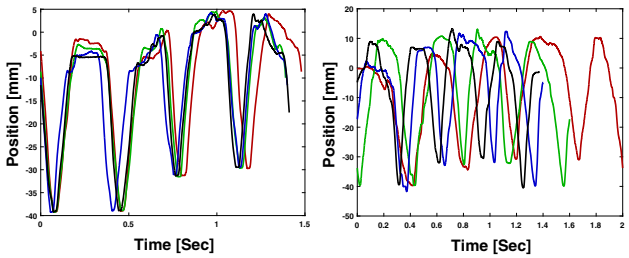
Autonomous execution of the task is desirable to avoid the challenges associated with teleoperation. But, generally, a robot is pre-programmed for performing different repetitive tasks. A slight change in a task or the environment requires re-programming of the robot, which can be a painstakingly tedious and temporally demanding process (Billard et al. 2008). Learning from Demonstrations (LfD), also known as imitation learning, provides an intuitive way to automate the task, by readily transferring new skills from humans to robots, and generalizes the task in slightly changed environments (Argall et al. 2009; Billard et al. 2008; Calinon and Lee 2018; Power et al. 2015).

Vision-based motion tracking systems can be utilized to capture human motions when demonstrating a task. For humanoid robots, bipedal systems, and robotic arms, this modality can be easily utilized to collect demonstrations since they have the similar kinematic structures as that of a human (Hu et al. 2014; Ott et al. 2008). The main demerit of this approach is that if the robot has a different kinematic structure with that of a human then it creates a correspondence problem. To rule out correspondence problem, kinesthetic teaching can be employed for collecting human demonstrations. In kinesthetic teaching, a human physically interacts with a robot to demonstrate a task (Lee and Ott 2010; Saveriano et al. 2015). Hence, kinesthetic teaching is one of the most widely utilized modalities for collecting human demonstrations. But, as mentioned earlier, there are situations where a direct human-robot interaction is not possible, as in deep sea (Havoutis and Calinon 2019) or outer space applications, etc. In such a scenario, teleoperation can be utilized for collecting demonstrations (Akgun et al. 2012; Hokayem and Spong 2006). Figure 1 shows the application of kinesthetic and teleoperation-based teaching for the peg-in-hole task (details in experiments) while the y-axis motion of the collected demonstrations can be visualized in Fig. 2. It can be observed that teleoperated demonstrations have high temporal and spatial variations, as compared to the demonstrations recorded via kinesthetic teaching. Due to this high inconsistency, existing LfD approaches tend to perform poorly for teleoperated demonstrations (Bukchin et al. 2002; Pervez et al. 2017).



(a) **(b)**

Fig. 1 **a** Kinesthetic teaching by directly holding the robot, **b** teleoperation-based teaching using 3-DOF phantom haptic devices



(a) **(b)**

Fig. 2 Y-axis motion for the four demonstrations recorded via **a** kinesthetic teaching and **b** teleoperation-based teaching for the peg-in-hole task

Although autonomous execution of a task on a robot relaxes the workload of a human operator, the issue of correct task execution, especially in critical and intricate motions, has to be addressed for effective performance. The efficiency and safety in critical motions, like performing surgeries, can be thoroughly ensured only when both a human and an autonomous artificial agent leverage from each other's capabilities (Dragan and Srinivasa 2012; Medina et al. 2012). If some DOFs in a task are repetitive while some have non-repetitive motions, then those repetitive DOFs can be encoded by LfD while those non-repetitive DOFs can be controlled by a human operator.

Hidden Markov Models (HMMs) can be utilized for skill learning in teleoperation (Yang et al. 1994). In Peters et al. (2003), NASA space humanoid robot learned how to perform low level task such as reaching and grasping, whereas high-level commands were given by the operator. Rozo et al. (2010a, b) used Gaussian Mixture Model (GMM) and Gaussian Mixture Regression (GMR) for teaching a rigid-container emptying skill to a robot via a haptic interface. Other works considered encoding of force/torque signal with large time discrepancies by using a GMM with temporal information encapsulated by an HMM (Rozo et al. 2013). In Schmidts et al. (2011), authors used a teleoperation setup for teaching human grasping skills, by utilizing motion and force data. Teleoperation can be also utilized for online human-in-the-loop robot learning schemes (Peternel et al. 2016, 2018). In Peternel and Babic (2013), the

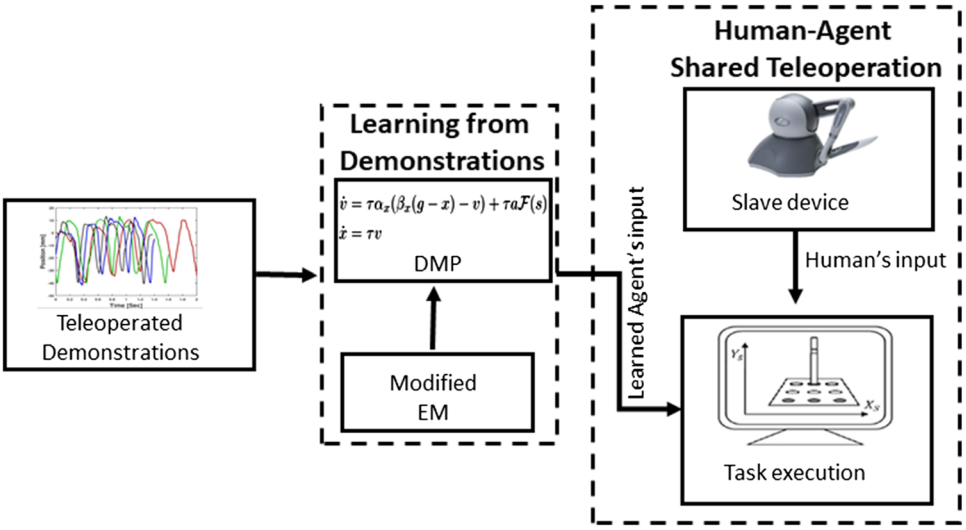


Fig. 3 Block diagram of human-agent shared teleoperation

authors employ Locally Weighted Projection Regression to gradually transfer control from a human demonstrator to an incrementally built autonomous controller. To the best of our knowledge, there has been no prior research on developing an LfD approach that can handle inconsistent demonstrations with large spatial and temporal variations. Most of the existing LfD approaches use Dynamic Time Wrapping (DTW) as a preprocessing step in order to handle some level of spatial and temporal variations.

In this work we compare kinesthetic teaching with the teaching via teleoperation and highlight some inherent challenges associated with teleoperated demonstrations. We propose a novel learning approach for handling the unique features encountered during teleoperation. Our approach is based on Dynamic Movement Primitives (DMP) for learning the demonstrations provided by an operator during teleoperation. In contrast to LfD approaches with DTW, our approach aligns and encodes the motion data simultaneously and incorporates the temporal and spatial variance by adapting the EM algorithm (Dempster et al. 1977; Pervez and Lee 2015). One may think that the mentioned problem could be solved by avoiding the use of an explicit temporal signal. However, time-invariant dynamical systems (Khansari-Zadeh and Billard 2011) have limitation on the kind of motions that can be encoded with a movement primitive. For instance, for a given position, if the end-effector has to move in different directions at different instances of time, then that kind of motion cannot be encoded with the approach presented in Khansari-Zadeh and Billard (2011). Examples are a peg being inserted into and withdrawn from a hole, or an intersecting point for drawing number *8*. In such cases, it is necessary to encode a temporal signal.

The major contributions of this work are:

- proposing a novel LfD method aimed at identifying and hence, overcoming the challenges associated with using teleoperation as an input modality for LfD, and
- extending the preliminary version of our work (Pervez et al. 2017) to the shared teleoperation architecture between a human operator and the artificial agent for cooperative teleoperation.

This paper presents two variants of the human-agent shared teleoperation architecture for dual-operator-single-slave teleoperation systems (pictorially represented in Fig. 5): human-synchronized and agent-synchronized shared teleoperation systems. The performance of the proposed human-agent shared teleoperation methods is evaluated through simulations and experiments. They are compared with the human-only teleoperation, in terms of the speed, accuracy, and the ease of execution of the task, while provisioned with a visual perspective distortion. A general overview of the human-agent shared teleoperation is presented in Fig. 3.

2 LfD for teleoperated demonstrations

While comparing kinesthetic teaching with teleoperation, the former is reported to be the more preferred and easier way of demonstrating a task (Akgun and Subramanian 2011; Fischer et al. 2016). Moreover, the demonstrations collected via kinesthetic teaching have a shorter time duration and a higher success rate of encoding a task. This is due to the operator's situational awareness of kinesthetic teaching, as compared to the difficulty of control in teleoperation.

Most of the existing LfD literature focuses on DTW (Sanguansat 2012) based motions alignment before encoding the demonstrations (Akgun et al. 2012; Calinon et al.

2007). DTW is an algorithm which measures the similarities between two temporal signals executed at different speeds. However, the two-step approach of DTW-based pre-alignment and then encoding of the trajectories does not provide an optimal solution due to the high level of temporal-spatial inconsistency associated with teleoperated demonstrations. To tackle this problem, we propose an EM-based algorithm for simultaneously aligning and encoding the teleoperated demonstrations.

2.1 Dynamic movement primitives (DMP)

The artificial agent proposed in this paper is implemented through DMP. In LfD, DMPs are widely used to learn motor actions as they can encode discrete as well as rhythmic movements (Schaal 2006). For encoding multiple DOFs, a separate DMP is learned for each considered DOF. In a DMP framework, a canonical system acts as a clock. For synchronizing motion of multiple DOFs, each DMP is driven by a common clock signal (Kober and Peters 2010). The canonical system for a discrete DMP is $\dot{s} = -\tau\alpha_s s$, where the parameter s is initialized to one and it monotonically decays to zero, τ is the temporal scaling factor while α_s determines the duration of the movement. Similarly, the canonical system for a rhythmic DMP is $\dot{s} = \tau\omega$, with the parameter s initialized to zero and it increases to 2π at the end of a cycle, and ω determines the phase rate of change of motion. The canonical system drives the second order transformed system:

$$\begin{aligned} \dot{v} &= \tau\alpha_x(\beta_x(g - x) - v) + \tau a\mathcal{F}(s) \\ \dot{x} &= \tau v \end{aligned} \tag{1}$$

where g is either a goal position for discrete DMPs or a mean position for rhythmic DMPs, a is an amplitude modifier term which is usually set as $g - x_0$ with x_0 being the starting position, while the parameters α_x and β_x are set such that the second order system is critically damped. The learning of the forcing term $\mathcal{F}(s)$ allows arbitrarily complex movements.

2.2 DMP learning using GMM

In the original DMP formulation (Sanguansat 2012), locally weighted regression with basis functions is utilized in order to learn the nonlinear forcing terms $\mathcal{F}(s)$. The forcing terms of a DMP can be modeled with any suitable function approximator (Pervez and Lee 2018; Stulp et al. 2013). In this work, we encode the forcing terms with a GMM. A GMM with k components is parameterized by $\boldsymbol{\Theta}_{(k)} = \{\pi_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}_{m=1}^{k}$ where $\pi_1, \ldots, \pi_k$ are mixing weights, $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ are means and $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_k$ are covariance matrices. The probability density function of $\mathbf{y}$ is said to follow the k-component GMM if it can be written as $\mathcal{P}(\mathbf{y}|\boldsymbol{\Theta}_{(k)}) = \sum_{m=1}^{k} \pi_m \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$, subject to the constraints $0 < \pi_m < 1$ and $\sum_{m=1}^{k} \pi_m = 1$. When human demonstrations are collected, each demonstration trajectory is linearly re-sampled to have n number of samples. Re-sampling is performed to ensure that each demonstration gets equal weightage. Without the re-sampling, the shorter demonstrations can get less importance in EM. For each trajectory, the required forcing terms of each DOF at each time instance s are calculated by rearranging the terms in the DMP equation:

$$\mathcal{F}(s) = \dot{v}/(\tau a) - (\alpha_x(\beta_x(g - x) - v))/a \tag{2}$$

Then, we have a sequence of pairs of position x and forcing term $\mathcal{F}(s)$. Assuming that we know the corresponding phase variable s, now we encode the joint distribution of these variables by using a GMM. The teleoperated demonstrations can have large temporal and spatial variation. To handle those variations, we employ the GMM learning approach presented in Pervez et al. (2017). During the reproduction phase, Gaussian Mixture Regression (GMR) is used for predicting the forcing term for a given phase signal s, which is plugged into the DMP equation to get the acceleration command. A separate controller is used for executing the motion.

2.3 EM algorithm for learning from asynchronous demonstrations

As discussed earlier, the teleoperated demonstrations can have temporal variations and hence, we do not associate the time-stamp of the system as the phase signal of each trajectory. In order to handle such inconsistent and asynchronous teleoperated demonstrations, we first separate the demonstrations into two parts: one trajectory is taken as a reference trajectory and the remaining trajectories are then synchronized with this trajectory. The reference trajectory is recommended to be taken as the trajectory with the minimum jerk.

Now, with k demonstrations, we create two data sets. The first data set contains the following variables: the reference trajectory, its forcing terms and the concatenated phase signals (linearly spaced). We call it a complete data set ($\mathbf{X}^{Com}$). The second data set contains the remaining trajectories and their forcing terms. Their phase variables s are unknown in this data set. Thus, it is termed as the incomplete data set ($\mathbf{X}^{InCom}$). The missing phase signals in the second data set will be estimated by synchronizing this data set with the first data set iteratively during EM. We will use the notation $\mathbf{x}_i^{Com}$ and $\mathbf{x}_i^{InCom}$ to denote ith column in complete ($\mathbf{X}^{Com}$) and incomplete data sets ($\mathbf{X}^{InCom}$) respectively.

$$\mathbf{X}^{Com} = \mathbf{X}_1 = \begin{bmatrix} \mathcal{F}_1(s_0) & x_{1,0} & s_0 \\ \vdots & \vdots & \vdots \\ \mathcal{F}_1(s_n) & x_{1,n} & s_n \end{bmatrix}^\top \quad \mathbf{X}^{InCom} = \begin{bmatrix} \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_k \end{bmatrix}^\top$$

Now we fit a GMM to the data sets by using an EM algorithm. The GMM parameters and the missing phase signals in each trajectory should be estimated. These parameters are initialized and are then iteratively updated during EM. The GMM parameters i.e. π, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are initialized by linearly dividing $\mathbf{X}^{Com}$ in time domain. The values of phase signals are initialized as a linearly increasing value from 0 to 2π in each trajectory for a rhythmic DMP and an exponentially decreasing value from 1 to 0 in each trajectory for a discrete DMP. Now we have n data points in the complete data set and $n_2 = n * (k-1)$ data points in the incomplete data set.

The EM algorithm can be categorized into the E- and M-steps as follows:

2.3.1 E-step

First, we separate the variables into two types, missing ($miss$) denoting the phase variable and observable (Obs) denoting all other variables except the phase variable.

$$\boldsymbol{\mu}_m = \begin{bmatrix} \boldsymbol{\mu}_m^{Obs} \\ \boldsymbol{\mu}_m^{miss} \end{bmatrix}, \boldsymbol{\Sigma}_m = \begin{bmatrix} \boldsymbol{\Sigma}_m^{Obs} & \boldsymbol{\Sigma}_m^{Obs,miss} \\ \boldsymbol{\Sigma}_m^{miss,Obs} & \boldsymbol{\Sigma}_m^{miss} \end{bmatrix}$$

In case of a rhythmic DMP, we first map all the phase variables within the interval $[\mu_m^{miss} - \pi, \mu_m^{miss} + \pi]$, for calculating the valid probabilities $p_{i,m}$ and $q_{j,m}$ for the mth GMM component:

$$p_{i,m} = \pi_m \mathcal{N}\left(\mathbf{x}_i^{Com}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\right)$$
$$q_{j,m} = \pi_m \mathcal{N}\left(\mathbf{x}_j^{InCom}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\right)$$

where the initialized (or updated during M-step) phase variables are used for probabilities calculation in incomplete data set.

The responsibility terms for the ith and jth data points in complete and incomplete data sets respectively are calculated as:

$$E[z_{i,m}|\mathbf{x}_i^{Com}, \boldsymbol{\theta}_t] = h_{i,m} = \frac{p_{i,m}}{\sum_{l=1}^{k} p_{i,l}}, \; h_m = \sum_{l=1}^{n} h_{l,m}$$

$$E[z_{j,m}|\mathbf{x}_j^{InCom}, \boldsymbol{\theta}_t] = r_{j,m} = \frac{q_{j,m}}{\sum_{l=1}^{k} q_{j,l}}, \; r_m = \sum_{l=1}^{n_2} r_{l,m}$$

In the incomplete data set, the prediction of the jth missing value with respect to the mth GMM component is done as follows:

$$\hat{x}_{j,m}^{miss} = \mu_m^{miss} + \boldsymbol{\Sigma}_m^{miss,Obs} (\boldsymbol{\Sigma}_m^{Obs})^{-1} \left(\mathbf{x}_j^{InCom,Obs} - \boldsymbol{\mu}_m^{Obs}\right)$$

With this predicted value, two additional expectations are calculated for the incomplete data set (Ghahramani and Jordan 1994):

$$E\left[z_{j,m}, x_j^{InCom,miss} | \mathbf{x}_j^{InCom,obs}, \theta_t\right] = q_{j,m}\left(\hat{x}_{j,m}^{miss}\right)$$
$$E\left[z_{j,m}, x_j^{InCom,miss} x_j^{InCom,miss\top} | \mathbf{x}_j^{InCom,obs}, \boldsymbol{\theta}_t\right] =$$
$$q_{j,m}\left(\boldsymbol{\Sigma}_m^{miss} - \boldsymbol{\Sigma}_m^{miss.obs} \boldsymbol{\Sigma}_m^{obs\,-1} \boldsymbol{\Sigma}_m^{miss.obs\top} + \hat{x}_{j,m}^{miss} \hat{x}_{j,m}^{miss\top}\right)$$

2.3.2 M-step

The mixing weights π_m are updated as:

$$\pi_m = \frac{h_m + r_m}{n + n_2}$$

while the GMM means $\boldsymbol{\mu}_m$ are updated as:

$$\boldsymbol{\mu}_m = \frac{\sum_{l=1}^{n} \mathbf{x}_l^{Com} h_{l,m} + \sum_{v=1}^{n_2} \mathbf{x}_v^{InCom} r_{v,m}}{h_m + r_m}$$

In case of a rhythmic DMP, the phase signal lies on a circular plane for which mean of cos and sin terms is required:

$$\bar{cx} = \frac{\sum_{l=1}^{n} \cos\left(x_i^{Com,miss}\right) h_{l,m} + \sum_{v=1}^{n_2} \cos\left(x_i^{InCom,miss}\right) r_{v,m}}{h_m + r_m}$$
$$\bar{sx} = \frac{\sum_{l=1}^{n} \sin\left(x_i^{Com,miss}\right) h_{l,m} + \sum_{v=1}^{n_2} \sin\left(x_i^{InCom,miss}\right) r_{v,m}}{h_m + r_m}$$

Afterwards, the phase variable for a rhythmic DMP is updated in the GMM means with these following conditions and wrapped in the interval $[0, 2\pi]$:

$$if(\bar{cx}) < 0) \quad \mu_m^{miss} = \tan^{-1}\frac{\bar{sx}}{\bar{cx}} + \pi$$
$$elseif(\bar{sx}) > 0) \quad \mu_m^{miss} = \tan^{-1}\frac{\bar{sx}}{\bar{cx}}$$
$$else \quad \mu_m^{miss} = \tan^{-1}\frac{\bar{sx}}{\bar{cx}} + 2\pi$$

As before, for a rhythmic DMP, we map phase variable within the interval $[\mu_m^{miss} - \pi, \mu_m^{miss} + \pi]$. Next, the covariances $\boldsymbol{\Sigma}_k$ are updated as:

$$\boldsymbol{\Sigma}_m = \frac{\sum_{i=1}^{n} h_{i,m} \left(\mathbf{x}_i^{Com} - \boldsymbol{\mu}_m\right)\left(\mathbf{x}_i^{Com} - \boldsymbol{\mu}_m\right)^\top + \sum_{l=1}^{n_2} \mathbf{A}_l}{h_m + r_m}$$

$$\mathbf{A}_l = r_{l,m} \begin{bmatrix} \mathbf{x}_l^{InCom,obs} - \boldsymbol{\mu}_m^{obs} \\ x_l^{InCom,miss} - \mu_m^{miss} \end{bmatrix} \begin{bmatrix} \mathbf{x}_l^{InCom,obs} - \boldsymbol{\mu}_m^{obs} \\ x_l^{InCom,miss} - \mu_m^{miss} \end{bmatrix}^\top = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{a}_{12} \\ \mathbf{a}_{21} & a_{22} \end{bmatrix}$$

where

$$\mathbf{A}_{11} = r_{l,m} \left(\mathbf{x}_l^{InCom,obs} - \boldsymbol{\mu}_m^{obs}\right)\left(\mathbf{x}_l^{InCom,obs} - \boldsymbol{\mu}_m^{obs}\right)^\top$$

$$\mathbf{a}_{21} = \left(E\left[z_{l,m}, x_l^{InCom,miss}\right] - r_{l,m}\mu_m^{miss}\right)\left(\mathbf{x}_l^{InCom,obs} - \boldsymbol{\mu}_m^{obs}\right)^\top$$

$$\mathbf{a}_{12} = \mathbf{a}_{21}^\top$$

$$a_{22} = E\left[z_{l,m}, x_l^{InCom,miss} x_l^{InCom,miss\top}\right] + r_{l,m}\mu_m^{miss}\left(\mu_m^{miss}\right)^\top - 2E\left[z_{l,m}, x_l^{InCom,miss}\right]\left(\mu_m^{miss}\right)^\top$$

For a detailed derivation of $\mathbf{A}_l$, see Pervez and Lee (2018). After updating the GMM parameters, the next step is to estimate the phase signal values in the incomplete data set. This is done by using Gaussian Mixture Regression but with responsibilities calculated using all of the variables. For updating the phase variable, the utilized phase values in the responsibility terms come either from initialization or from the last M-step. For a given input variable $\mathbf{x}_i^{InCom,obs}$ and a given Gaussian distribution m, the expected value of $x_i^{InCom,miss}$ is defined by:

$$\mathcal{P}\left(x_i^{InCom,miss} | \mathbf{x}_i^{InCom,obs}, m\right) = \hat{x}_{i,m}$$

where

$$\hat{x}_{i,m} = \mu_m^{miss} + \boldsymbol{\Sigma}_m^{miss,obs}\left(\boldsymbol{\Sigma}_m^{obs}\right)^{-1}\left(\mathbf{x}_i^{InCom,obs} - \boldsymbol{\mu}_m^{obs}\right)$$

By considering the complete GMM, we get:

$$E\left(x_i^{InCom,miss} | \mathbf{x}_i^{InCom,obs}\right) = \sum_{l=1}^{k} r_{i,l}\hat{x}_{i,l}$$

$$\text{with} \quad r_{i,m} = \frac{\pi_m \mathcal{N}\left(\mathbf{x}_i^{InCom}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\right)}{\sum_{l=1}^{k} \pi_l \mathcal{N}\left(\mathbf{x}_i^{InCom}; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l\right)}$$

where $E(x_i^{InCom,miss} | \mathbf{x}_i^{InCom,obs})$ is the updated phase variable value of the ith data point in the incomplete data set.

2.4 Performance of the learned artificial agent

First, we compare the performance of our learned model with some DTW-based pre-alignment LfD strategies for learning from asynchronous trajectories. Figure 4e contains the result of LfD method presented in Akgun et al. (2012), Calinon et al. (2007). In their approach, the trajectories are first aligned by DTW and then encoded with a GMM. The GMM is fitted to the phase signal and the spatial data. Finally GMR is used for motion reproduction. When applying GMR for a rhythmic task, the data for the circular dimension (phase signal) is always mapped in the interval $\mu - \mathbb{X}_i$ and $\mu + \mathbb{X}_i$ for calculating the valid responsibilities. The drawback of this approach is that with phase signal as the only input, the GMM-GMR based encoding does not consider the current position of the end-effector when generating the motion. Due to the high level of inconsistency in the data, the learned trajectory partially completes the task, by inserting peg in three out of four holes, as shown in Fig. 4e.

The next model that we have considered for comparison with our own model is the DMP-based encoding of the demonstrations, with the phase signal and the forcing terms encoded by using a GMM (Alizadeh 2014). Again the forcing terms of different trajectories are first aligned with DTW. This approach also fails to reproduce the task, as shown in Fig. 4f. Because of dissimilarity in the forcing terms of different trajectories, the DTW fails to align them properly.

Figure 4g shows the result of our approach using DMP-based encoding which successfully reproduces the task due to the simultaneous alignment and encoding performed during the proposed EM steps. Another benefit is that it also considers the current position of the end-effector along with the phase signal when doing motion reproduction. The starting value of phase signal can easily be inferred by linearly generating samples of phase signal in between 0 and 2π and then the acceleration value $\dot{v}$ for each of them is calculated by using the DMP Eq. (1). The sample which yields the lowest value of sum of absolute accelerations of all DMPs is used as the starting point for integrating the canonical system.

3 Human-agent shared teleoperation

In teleoperation applications, there can be cases where some DOFs in a task are repetitive, while the remaining DOFs have non-repetitive motion. In such cases, we can introduce shared teleoperation instead of either human-only teleoperation or even autonomous execution of the task. Thus, the DOFs which have repetitive motion, can be encoded by LfD and executed autonomously through the learned artificial agent, and the remaining DOFs, which are either non-repetitive or highly uncertain, can be controlled by a human operator. That way, we can expand the application area of the proposed skills

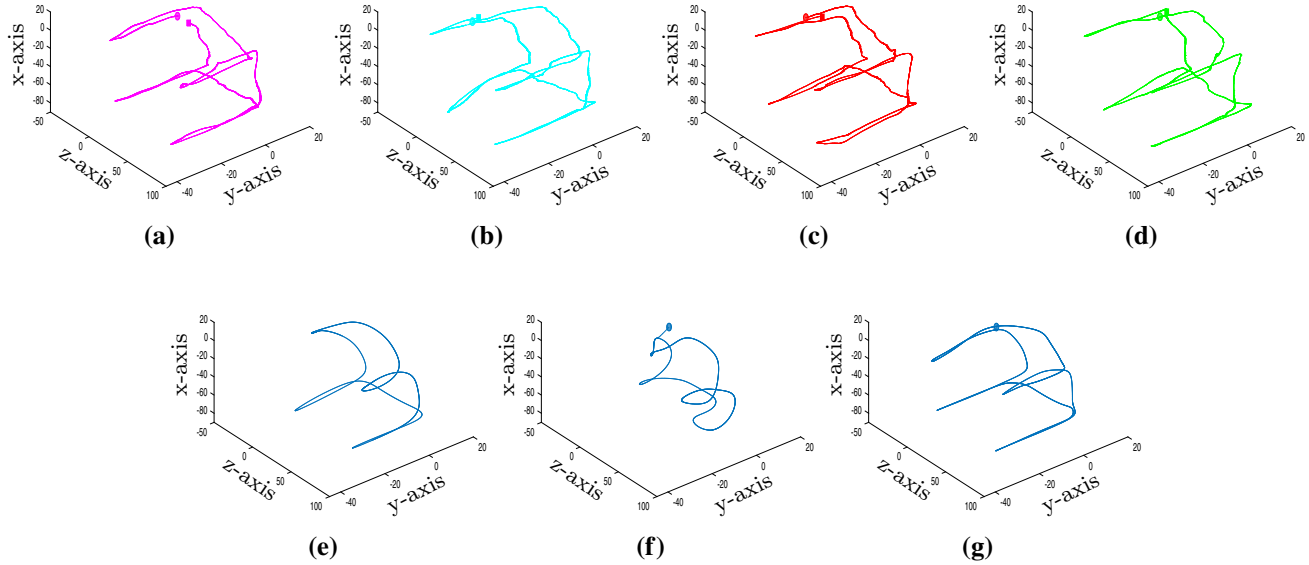


Fig. 4 All axes are in millimeters. The circle represents the starting point and square symbolizes the ending point of each demonstration trajectory. **a**–**d** Recorded motions of the peg-in-hole task for the four teleoperation demonstrations. **e** Reproduced motion with DTW + GMM based encoding of spatial data with GMR based motion reproduction, **f** reproduced motion with DMP model with DTW + GMM based encoding of forcing terms and **g** reproduced motion with the learned agent in our approach

learning method to more general tasks by removing the constraint that all DOFs should be fully repetitive.

However, in order to perform human-agent shared teleoperation, the main challenge is how to create synchronization between the generated motion of the learned agent and the motion command from the human operator on the fly. To address this problem, we have proposed two human-agent control sharing architectures in this paper: human-synchronized shared teleoperation and agent-synchronized shared teleoperation. In both of these architectures, the human operator and the agent exercise decoupled fixed control authority of some DOFs over the slave, as shown in Fig. 5. The merging and the dividing block in the figure combines the control inputs from both human operator and the agent into one consolidated velocity command to be sent to the slave over a network:

$$v_{slave} = \gamma \times v_{operator} + (1 - \gamma) \times v_{Agent} \tag{3}$$

In Eq. (3), v_{slave}, $v_{operator}$ and v_{Agent} are the velocities of the slave, operator, and agent, respectively. γ is the dominance value of operator in an axis. Also, for a given DOF, γ is either zero or one in our case, due to the decoupled control considered here, where zero and one symbolize no authority and complete authority over a respective axis, respectively. For multiple operators, the *dividing* part enables the division of the force feedback from the slave's environment between the operators controlling the slave. Since there is only one human operating in our case, all the haptic force is fed back to the only human operator.

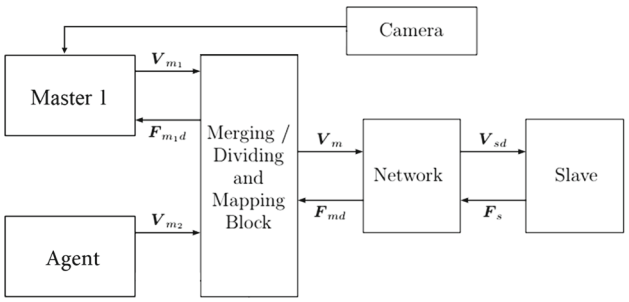


Fig. 5 Fixed control authority distribution: the division of control authority is based on Gromov et al. (2012), where velocity commands from both human and agent are merged before rendered onto the slave. The haptic force feedback from the slave's end is subsequently fed back to the human operator for perceptive guidance

3.1 Human-synchronized shared teleoperation

In the human-synchronized shared teleoperation, the motions of the human operator and that of the agent's are completely decoupled. This means that both the human and the agent provide velocity commands to the slave independently. In order to synchronize the overall motion of the task, the human adapts his/her motion to that of the agent's by utilizing his/her perceptive skills for a proactive alignment. The two key perceptive mediums used by human are sight and haptic (sense of touch). The agent's motion generated by the slave is fed back to the human in the form of a haptic force feedback, to guide the human with regard to the agent's and subsequently slave's motion. The control flow of the human-synchronized shared

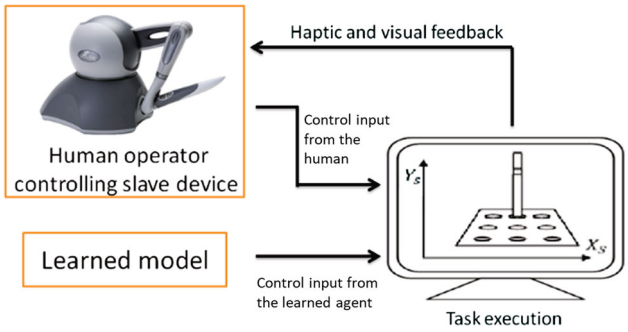


Fig. 6 In human-synchronized shared teleoperation architecture, decoupled control inputs from a human and agent are executed simultaneously. Human synchronizes his/her motion with the learned artificial agent by utilizing the haptic and visual feedback from the slave device

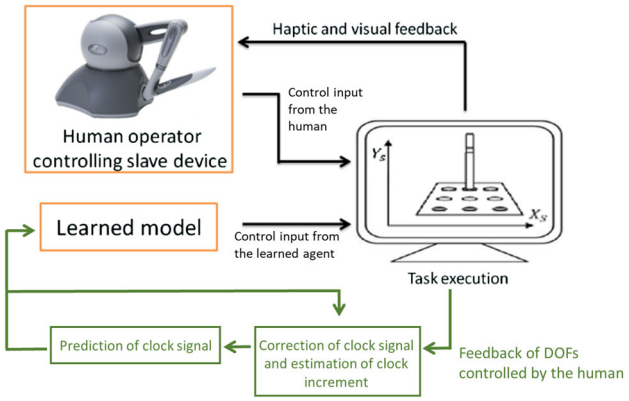


Fig. 7 In agent-synchronized shared teleoperation architecture, control inputs from a human and agent are synchronized through a clock signal that drives the learned model's motion. The value of this clock signal is actively updated based on the human operator's control input

teleoperation architecture can be visualized in Fig. 6, where the human operator utilizes the visual and haptic feedback coming from the slave device, to synchronize his/her motion while sharing control with the learned artificial agent.

3.2 Agent-synchronized shared teleoperation

In agent-synchronized shared teleoperation, unlike the human-synchronized one, the learned agent (DMP) actively participates in synchronizing its motion with that of the human operator's motion, as shown in Fig. 7. The basic idea in this architecture is to reproduce the learned model's motion based on human's control input. This is done by predicting the DMP's clock signal, for the given robot configuration, which is then utilized for generating the motion of the autonomous DOFs. Hence, the motion of the learned agent is dependent on the human operator's motion and clock feedback value from the last execution of motion.

The pseudocode for the agent-synchronized shared teleoperation is presented in Table 1. The initial clock signal value is predicted by a numerical maximum likelihood approach for the current end-effector configuration. Linearly spaced samples of phase variable are generated and the likelihood value of the generated samples along with the current end-effector configuration is evaluated for the learned GMMs. The sample which yields the maximum likelihood value is selected as the predicted starting value for the clock signal.

For reproducing the motion of the autonomous DOFs based on the learned model, the current clock signal value along with the model parameters is used to estimate the forcing term for the DMP, as in Eq. (2). The velocity command thus obtained after numerical integration of acceleration obtained from Eq. (1) is used in concatenation with the human control input for shared control. But for generating the motion of the autonomous DOFs for the next time step, it is important to synchronize them with the motion of the DOFs to be controlled by the human. For that the current clock signal value is re-estimated after motion command at each time step. To achieve this, an array of clock signal values is generated within a predefined window around the current clock value:

$$s_{\mathtt{Samp}} = [s_{\mathtt{cur}} - \delta \quad s_{\mathtt{cur}} \quad \ldots \quad s_{\mathtt{cur}} + N\delta]^{\top}$$

The generated clock values, along with the slave's current configuration (given both master and slave have no correspondence discrepancies) and learned model parameters, is run through a numerical maximum likelihood round along the DOF(s) controlled by the human:

$$\mathrm{Prob}_o = \mathcal{N}\left(s^o_{\mathtt{Samp}}; s^o_{\mathtt{Samp}}, \sigma_s^2\right) \prod_{d1=1}^{D1} \mathrm{Temp}_{d1}$$

where

$$\mathrm{Temp}_{d1} = \sum_{m=1}^{k} \pi_{m,d1} \mathcal{N}\left(\left[x_{d1} \quad s^o_{\mathtt{Samp}}\right]'; \boldsymbol{\mu}^{[2\,3]}_{m,d1}, \boldsymbol{\Sigma}^{[2\,3]}_{m,d1}\right)$$

The sample that maximizes the likelihood value is now used as the corrected value of clock signal for the next step:

$$s_{\mathtt{cur}} = s^{\hat{o}}_{Samp}.$$

where

$$\hat{o} = \underset{o}{\mathrm{argmax}}\{\mathrm{Prob}_o, o = 1, \ldots, N+1\}$$

The clock signal is an increasing function for a rhythmic DMP. So, if the predicted clock value is greater than the current clock value, we simply update the current value to it.

This corrected clock signal is now utilized for predicting the clock value for next time step. This is done by increasing

Table 1 Agent-synchronized shared teleoperation by using the learned GMMs

Input: $\boldsymbol{\Theta 1} = \{\{\pi_{m,d1}, \boldsymbol{\mu}_{m,d1}, \boldsymbol{\Sigma}_{m,d1}\}_{m=1}^{k}\}_{d1=1}^{D1}, \sigma_s, N2, N, \Delta, \delta$

$\boldsymbol{\Theta 2} = \{\{\pi_{m,d2}, \boldsymbol{\mu}_{m,d2}, \boldsymbol{\Sigma}_{m,d2}\}_{m=1}^{k}\}_{d2=1}^{D2}$

Initialize: $\text{Step}_{\text{sum}} = 0, l = 0$

Predict initial clock value

$s_{\text{Samp}} = [0 \;\; \delta \; 2\delta \; \ldots \; 2\pi]'$ ▷ Generate $N2$ samples

for $o = 1$ to $N2$ **do**

  for $d1 = 1$ to $D1$ **do** ▷ DOFs controlled by operator

$$\text{Temp}_{d1} = \sum_{m=1}^{k} \pi_{m,d1} \mathcal{N}([x_{d1} \; s^{o}_{\text{Samp}}]'; \boldsymbol{\mu}^{[2\ 3]}_{m,d1}, \boldsymbol{\Sigma}^{[2\ 3]}_{m,d1})$$

  end for

  for $d2 = 1$ to $D2$ **do** ▷ DOFs to be automated

$$\text{Temp}_{d2} = \sum_{m=1}^{k} \pi_{m,d2} \mathcal{N}([x_{d2} \; s^{o}_{\text{Samp}}]'; \boldsymbol{\mu}^{[2\ 3]}_{m,d2}, \boldsymbol{\Sigma}^{[2\ 3]}_{m,d2})$$

  end for

$$\text{Prob}_o = \mathcal{N}(s^{o}_{\text{Samp}}; s^{o}_{\text{Samp}}, \sigma_s^2) \prod_{d1=1}^{D1} \text{Temp}_{d1} \prod_{d2=1}^{D2} \text{Temp}_{d2}$$

end for

$\hat{o} = \underset{o}{\text{argmax}}\{\text{Prob}_o, o = 1, \ldots, N2\}$

$s_{\text{cur}} = s^{\hat{o}}_{Samp}$

while (GenerateMotion==1) **do**

  Generate motion

  $\mathcal{F}(s) \leftarrow \text{GMR}(s_{\text{cur}}, \boldsymbol{\Theta 2})$ ▷ For autonomous DOFs

  $\dot{v} = \tau\alpha_x(\beta_x(g - x) - v) + \tau a\mathcal{F}(s)$ ▷ Apply control input

  Re-estimate clock signal

  $s_{\text{Samp}} = [s_{\text{cur}} - \delta \;\; s_{\text{cur}} \; \ldots \; s_{\text{cur}} + N\delta]'$ ▷ Generate $N + 2$ samples

  for $o = 1$ to $N + 1$ **do**

    for $d1 = 1$ to $D1$ **do** ▷ DOFs controlled by operator

$$\text{Temp}_{d1} = \sum_{m=1}^{k} \pi_{m,d1} \mathcal{N}([x_{d1} \; s^{o}_{\text{Samp}}]'; \boldsymbol{\mu}^{[2\ 3]}_{m,d1}, \boldsymbol{\Sigma}^{[2\ 3]}_{m,d1})$$

    end for

$$\text{Prob}_o = \mathcal{N}(s^{o}_{\text{Samp}}; s^{o}_{\text{Samp}}, \sigma_s^2) \prod_{d1=1}^{D1} \text{Temp}_{d1}$$

  end for

$\hat{o} = \underset{o}{\text{argmax}}\{\text{Prob}_o, o = 1, \ldots, N + 1\}$

$l = l + 1$

  if $s^{\hat{o}}_{Samp} < s_{\text{cur}}$ **then**

$\text{Step}_{\text{sum}} = \text{Step}_{\text{sum}} + \Delta$

$s_{\text{cur}} = s_{\text{cur}} + \Delta$

  else

$\text{Step}_{\text{sum}} = \text{Step}_{\text{sum}} + (s^{\hat{o}}_{Samp} - s_{\text{cur}})$ ▷ Accumulate step size

$s_{\text{cur}} = s^{\hat{o}}_{Samp}$ ▷ Correct clock signal

$s_{\text{cur}} = s_{\text{cur}} + \frac{\text{Step}_{\text{sum}}}{l}$ ▷ Predict clock signal

  end if

end while

its value with an average increment of the clock signal values observed from the previous time steps:

$$s_{\text{cur}} = s_{\text{cur}} + \frac{\text{Step}_{\text{sum}}}{l}$$

where Step_{sum} accumulates all the corrections of the previous clock signals: $\text{Step}_{\text{sum}} = \text{Step}_{\text{sum}} + (s^{\hat{o}}_{Samp} - s_{\text{cur}})$. Thus the magnitude of the increment is also learned on the fly.

If the estimated value after maximum likelihood approach is less than the current value, we retain the current value as it is, and then increment this current value by a small predefined amount Δ for the next time step:

$$s_{\text{cur}} = s_{\text{cur}} + \Delta$$

For the discrete DMPs, the same approach can be applied but with decreasing values, instead of increasing values of the clock signal. In case of a collision or an external interference, the time development of the canonical system can also be automatically halted by utilizing the error in between the desired and the actual end-effector positions, as in Schaal et al. (2007).

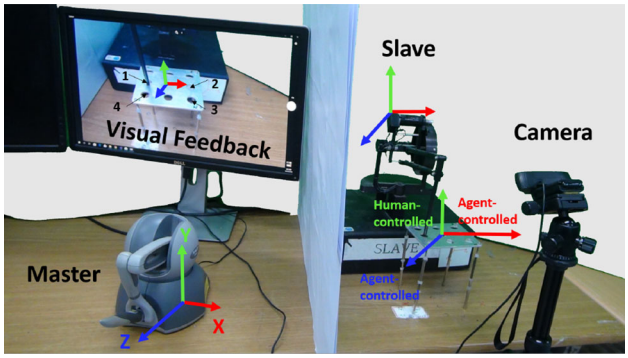


Fig. 8 Master (left) and slave (right) sides—perspective distortion due to the camera position on the slave side

4 Experimental results

4.1 Experimental setup

Our proposed algorithm is evaluated experimentally on a peg-in-hole task with a master-slave teleoperation system. The setup consists of a 3-DOF SensAble PHANToM Omni as a master device and a 3-DOF PHANToM Premium 1.5A as a slave device. The pair of master and slave devices run on the same computer. A web-camera streams the visual feedback from the slave environment to the human operator. The camera position and the lack of depth information in the two dimensional camera images introduce a perspective distortion which inhibits a clear visual perception for the operator. An aluminum plate with 4 holes is placed under the slave to serve as the task rig for the peg-in-hole task, as illustrated in Fig. 8. For the experimental evaluation of shared teleoperation, the human operator observes the visual feedback on the monitor in front and controls the motion of the slave device in y-axis only, whereas the artificial agent controls both the x and z axes of the motion. One execution cycle constitutes insertion of the slave robot end-effector into four holes of the task rig in clockwise direction, while starting and ending above the same hole. The operator's master device receives

a haptic force feedback from the slave device. This provides a kinesthetic coupling between the artificial agent and the human operator. The haptic feedback increases the awareness of the operator about the motion of the agent, thus easing the task execution for cooperative teleoperation.

For encoding the DMP, a human operator performs four teleoperated demonstrations of the peg-in-hole task. During each cycle, the Cartesian coordinates of the slave's end-effector's position are recorded for utilizing as an input to the learning algorithm. The demonstrated trajectories can be visualized in Fig. 4a–d.

4.2 Simulation of agent-synchronized shared teleoperation

We verified the performance of our proposed learned model as an artificial agent in a computer simulated shared teleoperation setting. For shared teleoperation simulations, the values of N, $N2$, δ and Δ are set to 4000, 1300, 2×10^{-4} and 10^{-3} respectively. Furthermore, an operator is asked to demonstrate a single motion of the peg-in-hole task as a reference trajectory. Now, we use this reference trajectory along with the learned agent in order to simulate a shared teleoperation architecture. It is to be noted here that the reference trajectory is passed as an input in an open loop manner because the haptic and visual feedback in Fig. 7 is missing when performing shared teleoperation in the simulation. We only pass the motion of y-axis from the reference trajectory as human input while that of x and z axes are autonomously generated. The learned model successfully completes the peg-in-hole task as shown by the generated motion in Fig. 9a while the evolution of its corresponding clock signal is shown in Fig. 9c.

Now we evaluate the performance for temporal variation in the execution of the motion as well as for the control of multiple DOFs by the operator. We replay the recorded motion at different speeds, in order to check the synchronization capability of the learned agent. Figure 9b shows the result of successful motion execution, when x and y axes of the motion come from the reference trajectory while the motion of z-axis is autonomously generated. In this simulation, we significantly decreased the speed of execution of the motion of DOFs controlled by the reference trajectory. As one can see from Fig. 9d with Fig. 9c, the generated clock signal also increases relatively slowly. Figure 9e shows the result of successful motion execution in another simulation setting, where the motion of y and z axes comes from the reference trajectory, the motion of x-axis is autonomously executed, and the speed of execution of the motion of DOFs controlled by the reference trajectory is significantly increased. The generated clock signal adapts accordingly by increasing faster. This can be observed by comparing Fig. 9g with Fig. 9c. In all of the simulations of shared teleoperation, the generated clock signal does not simply increase linearly but small adjustments can be observed in the generated clock values for synchronizing the motion of autonomous DOFs with the DOFs controlled by the reference trajectory.

The generated motion can also switch from shared teleoperation to fully autonomous mode and vice versa. If an operator wants to rest or is confident of correct motion execution by the artificial agent then they can enable the fully autonomous mode. Figure 9f shows the result when half of the motion is generated in shared teleoperation setting while the second half of the motion is generated in fully autonomous mode. For full autonomy, the clock signal increment becomes linear, as is shown in Fig. 9h. Due to the use of a dynamical system in the DMP model, the control transitions smoothly from shared teleoperation to fully autonomous mode, as shown in Fig. 9f.

4.3 Experimental evaluation of human-agent shared teleoperation

Eight engineering students, both male and female, ages ranging from 23 to 33, participated in performing 4 trials for each of the three types of experiments - human-only teleoperation, human-synchronized shared teleoperation and agent-synchronized shared teleoperation. The subjects were given sufficient time to familiarize themselves with the setup of the three types of experiments. The three experiments (with contiguous 4 trials of each) were performed by the subjects in the same order as mentioned above. None of the subjects had any prior experience in teleoperation, let alone with Phantom devices. Thus, they were provisioned with preliminary familiarization trials prior to each of the three types of experiments. The subjects indicated of their readiness to perform each experiment after sufficient familiarization with the setup, of their own accord. The subjects observe the visual feedback from the camera placed at the slave's end. During the shared teleoperation experiments, both the agent and the human operator have complete control authority over their respective axes - human controls y-axis motion only, whereas the agent controls x and z axes motion. In order to evaluate the performance of the two proposed shared teleoperation approaches against the human-only teleoperation, we recorded the execution time, the rate of collision and the overall workload index (NASA-TLX Hart and Staveland 1988).

Execution time Each execution cycle of the peg-in-hole task comprised of moving to the four holes of the rig and inserting the end-effector into them. Among the three teleoperation architectures, the agent-synchronized shared teleoperation on average enabled the fastest motion executions, as shown in Fig. 10. This is due to the prediction of the next clock signal based on human operator's input. This allows the execution speeds to be variable for different subjects, and hence,

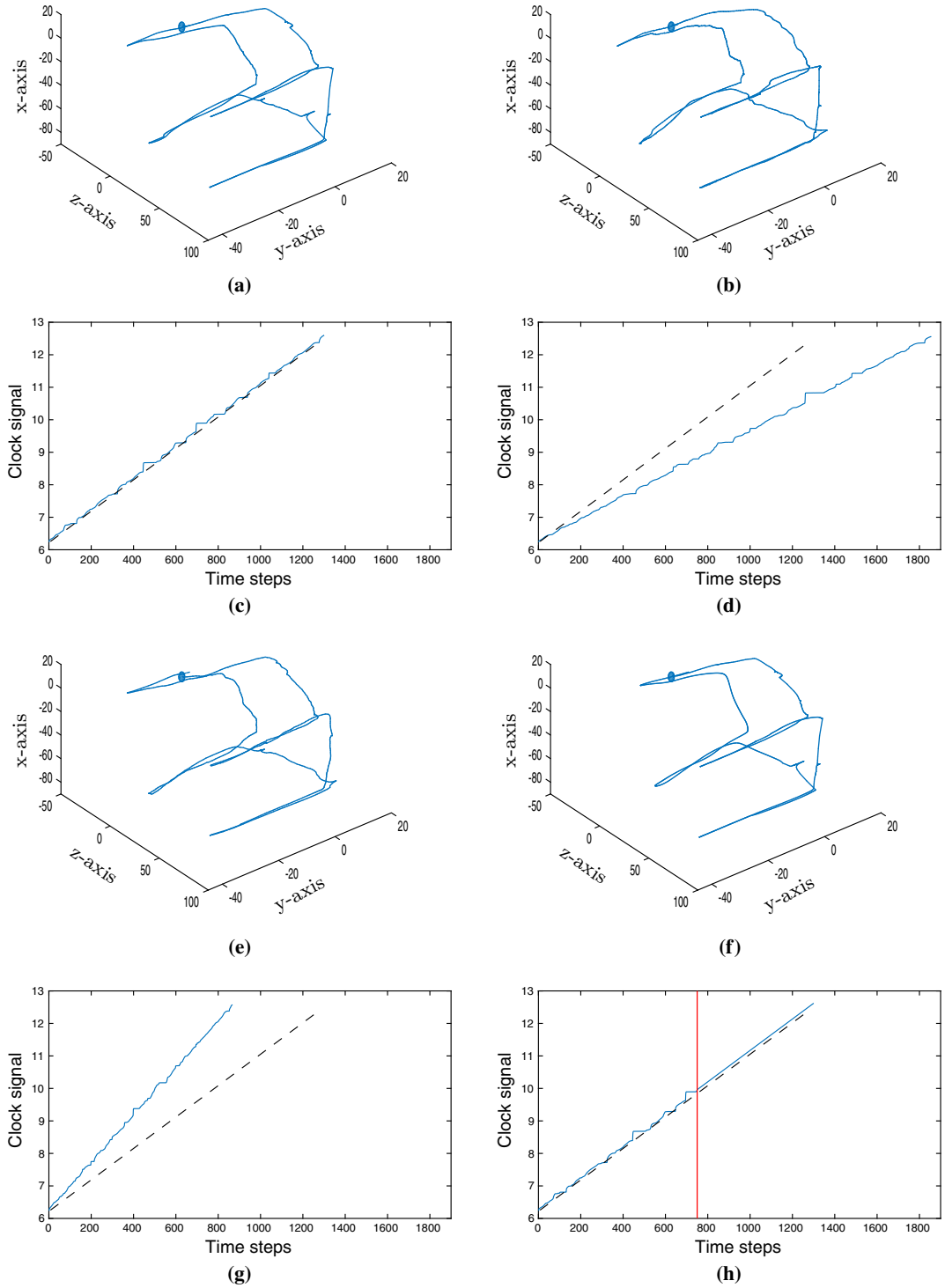


Fig. 9 **a** y-axis of the recorded motion is used as the operator's control input while the motion of x and z axes is autonomously generated. **b** x and y axes of the recorded motion is used as the operator's control input while the motion of z-axis is autonomously generated. The speed of operators motion is significantly slower than the demonstrated motions in this experiment. **c** Generated clock signal corresponding to **a** for reference, the black dashed line in **b**, **d**, **g** and **h** shows the clock signal for completely autonomous execution of the task without shared teleoperation control, as in Fig. 4g. **d** Generated clock signal corresponding to **b**. **e** y and z axes of the recorded motion is used as the operator's input while the motion of x-axis is autonomously generated. The speed of operators motion is significantly faster than the demonstrated motions in this experiment. **f** y-axis of the recorded motion is used as the operator's control input while the motion of x and z axes is autonomously generated during first half of the motion. The later half of the motion is generated in fully autonomous mode. **g** Generated clock signal corresponding to **e**. **h** Generated clock signal corresponding to **f**. The switching point from shared teleoperation to full autonomy is indicated by the red line (Color figure online)

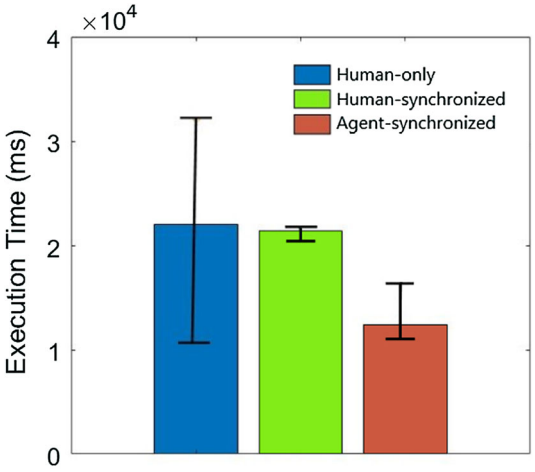


Fig. 10 Average execution time of the peg-in-hole task for the two shared teleoperation approaches compared with the human-only teleoperation experiments. Error bars indicate the minimum and maximum values

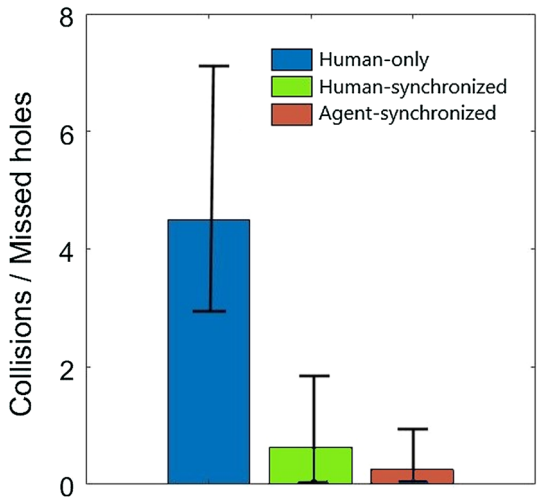


Fig. 11 Average rate of collision/missed holes of the peg-in-hole task for the two shared teleoperation approaches compared with the human-only teleoperation experiments. Error bars indicate the minimum and maximum values

brisked up at points where human intended on moving faster, whereas slowed down (less increment in the clock value) where human wanted to take more time at critical points of the task. Moreover, in the human-synchronized shared teleoperation, the execution time, measured in milliseconds, was almost constant throughout the course of experimentation across all subjects, with only minor variations due to the noise introduced by the subjects (time taken by subjects to stop the recording manually). This is because the execution time in this setting is mainly governed by the artificial agent to complete the task. Finally, the human-only teleoperation posed the most difficulty and labour for the subjects, primarily due to the visual distortion and zero haptic assistance from any external agents, thereby making it the most slowest of the three teleoperation modes for the same task.

Rate of collision and missed holes The three teleoperation approaches were evaluated based on the rate of collision and missed holes recorded while experimenting. By definition, *collision* here means involuntarily hitting the rig with the slave's end-effector anywhere apart from the hole itself (including the walls surrounding the holes). It also incorporates dismantling the rig completely due to a delayed or brisk move of the human controlled axis. *Missing* simply symbolizes losing the opportunity to insert the slave end-effector into the hole of the rig as per the task's requisite steps. Additionally, the maximum number of collisions incurred by each subject throughout the four trials of each teleoperation setting was averaged across all subjects. Figure 11 clearly shows that the subjects had much less collisions and missed holes with the task rig in the two shared teleoperation approaches, as compared to the human-only teleoperation. The reason being that in the human-synchronized control, the agent controlled the slave to traverse through the rig as per the prescribed steps of motion and would only station itself right above each hole, which gives the operator the time as well as the ease to just insert the end-effector when it is at that point. Similarly, the agent-synchronized shared teleoperation showed best performance (no missed holes), since the slave only moved to the next step based on the human operator's current position, thereby synchronizing its speed with the human's speed at every instant. Whereas, the human-only teleoperation incurred the most collisions and holes' misses.

Workload index The NASA-TLX subjective assessment in terms of the overall workload is shown in Fig. 12 for each of the teleoperation methods. As a consequence of all the satisfactory results in the previously mentioned metrics, both of the proposed shared teleoperation approaches have ranked subjectively to have a lower index of workload. The subjects reported to have felt complete ease at having to only care about inserting the end-effector into the holes (y-axis motion) while the agent controlled the rest of the motion (x and z axes). But, most of them rated the human-only shared teleoperation as being the most *forgiving* of all in terms of collision rectification. But on the flip side, they felt frustration when they could not correctly perceive the slave's environment through visual feedback which they attributed to their poor performance in the human-only teleoperation architecture.

Between the two shared teleoperation settings, the human-synchronized one takes preeminence over its agent-synchronized counterpart slightly. This precedence, even though by a slight margin, could perhaps be accounted for due to an obvious factor. The human subjects motion had no affect on the agent's motion in the human-synchronized mode, as the agent's motion is executed in an open loop man-

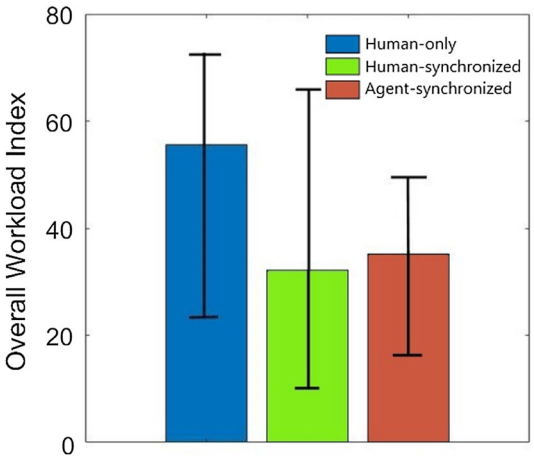


Fig. 12 Overall workload index of the peg-in-hole task for the two shared teleoperation approaches compared with the human-only teleoperation experiments. Error bars indicate the minimum and maximum values

ner. Whereas, in the agent-synchronized mode, the human operators had to keep the robot's end-effector within the range of the demonstrated trajectories so that the agent could correctly infer the clock signal's value.

Both the human-agent shared teleoperation modes served to be least laborious because of the task sharing with the learned artificial agent.

Phase signal There is no clock signal involved in human-only teleoperation while the clock signal increases linearly in human-synchronized shared teleoperation. For the value of the generated clock signal s during the agent-synchronized shared teleoperation experiments, the increment of clock signal adapts based on human operator's input, as it is also shown in the simulation. The force feedback has no direct effect on the autonomous DOFs, as only the human operator receives the force feedback. The autonomous DOFs are solely driven by the DMP, while its speed is regulated by the motion of the human operator. Since DMP itself is inherently a stable system, no instability issues arise in the shared teleoperation architecture.

The value of s, that was generated on the fly based on the current position of the master device, does not simply increase linearly, but in fact, adjustments can be observed in the generated clock values for synchronizing the motion of the autonomously generated DOFs with the DOFs controlled by the human operator. This phenomenon is visualized in Fig. 13 which shows the generated clock signal for an agent-synchronized shared teleoperation experiment. It can be observed that the agent is proactive in its approach to synchronize with human motion in real time.

Summary of experiments Hence, conclusively, the agent-synchronized based teleoperation mode was evaluated to be the one with the most accurate execution of task while taking

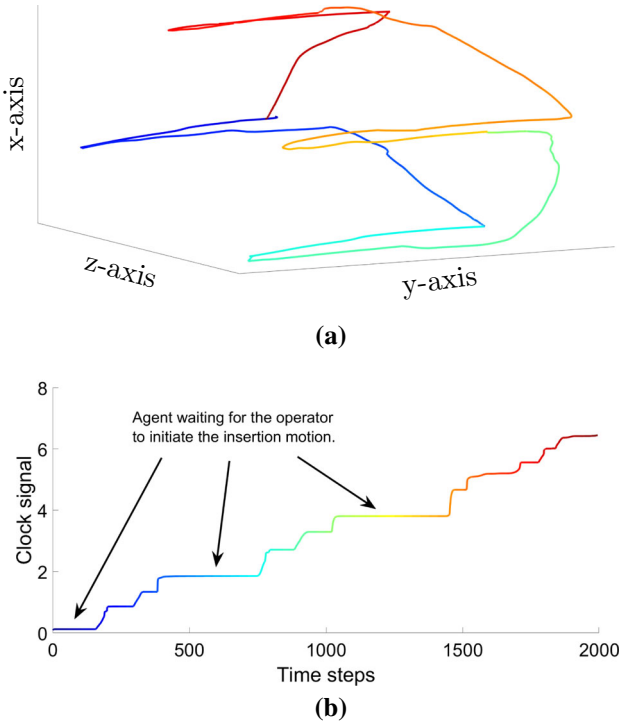


Fig. 13 **a** Generated motion for *one* complete cycle of the peg-in-hole task and **b** its corresponding clock signal, for the agent-synchronized shared teleoperation experiment

the least time to perform the experiments across all subjects. Whereas human-synchronized shared control came out to be the least burden-some of all.

5 Conclusion

LfD provides a way to automate a repetitive task by utilizing human teleoperated demonstrations. LfD can be challenging for teleoperation, due to the large temporal and spatial variations in teleoperated demonstrations. To address this problem, we proposed an EM-based DMP approach for simultaneously aligning and encoding the motion trajectories. The proposed approach shows superior performance as compared to DTW based pre-alignment and then encoding of motion trajectories. It has also been shown that the learned model can be utilized in human-agent shared teleoperation setup, of which we have proposed two variants - human-synchronized, and agent-synchronized shared teleoperation. The agent in both approaches is learned through teleoperated demonstrations. The proposed approaches show significant performance improvement given visual perspective distortion which deteriorated the efficiency significantly when utilizing the human-only teleoperation.

Furthermore, between the two proposed shared teleoperation approaches, the agent-synchronized one reduces the

total execution time of the peg-in-hole task. Also, it has a slightly more accurate execution of the task than the human-synchronized one. However, it possess a slightly higher workload for an operator, since the operator has to follow the demonstrated intervals and has to also cooperate with an active agent. Moreover, due to DMP-based LfD approach , our generated artificial agent can scale up and/or down its meta parameters, like spatial or temporal scaling of the motions.

References

Akgun, B., & Subramanian, K. (2011). Robot learning from demonstration: kinesthetic teaching vs. teleoperation. Unpublished manuscript.

Akgun, B., Subramanian, K., & Thomaz, A. (2012). Novel interaction strategies for learning from teleoperation. In *AAAI fall symposium series* (pp. 2–9).

Alizadeh, T. (2014). *Statistical learning of task modulated human movements through demonstration*. Ph.D. thesis, Istituto Italiano di Tecnologia.

Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, *57*(5), 469–483.

Billard, A., Calinon, S., Dillmann, R., & Schaal, S. (2008). Robot programming by demonstration. *Springer handbook of robotics* (pp. 1371–1394). Berlin: Springer.

Bukchin, J., Luquer, R., & Shtub, A. (2002). Learning in tele-operations. *IIE Transactions*, *34*(3), 245–252.

Calinon, S., Guenter, F., & Billard, A. (2007). On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, *37*(2), 286–298.

Calinon, S., & Lee, D. (2018). Learning control. In P. Vadakkepat & A. Goswami (Eds.), *Humanoid robotics: A reference*. Berlin: Springer.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society Series B (Methodological)*, *39*, 1–38.

Dragan, A. D., & Srinivasa, S. S. (2012). Assistive teleoperation for manipulation tasks. In *Proceedings of the seventh annual ACM/IEEE international conference on human-robot interaction* (pp. 123–124). ACM.

Fischer, K., Kirstein, F., Jensen, L. C., Kr, N., Kukli, K., aus der Wieschen, M., & avarimuthu, T. (2016). A comparison of types of robot control for programming by demonstration. In *ACM/IEEE international conference on human-robot interaction (HRI)* (pp. 213–220).

Ghahramani, Z., & Jordan, M. I. (1994). Supervised learning from incomplete data via an EM approach. *Advances in neural information processing systems* (Vol. 6). Princeton: Citeseer.

Gromov, B., Ivanova, G., & Ryu, J. H. (2012). Field of view deficiency-based dominance distribution for collaborative teleoperation. In *12th international conference on control, automation and systems (ICCAS), 2012* (pp. 1990–1993). IEEE.

Hart, S. G., & Staveland, L. E. (1988). Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Advances in psychology* (Vol. 52, pp. 139–183). Amsterdam: Elsevier.

Havoutis, I., & Calinon, S. (2019). Learning from demonstration for semi-autonomous teleoperation. *Autonomous Robots*, *43*(3), 713–726. https://doi.org/10.1007/s10514-018-9745-2.

Hokayem, P. F., & Spong, M. W. (2006). Bilateral teleoperation: An historical survey. *Automatica*, *42*(12), 2035–2057.

Hu, K., Ott, C., & Lee, D. (2014). Online human walking imitation in task and joint space based on quadratic programming. In *IEEE international conference on robotics and automation* (pp. 3458–3464).

Khansari-Zadeh, S. M., & Billard, A. (2011). Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, *27*(5), 943–957.

Kober, J., & Peters, J. (2010). Imitation and reinforcement learning. *IEEE Robotics & Automation Magazine*, *17*(2), 55–62.

Lee, D., & Ott, C. (2010). Incremental motion primitive learning by physical coaching using impedance control. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 4133–4140).

Medina, J., Lee, D., & Hirche, S. (2012). Risk-sensitive optimal feedback control for haptic assistance. In *2012 IEEE international conference on robotics and automation* (pp. 1025–1031). IEEE.

Ott, C., Lee, D., & Nakamura, Y. (2008). Motion capture based human motion recognition and imitation by direct marker control. In *IEEE-RAS international conference on humanoid robots* (pp. 399–405).

Pervez, A., Ali, A., Ryu, J. H., & Lee, D. (2017). Novel learning from demonstration approach for repetitive teleoperation tasks. In *World haptics conference (WHC), 2017 IEEE* (pp. 60–65). IEEE.

Pervez, A., & Lee, D. (2015). A componentwise simulated annealing em algorithm for mixtures. In *Joint German/Austrian conference on artificial intelligence (KI)* (pp. 287–294).

Pervez, A., & Lee, D. (2018). Learning task-parameterized dynamic movement primitives using mixture of gmms. *Intelligent Service Robotics*, *11*(1), 61–78.

Peternel, L., & Babic, J. (2013). Humanoid robot posture-control learning in real-time based on human sensorimotor learning ability. pp. 5329–5334. https://doi.org/10.1109/ICRA.2013.6631340.

Peternel, L., Öztop, E., & Babic, J. (2016). A shared control method for online human-in-the-loop robot learning based on locally weighted regression. In *IROS* (pp. 3900–3906). IEEE.

Peternel, L., Petriăź, T., & Babiăź, J. (2018). Robotic assembly solution by human-in-the-loop teaching method based on real-time stiffness modulation. *Autonomous Robots*, *42*(1), 1–17. https://doi.org/10.1007/s10514-017-9635-z.

Peters, R. A., Campbell, C. L., Bluethmann, W. J., & Huber, E. (2003). Robonaut task learning through teleoperation. In *IEEE international conference on robotics and automation* (pp. 2806–2811).

Power, M., Rafii-Tari, H., Bergeles, C., Vitiello, V., & Yang, G. Z. (2015). A cooperative control framework for haptic guidance of bimanual surgical tasks based on learning from demonstration. In *IEEE international conference robotics and automation (ICRA)* (pp. 5330–5337).

Rozo, L., Jiménez, P., & Torras, C. (2013). A robot learning from demonstration framework to perform force-based manipulation tasks. *Intelligent Service Robotics*, *6*(1), 33–51.

Rozo, L., Jimenez Schlegl, P., & Torras, C. (2010). Sharpening haptic inputs for teaching a manipulation skill to a robot. In *IEEE international conference on applied bionics and biomechanics* (pp. 331–340).

Rozo, L. D., Jiménez, P., & Torras, C. (2010). Learning force-based robot skills from haptic demonstration. *CCIA* (pp. 331–340). Washington, DC: CCIA.

Sanguansat, P. (2012). Multiple multidimensional sequence alignment using generalized dynamic time warping. *WSEAS Transactions on Mathematics*, *11*(8), 668–678.

Saveriano, M., An, S., & Lee, D. (2015). Incremental kinesthetic teaching of end-effector and null-space motion primitives. In *IEEE international conference on robotics and automation* (pp. 3570–3575).

Schaal, S. (2006). Dynamic movement primitives—a framework for motor control in humans and humanoid robotics. *Adaptive motion of animals and machines* (pp. 261–280). Berlin: Springer.

Schaal, S., Mohajerian, P., & Ijspeert, A. (2007). Dynamics systems vs. optimal controla unifying view. *Progress in Brain Research*, *165*, 425–445.

Schmidts, A. M., Lee, D., & Peer, A. (2011). Imitation learning of human grasping skills from motion and force data. In *International conference on intelligent robots and systems (IROS), 2011 IEEE/RSJ* (pp. 1002–1007). IEEE.

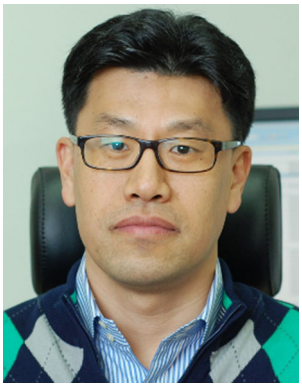
Stulp, F., Raiola, G., Hoarau, A., Ivaldi, S., & Sigaud, O. (2013). Learning compact parameterized skills with a single regression. In *13th IEEE-RAS international conference on humanoid robots (humanoids)* (pp. 417–422).

Usmani, N. A., Kim, T. H., & Ryu, J. H. (2015). Dynamic authority distribution for cooperative teleoperation. In *International conference on intelligent robots and systems (IROS), 2015 IEEE/RSJ* (pp. 5222–5227). IEEE.

Yang, J., Xu, Y., & Chen, C. S. (1994). Hidden Markov model approach to skill learning and its application to telerobotics. *IEEE Transactions on Robotics and Automation*, *10*(5), 621–631.

Affan Pervez did B.E. Mechatronics Engineering (2005–2009) from National University of Sciences and Technology, Pakistan and Masters in Machine Learning (2010–2012) from Kungliga Tekniska Hogskolan (KTH), Sweden. He completed his PhD (2013–2018) in Electrical Engineering from human-centered Assistive Robotics group, Technical University of Munich, Germany. His areas of interest include machine learning, robotics, teleoperation, imitation learning and deep learning.

Hiba Latifee did her B.E.(2012–2016) in Computer and Information Systems Engineering from NED University of Engineering and Technology (NEDUET), Pakistan. She is currently a Masters student, pursuing a Mechanical Engineering major from KOREATECH, Cheonan, South Korea. Her research areas include control and optimization, teleoperation, learning from demonstration and incremental learning.



Jee Hwan Ryu received the B.S. degree in mechanical engineering from Inha University, Incheon, South Korea, in 1995, and the M.S. and Ph.D. degrees in mechanical engineering from Korea Advanced Institute of Science and Technology, Daejon, South Korea, in 1995 and 2002, respectively. He is a Professor in the Department of Mechanical Engineering, KOREATECH, Cheonan, South Korea. His research interests include haptics, telerobotics, exoskeletons, and autonomous vehicles.

Dongheui Lee obtained her B.S. (2001) and M.S. (2003) degrees in mechanical engineering at Kyung Hee University, Korea and a PhD degree (2007) from the department of Mechano-Informatics, University of Tokyo, Japan. She was a research scientist at the Korea Institute of Science and Technology (KIST) (2001–2004), Project Assistant Professor at the University of Tokyo (2007–2009), and Assistant Professor at TUM (2009–2017). Since 2017, she is Associate Professor at the TUM Department of Electrical and Computer Engineering, and director of a Human-centered assistive robotics group at the German Aerospace Center (DLR). In recognition of her research achievements, she was awarded a Carl von Linde Fellowship at the TUM Institute for Advanced Study (2011) and a Helmholtz professorship prize (2015).