Interactive Imitation Learning of Bimanual Movement Primitives

Giovanni Franzese, *Graduate Student Member, IEEE*, Leandro de Souza Rosa, Tim Verburg, Luka Peternel, *Member, IEEE*, and Jens Kober, *Senior Member, IEEE*

***Abstract*—Performing bimanual tasks with dual robotic setups can drastically increase the impact on industrial and daily life applications. However, performing a bimanual task brings many challenges, such as synchronization and coordination of the single-arm policies. This article proposes the safe, interactive movement primitives learning (SIMPLe) algorithm, to teach and correct single or dual arm impedance policies directly from human kinesthetic demonstrations. Moreover, it proposes a novel graph encoding of the policy based on Gaussian process regression where the single-arm motion is guaranteed to converge close to the trajectory and then toward the demonstrated goal. Regulation of the robot stiffness according to the epistemic uncertainty of the policy allows for easily reshaping the motion with human feedback and/or adapting to external perturbations. We tested the SIMPLe algorithm on a real dual-arm setup where the teacher gave separate single-arm demonstrations and then successfully synchronized them only using kinesthetic feedback or where the original bimanual demonstration was locally reshaped to pick a box at a different height.**

***Index Terms*—Bimanual manipulation, impedance control, interactive imitation learning (IIL), movement primitives (MPs).**

I. Introduction

Modern society is faced with the lack of workforce in various repetitive jobs, such as reshelving products in supermarkets or handling heavy luggage in airports. Robots appear to be the most promising solution to mitigate the negative effects of the declining workforce and perform these various complex tasks [1]. To work in variable and unstructured environments, robots must be dexterous and intelligent to quickly learn the job while interacting safely with other robots, objects, and humans. However, traditional task-specific robot programming by experts fails to achieve such dexterity and intelligence due to the time-consuming process and poor adaptability of tailored solutions.

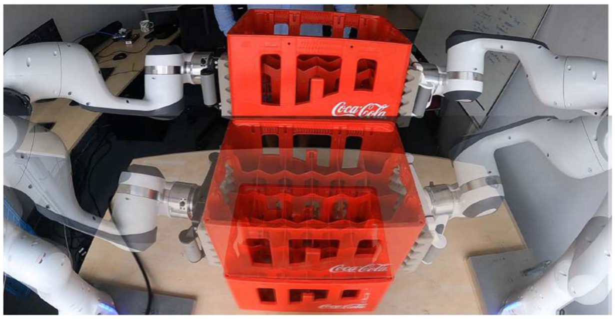

Fig. 1. Example of possible application of bimanual manipulation: performing stacking of crates.

Recent advances in machine learning, namely in learning from demonstration (LfD), have enabled robots to learn directly from (nonexpert) human demonstrations without needing complex task-specific programming or long and dangerous exploration. Branching from LfD, interactive imitation learning (IIL) approaches [2] allow human teachers to provide interactive demonstrations and corrections to the robot, exploring the advantage that the latter is much more sample efficient than the former, thus reducing the burden on the human teacher. IIL methods cover many feedback modalities (e.g., correction, evaluative, and qualitative), can be used to learn different models (e.g., policies and objective functions), and leverage several function approximators [e.g., neural networks (NNs), dynamic movement primitives (DMPs), hidden Markov models (HMMs), and Gaussian process (GPs)].

While tasks that require only one arm have been explored extensively in the literature, more complex tasks, which require a bimanual setup have only recently been targeted. Among such tasks, picking large objects in unstructured environments (Fig. 1) [1], assisting the elderly [3], [4], surgery tasks [5], or complex assembly tasks [6] are shown to require dexterous bimanual setups. Factory assembly, logistics, and household applications of bimanual robots have been known for decades [7], [8].

Manuscript received 3 October 2022; revised 21 February 2023 and 23 May 2023; accepted 13 June 2023. This work was supported by the European Research Council Starting Grant TERI Teaching Robots Interactively, under Project 804907. Recommended by Technical Editor A. Alanis and Senior Editor J. Ueda. *(Corresponding author: Giovanni Franzese.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by TU Delft Human Research Ethics Committee under Application No. 1729, and performed in line with the Helsinki Declaration on Human Subject Research.

The authors are with the Department of Cognitive Robotics, 3mE,Delft University of Technology, 2628 CN Delft, The Netherlands (e-mail: g.franzese@tudelft.nl; l.desouzarosa@tudelft.nl; t.verburg-1@student.tudelft.nl; l.peternel@tudelft.nl; j.kober@tudelft.nl).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TMECH.2023.3295249.

Digital Object Identifier 10.1109/TMECH.2023.3295249

However, the increased number of degrees of freedom (DoFs) (the curse of dimensionality) implies an increased teaching complexity and the necessity of skilled human teachers who knows how to interface with the bimanual robotic platform.

In this article, we contribute with the safe interactive movement primitive learning (SIMPLe) algorithm and propose the following.

1) The design of a bimanual impedance controller with variable Cartesian stiffness; safety constraints on the maximum applicable force and execution velocity are also formulated.
2) A novel movement primitive (MP) formulation that allows efficiently learning long horizon tasks from a single demonstration and executes the motion in a reactive way.
3) Efficient corrections of the robot's policy directly from kinesthetic feedback, allowing for fine-tuning the demonstrations. Thanks to this, the user can show single arms' trajectories and fine-tune them when transferring the policies onto a bimanual task.

To validate the proposed method, we conducted a series of experiments. The first three are technical experiments related to the main contributions that highlight and test different functionalities of the method. The last two are supplementary user studies to evaluate the type of data input for the proposed by comparing two human demonstration approaches and to evaluate giving corrections compared to giving new demonstrations. These additional insights can provide a better understanding of the input data generation method and adjustments of the robot's skill for bimanual cases.

II. Related Works

A. Bimanual Teaching Frameworks

Like with single-arms, preplanning, and manual coding of multiarm manipulation is a tedious process. An alternative is learning from *human demonstrations*, where a user can guide the robot on how to execute the desired tasks. However, when the user controls the dual (or multiple) robot setup, the physical and cognitive load increases drastically. Using priors, shared control, or task scaffolding, i.e., dividing the teaching into smaller parts, can substantially decrease the demonstrator workload and make the teaching easier and the learning faster.

Recent works on the control side of bimanual manipulation leverage *shared control* strategies for reducing the burden of *teleoperated* bimanual tasks. For example, [9] proposes a shared controller for helping the user to perform bimanual manipulation: it maintains the manipulators' relative position (or orientation) while the user controls the translations or rotations. Similarly, [10] classifies human demonstrations in four teaching modalities: self handover, one-hand fixed, one-hand seeking, and fixed offset; when performing teleoperation, a trained classifier detects the most likely modality and adapts the constraints of the bimanual controller accordingly.

On the side of *shared control*, [11] extends the Roboturk platform by having each arm teleoperated by a different teacher, reducing the cognitive load and enabling teaching tasks with more than two arms. Moreover, ongoing research [12] presents a controller that enables inputs from a teleoperating user and local kinesthetic perturbations. In our work, we focus on teaching bimanual policies from a single human teacher by teaching single-arm policies independently and then interactively reshaping them for successful coordination or adaptation to a new scenario. The goal is to enable nonexpert users to teach complex bimanual tasks.

B. Bimanual Coordination Policies

During autonomous execution, disturbing one of the arms in a detached bimanual system can break the synchrony of the movements, making it necessary to provide both movement recovery and resynchronization capabilities. The way the policy is encoded, e.g., time-dependent versus position-dependent, or the chosen function approximation, e.g., a DMPs, HMM, and GP [13] can change the disturbance rejection of the robot.

To this end, the method in [14] uses a prior on the relative position of the two manipulators and a timing dependence in the HMM formulation to synchronize the movement of arm manipulators. Other approaches propose to create a "leader and follower" movement by adding a coupling term [15], a regulation term [16], or a deterministic encoding of trajectories with DMPs [17]. Alternatively, the epistemic uncertainty of GPs can be used for switching the behavior of the arms from follower to leader (and vice-versa) [18]. This leader–follower learning paradigm makes the system react differently according to which arm is perturbed. Alternatively, the task prior symmetry can be used for easily encoding and synchronizing the task. For example, [19] proposes a bimanual policy for picking and throwing nonstationary objects by learning a *symmetric* dynamical system (DS) policy. In this case, perturbing any of the two arms would always make the other react.

Other approaches focus on achieving synchrony and coordination by segmenting the trajectories and reproducing them in sequence or according to a *hierarchical* representation of the task. The advantage of such approaches is that the sequencing provides an implicit synchronization on a higher level, making the lower level problem easier. A common approach for this scheme is to learn policies for performing predefined subtasks, and a higher level policy that creates a sequence from demonstrations [20], [21]. Alternatively, the task can have a predefined structure of subtasks based on heuristics, and synchrony is achieved with a subtask scheduler [22]. Segmentation has also been used for deep-learning bimanual tasks in [23], where lower level policies are learned for each segment and higher ones for sequencing them. In this direction, [24] proposes a framework for multiarm task-space control with smooth transitions from independent behaviors, e.g., when reaching goals, to dependent ones, e.g., when performing a dual-arm manipulation.

Our proposed approach differs from the approaches mentioned previously in two ways. First, these approaches fall under the LfD category while our proposed SIMPLe framework is an IIL algorithm, and to the best of our knowledge, SIMPLe is the first framework for learning of bimanual tasks from interactive corrections. Second, our interactive framework avoids heuristics for coordinating policies for each arm in a bimanual setup by using human feedback to regulate each arm's dynamics

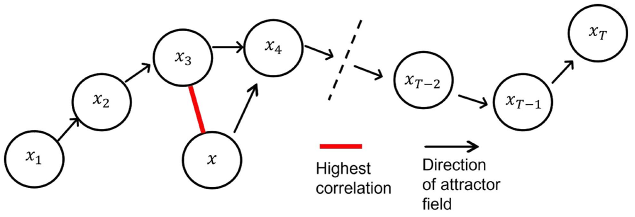

Fig. 2. Representation of a trajectory as a chain of events. The state $\boldsymbol{x}$ is the aggregation of the robot pose and time, where $\boldsymbol{x}_i$ is ith element in the reference trajectory. Every element of the trajectory ith has as element goal the next point on it $(i+1)$th depicted by a forward arrow. The $\boldsymbol{x}$ gets a unitary correlation with the closest element in the trajectory mth and then as the goal the location of $(m+1)$th state on the chain of events. The uncertainty is given by the distance from the $\boldsymbol{x}$ and its correlated point on the trajectory.

before transferring it to a bimanual policy. Then, when the bimanual policy is executed, the robot's reaction to disturbances depends on the mechanical coupling of the end-effectors (see Section IV-C), or on chosen input state for the policy (see Section III).

C. Motion Stability

The stability of the bimanual operation is another key aspect. When learning from a small amount of data, in particular, the stability of the learned behavior can be jeopardized when demonstrations are imperfect. In [25], [26], an LFD approach is combined with a learned controller that adapts the motion to keep the learned trajectory stable when facing external forces. In [19], the motion is divided into one DS for each subgoal with a hand-designed vector field that brings the robot always close to the connecting lines of subgoals. Our proposed MPs have the objective of learning long-horizon MP with only one final goal and to obtain the stability property as an emerging behavior of the motion encoding (see Section III-C).

Next, Section III introduces the novel GP-based formulation used for modeling MPs, Section IV introduces the proposed SIMPLe algorithm and how we use it for performing interactively learning bimanual MPs, Section V shows different applications and user-cases. Finally, Section VI concludes this article with final remarks and future works.

III. Movement Representation

Section III-A presents the proposed graph Gaussian process (GGP) formulation, Section III-B the proposed trajectory learning framework and its benefits for safety, Section III-C presents the stability achieved with the proposed framework, and Sections III-D and III-E compare learning trajectories using traditional GPs and the proposed GGP.

A. Movement Learning With GP

To learn the model of the demonstrated trajectories, we chose GPs because it is a flexible nonparametric regression method where the kernel choice can be used to increase the inductive bias on the generalization of unseen points, which is prohibitive using function approximators, such as DMPs or NNs. Furthermore, its solid statistical formulation provides both the mean and the epistemic uncertainty of the prediction [13] that can be used for disturbance rejection or stiffness regulation [27].

Given the training data composed by a set of states $\boldsymbol{\mathcal{X}}$ and their respective labels $\boldsymbol{\mathcal{Y}}$, the prediction mean and variance at the evaluation point $\boldsymbol{x}$ follow, respectively:

$$\mu(\boldsymbol{x}) = \boldsymbol{k}_{\star}^{\top}\boldsymbol{\mathcal{K}}^{-1}\boldsymbol{\mathcal{Y}} \tag{1}$$

$$\sigma(\boldsymbol{x}) = \boldsymbol{k} - \boldsymbol{k}_{\star}^{\top}\boldsymbol{\mathcal{K}}^{-1}\boldsymbol{k}_{*} \tag{2}$$

where, the $\boldsymbol{k} = \boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x})$ is the variance of a single evaluation point $\boldsymbol{x} \notin \boldsymbol{\mathcal{X}}$, $\boldsymbol{k}_{\star} = \boldsymbol{k}(\boldsymbol{\mathcal{X}}, \boldsymbol{x})$ is the variance between $\boldsymbol{x}$ and the training inputs $\boldsymbol{\mathcal{X}}$, and $\boldsymbol{\mathcal{K}} = \boldsymbol{\mathcal{K}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}})$ is the covariance matrix of the training data representing the leaned model [13]. Note that $\boldsymbol{k}$, $\boldsymbol{k}_{\star}$, and $\boldsymbol{\mathcal{K}}$ are based on the kernel functions and their hyperparameters, which are used for incorporating prior knowledge into the process.

In particular, the kernel determines the interpolation and extrapolation behaviors and when using a distance-based kernel, e.g., radial basis function, the prediction converges to the mean of the GP, usually set to zero. Our objective is to have a mean function that extrapolates, i.e., does not return a vanishing prediction, but without losing the measure of epistemic uncertainty. For this reason, by correlating with only the closest neighbor, in the dataset and changing the kernel definition to

$$\tilde{\boldsymbol{k}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{cases} 1, & \text{if } \boldsymbol{k}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \max(\boldsymbol{k}(\boldsymbol{x}_i, \boldsymbol{\mathcal{X}})) \\ 0, & \text{otherwise} \end{cases} \forall \boldsymbol{x}_j \in \boldsymbol{\mathcal{X}}. \tag{3}$$

In simple terms, given a point $\boldsymbol{x}_i$, the correlation is 1 only if that is the maximum obtainable correlation when correlating $\boldsymbol{x}_i$ with all $\boldsymbol{x}_j \in \boldsymbol{\mathcal{X}}$. With the new kernel, the prior covariance matrix becomes

$$\Sigma_{\text{prior}} = \begin{bmatrix} \tilde{\boldsymbol{\mathcal{K}}} & \boldsymbol{k}_{\star} \\ \tilde{\boldsymbol{k}}_{\star}^{\top} & \boldsymbol{k} \end{bmatrix}. \tag{4}$$

Note that, since the last column is for $\boldsymbol{x}_j \notin \boldsymbol{\mathcal{X}}$, the saturation is not applied. Thus, the resulting prior covariance matrix is not symmetric anymore, making the new process a pseudo-GP. After the conditioning on the data points, the new pseudo-GP posterior becomes

$$\mu(\boldsymbol{x}) = \tilde{\boldsymbol{k}}_{\star}^{\top}\tilde{\boldsymbol{\mathcal{K}}}^{-1}\boldsymbol{\mathcal{Y}} = \tilde{\boldsymbol{k}}_{\star}^{\top}\boldsymbol{\mathcal{Y}} \tag{5}$$

$$\sigma(\boldsymbol{x}) = \boldsymbol{k} - \tilde{\boldsymbol{k}}_{\star}^{\top}\boldsymbol{k}_{\star}. \tag{6}$$

In simple terms, $\tilde{\boldsymbol{k}}_{\star}$ selects as mean the label of the closest point in the database, computing the uncertainty according to the relative position between the query and the selected points.

In addition, by saturating the covariance matrix $\boldsymbol{\mathcal{K}}$, each trajectory element has its highest correlation with themselves: the new saturated correlation matrix, $\tilde{\boldsymbol{\mathcal{K}}}$, is the identity matrix, thus eliminating the computationally heavy $\mathcal{O}(n^3)$ matrix inversion. However, with this approximation, we are losing interpolation/smoothing properties. Meaning that the provided trajectory data must be without drastic jumps. In practice, recording trajectories with a high enough frequency ($>$ 10Hz) and/or smoothing the data makes the use of the proposed approximation doable. It is worth mentioning that the presented formulation is tailored for the specific application of movement learning and does not necessarily substitute general approximation methods, such as local models [28] or variational approximations [29]. A detailed comparison between GPs and GGPs for trajectory learning is presented in Section III-D.

B. Representing Trajectories as Graphs

Our goal is to perform safe control during the general or corrective interactions between robots and humans. To that goal, we start from a recorded trajectory demonstration, defined as an array of n end-effector poses $\boldsymbol{\xi} = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{n-1}\} \in \mathbb{R}^3$ and the timestamp of each respective pose $\boldsymbol{\tau} = \{t_0, \ldots, t_{n-1}\} \in \mathbb{R}$, and a final pose and time $\boldsymbol{x}_n, t_n$, used to fit a policy π. The trajectory can be seen as a sequence of events, represented as a graph with edges representing transitions from the state at time t_i to the state at t_{i+1}. Given the adopted GP approximation, during the policy execution, the most correlated point is selected on the trajectory, and its label is selected as the goal, see Fig. 2. We denote the policy as a GGP.

However, the input type of the policy can completely change the robot's behavior. For example, a pose-only "feedback" policy, $\pi_{\boldsymbol{x}} : \boldsymbol{x} \rightarrow \boldsymbol{x}_g$ is a fully reactive policy which computes the next Cartesian pose for the end-effector $(\boldsymbol{x}_g)$, based on the current one $(\boldsymbol{x})$. Such policies are safer since they make the robot to wait when its path is obstructed and allow it to rejoin the trajectory on its closest point under perturbations [27]. However, they cannot deal with movement ambiguities and time-dependent movements.

Alternatively, a time-only dependent policy, $\pi_t : t \rightarrow \boldsymbol{x}_g$, computes $(\boldsymbol{x}_g)$ based on the current time (t). This type of policy can deal with movement ambiguities, e.g., when the demonstrated trajectory crosses itself, and with time-dependent movements, i.e., when the movement has to be temporarily paused at a specific position. However, such "feed-forward" policies are not a safe choice since the attractor moves on the trajectory without considering dangerous interactions with the human and with the environment.

Instead, we proposed the usage of pose and time-*belief* dependent policies, $\pi_{\boldsymbol{x},t^b} : \boldsymbol{x}, t^b \rightarrow \boldsymbol{x}_g, t^b_g$, which computes the pose goal and a new time belief $(\boldsymbol{x}_g, t^b_g)$ based on the current ones $(\boldsymbol{x}, t^b)$. Note that the time-*belief* is updated with the time of the selected goal in the trajectory. Encoding both pose and time belief allows for obtaining safe policies capable of handling time-dependent movements and ambiguities.

As such, SIMPLe can be used with models fitted as time-dependent, pose-dependent, or pose and time-dependent policies by setting the GGP states as $\boldsymbol{x} = t$, $\boldsymbol{x} = \boldsymbol{x}$, or $\boldsymbol{x} = [\boldsymbol{x}, t^b]^\top$, respectively, and selecting a kernel for fitting the trajectories w.r.t. time $(\boldsymbol{\mathscr{k}}(t, \boldsymbol{\tau}))$, such as in [30], position $(\boldsymbol{\mathscr{k}}(\boldsymbol{x}, \boldsymbol{\xi}))$, such as in [27], or both of them, as proposed in SIMPLe, which is obtained by multiplying the time and the pose-dependent kernels, i.e., $\boldsymbol{\mathscr{k}}([\boldsymbol{x}, t^b], [\boldsymbol{\xi}, \boldsymbol{\tau}]) = \boldsymbol{\mathscr{k}}(\boldsymbol{x}, \boldsymbol{\xi}) \circ \boldsymbol{\mathscr{k}}(t, \boldsymbol{\tau})$.

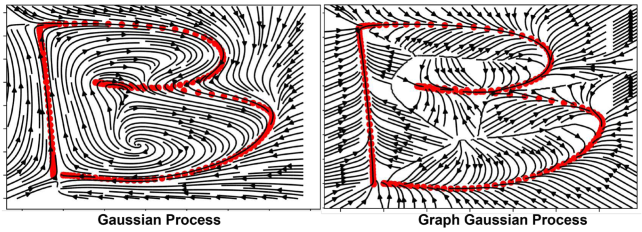

Fig. 3. Comparison of the fitting of a trajectory with the shape of a "B" with *position dependent* Gaussian Process (GP) and Graph Gaussian Process (GGP). The red dots are the recorded demonstrations, and the stream curves are the learned behavior.

In the context of trajectory learning, the labels are set as the aggregation states in the demonstration, which follow each state in the demonstration, i.e., $\boldsymbol{\mathscr{Y}} = [\boldsymbol{\xi}^d, \boldsymbol{\tau}^d]^\top = [\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}, \{t_1, \ldots, t_n\}]^\top$.

C. Stability Analysis

From this GGP-based formulation, we can also conclude the following.

Proposition 1: Using the trajectory graph representation, the motion always converges on the proximity of the demonstration and continues toward the end of it.

Proof: Since the vector $\tilde{\boldsymbol{\mathscr{k}}}_\star^\top$ is correlating the current position of the end-effector with only one node of the trajectory, and if there is no overlap on the trajectory, the robot will move toward the goal of the closest node. Then, node by node, it continues toward the end of the trajectory. ■

A great advantage of the pose and time trajectory encoding is that overlapping is no longer possible as the demonstrator cannot show two different robot positions simultaneously, leading to the absence of overlapping nodes, ambiguities, or undesired loops, guaranteeing that the hypothesis in the proof of convergence is satisfied. However, this also means that when *only* computing the correlation as a function of position, no physical overlapping of the trajectory can be demonstrated, such as when drawing an eight [31].

D. Comparison Between GPs and GGPs for Policy Learning

Fig. 3 shows the different behavior in learning to draw the letter "B" (database from [30]) using a GP and a GGP using only the 2-D position. The first thing to highlight is the effect of the kernel saturation in a faster convergence closer to the trajectory of the GGP compared with the GP. As a consequence, when the robot is perturbed, the motion tends to go closer to the trajectory and continue from there. Nevertheless, this difference in the vector fields does not lead to unsafe sudden motions straight toward the attractor due to the proposed attractor and stiffness regularization/saturation described in Section IV-C.

The letter "B" shows a clear ambiguity at the overlapping of the trajectory between the two humps. The robot must first move in and then move out of the intersection on the same line in order to continue toward the end of the trajectory. The

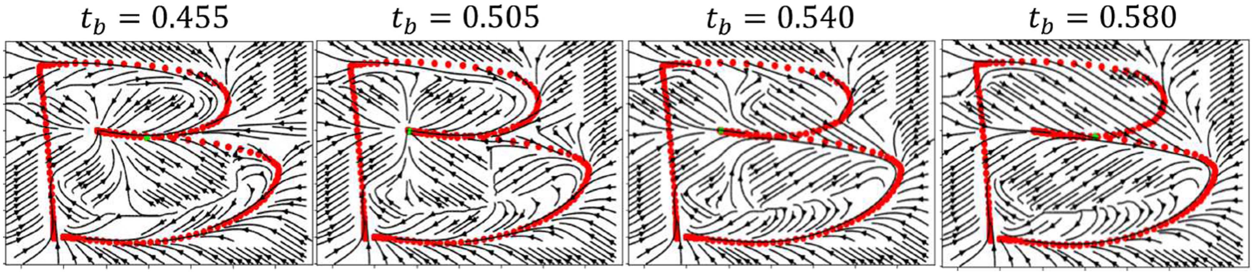

Fig. 4. Attractor vector-field of a GGP when the conditioning of the kernel with different time belief. t_b is the time belief normalized by the total time of the trajectory, i.e., $0 \leq t_b \leq 1$. The corresponding element of the chain for every case is highlighted with a green circle. The red circles are the collection of points of the trajectory. In this example, the time belief is the time of the next element of the chain.

learned behavior of the two fitting methods is different. The GP removes the overlapping ambiguity by considering it as noise. This results in cutting the motion without going down to the intersection of the curves, losing tracking accuracy. On the other hand, in the line overlapping, the GGP has a vector field pointing left when approaching from below and to the right when approaching from the top. This may lead to an ambiguous situation that can cause the robot to get stuck locally or, in general, not track the motion correctly. This motivates the use of a position and time-dependent policy, to remove any possible state overlapping.

E. Movement Disambiguation Using Pose and Time-Dependent Policy

As explained in Proposition 1, no loops in the chain are allowed to guarantee good trajectory tracking. Thus, our solution is to consider also the time belief (t^b) in the state. Fig. 4 shows the evolution of the vector field for different time beliefs. The chain element of the trajectory for the t^b indicated above the figure is highlighted with a green dot. From the figure, it is possible to observe how the previously encountered ambiguity is elegantly solved. In fact, the robot gets into the valley and then out without getting stuck.

In order to simulate the behaviors of a GGP with or without a self-update of the time belief, 200 different trajectories are rolled out starting from the origin of the demonstration. In order to take into account the inaccuracy of the low-level (impedance) controller, a Gaussian noise of magnitude 0.01 is added to the attractor when computing the new position. Fig. 5 depicts the mean and standard deviation of the trajectories. When the time dependence is active (left-hand side of the figure), the trajectory always converges to the end, and the fluctuations are bounded. When only the position is considered (right-hand side of the figure), the variability of the sampled trajectories increases, and the tracking is good on average until the start of the two humps intersection, from where the performance degrades due to the ambiguous states.

IV. SIMPLe: Safe Interactive Movement Primitives Learning

The proposed SIMPLe framework summarized in Algorithm 1 consists of three main parts. First, the human teacher

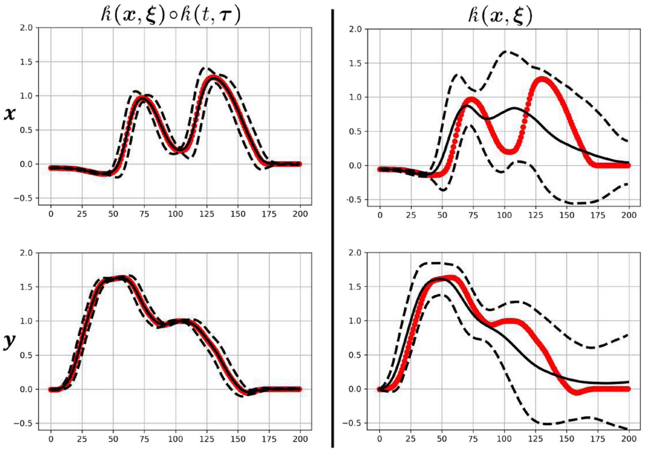

Fig. 5. 200 roll-outs of 200 steps. The attractor position is injected with a Gaussian noise of zero mean and std of 0.01. When the kernel is also taking into account the time belief, the motion is more robust when encountering ambiguity in the intersections of the two curves. Otherwise, ambiguity can lead to divergent behaviors. Red: original demonstration. Black: the average and standard deviation of the execution.

provides kinesthetic demonstrations (see Section IV-A), from which a time and position-dependent model (see Section III) is learned. Second, the proposed method enables the human to provide demonstrations and to make interactive corrections (see Section IV-A), which are leveraged for learning the trajectories and synchronization of bimanual tasks (see Section IV-B). And third, the bimanual task can be executed. We employ a Cartesian impedance control to facilitate physical interactions during demonstrations, corrections and autonomous execution (see Section IV-C), safety is ensured thanks to the proposed stiffness regulation (see Section IV-D) and coupling between manipulators (see Section IV-E). Our method aims to enhance the teaching ability of nonexpert users while guaranteeing a safe interaction while teaching, correcting, and executing bimanual tasks. To cope with the complexity of teaching bimanual tasks, SIMPLe provides an interactive kinesthetic teaching (KT) approach allowing to teach one arm at a time and then to teach how to synchronize them using touch by leveraging the time and pose-dependent GGP formulation presented in Section III. To the best of our knowledge, SIMPLe is the first framework to employ IIL on bimanual setups. Nevertheless,

Algorithm 1: SIMPLe.

```
   /* a) Passive Kinesthetic Teaching */
1  while Trajectory Recording do
2      $[\boldsymbol{x}, t]^{\top} = \text{Receive}(\boldsymbol{x}, t)$
3      $\mathfrak{X} = \mathfrak{X} \uplus [\boldsymbol{x}, t]^{\top}$
4  end
5  $\boldsymbol{\mathcal{Y}}_i = \mathfrak{X}_{i+1}, \forall i \in \{1, 2, \ldots, n-1\}$, with $n = \dim(\mathfrak{X})$
   /* b) Active Kinesthetic Teaching */
6  $t^b = 0$
7  while Control Active do
8      $[\boldsymbol{x}, t]^{\top} = \text{Receive}(\boldsymbol{x}, t)$
9      $[\boldsymbol{\mu}, \boldsymbol{\sigma}] = \pi_{\boldsymbol{x}, t^b}(\boldsymbol{x}, t^b)$
10     $[\boldsymbol{x}_g, t^b_g]^{\top} = \boldsymbol{\mu}$
11     $t^b = t^b_g$
12     $\text{Send}(\text{saturate}(\Delta \boldsymbol{x}), \hat{\boldsymbol{K}})$    // see Sec. IV-C
13     $\mathfrak{X} = \mathfrak{X} \uplus [\boldsymbol{x}, t]^{\top}$
14 end
15 $\boldsymbol{\mathcal{Y}}_i = \mathfrak{X}_{i+1}, \forall i \in \{1, 2, \ldots, n-1\}$, with $n = \dim(\mathfrak{X})$
   /* c) Execution */
16 $t^b = 0$
17 while Control Active do
18     $\text{Receive}(\boldsymbol{x})$
19     $[\boldsymbol{\mu}, \boldsymbol{\sigma}] = \pi_{\boldsymbol{x}, t^b}(\boldsymbol{x}, t^b)$
20     $[\boldsymbol{x}_g, t^b_g]^{\top} = \boldsymbol{\mu}$
21     $t^b = t^b_g$
22     $\text{Send}(\text{saturate}(\Delta \boldsymbol{x}), \hat{\boldsymbol{K}})$    // see Sec. IV-C
23 end
```

SIMPLe does not restrict users from teaching (and correcting) both arms simultaneously, and it can be applied for single-arm manipulation tasks without any loss of generality.

A. Teaching From Kinesthetic Demonstrations and Corrections

LfD allows nonexpert users to program robots to perform complex tasks without any programming knowledge. Different interfaces can be used to transfer data to the robot, such as teleoperation devices, touch screens or physical interaction with the robot's embodiment, obtaining a KT approach. When the user is teaching a task, the stiffness and damping of the Cartesian impedance controller are set to zero, allowing the user to easily move the robot. The positions $\boldsymbol{\xi}$ and times $\boldsymbol{\tau}$ of the demonstrated trajectories are recorded, and their respective goals, $\boldsymbol{\xi}^d$ and $\boldsymbol{\tau}^d$ are obtained by shifting $\boldsymbol{\xi}$ and $\boldsymbol{\tau}$ forward in time (Algorithm 1, lines 1 to 3).

After learning the motion from a kinesthetic demonstration, the user can reshape the trajectory of each arm to achieve, for example, coordination between the arms in the execution of the task. Given the Cartesian impedance controller (see Section IV-C) kinesthetic corrections can be performed by simply applying an external force. Such a controller allows for the human to be in full control if the stiffness is set to zero, or the robot can gradually increase its control by regulating the stiffness.

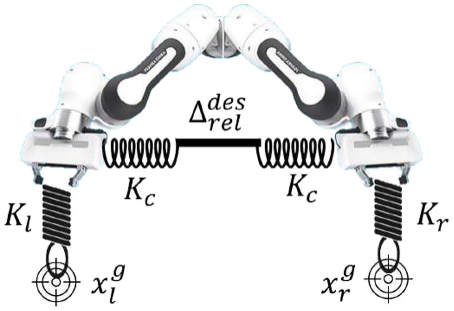

Fig. 6. Bimanual impedance controller. For simplicity the spring-damper system is represented only with a spring.

In addition, given the time and pose-dependent policy (see Section III), the demonstrator can also drag the robot forward or backward in time along its trajectory. This property can be used, for example, to make the execution of the initial demonstration faster [32], [33], to make the robot throw objects [19], or for synchronization learning, as proposed in this article.

B. Interactive Learning of Bimanual Tasks

When teaching bimanual tasks, it is not always easy or feasible to provide kinesthetic demonstrations with both arms simultaneously, especially when using large redundant manipulators. In addition, even when skilled users are able to teach a bimanual task by moving each end-effector with a single hand, they may perform a suboptimal trajectory, or an ineffective one, given the task complexity.

In SIMPLe, the movement of each arm can be executed independently according to the GGP formulation described in Section III.

The proposed interactive learning method offers many possibilities for nonexpert users to teach complex bimanual tasks. For example, they can demonstrate the movement for picking up a box one arm at a time and then learn to coordinate the two independent trajectories and apply enough pressure on the sides of the box to execute the task successfully. Moreover, learning repetitive tasks, such as object handover, can also be initially demonstrated one arm at a time and later use kinesthetic corrections to learn how to coordinate both arms. Thanks to the calculation of the model as a function of position and time (belief), the user can also bring the robot back to the start of the trajectory and teach (with minimum interaction effort) to perform the task multiple times.

C. Safe Cartesian Impedance Control

Fig. 6 illustrates the employed Cartesian impedance controller in each of the two manipulators, which emulates the behavior of a mass-spring-damper system

$$\boldsymbol{\Lambda}(\boldsymbol{q})\ddot{\boldsymbol{x}} = \boldsymbol{K}\Delta\boldsymbol{x} - \boldsymbol{D}\dot{\boldsymbol{x}} + \boldsymbol{f}_{ext} \tag{7}$$

where, $\boldsymbol{\Lambda}(\boldsymbol{q})$ is the Cartesian inertia matrix of the physical system, $\boldsymbol{K}$ and $\boldsymbol{D}$ are stiffness and damping that symmetric, positive-definite matrices; $\boldsymbol{f}_{ext}$ is the total external force, and

$\Delta \boldsymbol{x} = \boldsymbol{x}_g - \boldsymbol{x}$ is the distance between the goal and end-effector poses. The damping matrix can be designed to simulate a critical damping system [34]. In this framework, after computing the orthogonal decomposition of $\boldsymbol{K}$, i.e., $\boldsymbol{K} = \boldsymbol{R}\tilde{\boldsymbol{K}}\boldsymbol{R}^T$ [1], where $\tilde{\boldsymbol{K}}$ is a diagonal matrix, then $\tilde{\boldsymbol{D}} = 2\tilde{\boldsymbol{K}}^{1/2}$ and $\boldsymbol{D} = \boldsymbol{R}\tilde{\boldsymbol{D}}\boldsymbol{R}^T$. Please notice that $\tilde{\boldsymbol{K}}$ is a diagonal matrix; hence, the square root is applied to every element of the diagonal.

The Cartesian impedance controller takes as input a stiffness matrix and a displacement vector (lines 13 and 24 in Algorithm 1); in order to enhance safety when interacting with humans [35], it is necessary to saturate the attractor displacement and the stiffness to a maximum safe value. To help the definition of the bound, we can compute them as a function of the desired maximum free-movement velocity ($\boldsymbol{v}_{\max}$) and maximum applicable static force of the end-effector($\boldsymbol{F}_{\max}$) (in absolute values).

First, we compute an upper bound for the maximum displacement. Considering (7), when the robot is in free-movement, i.e., $\boldsymbol{f}_{ext} = 0$, the maximum velocity happens for $\ddot{x} = 0$, that is to say

$$\boldsymbol{D}|\dot{\boldsymbol{x}}| = \boldsymbol{K}|\Delta \boldsymbol{x}|. \tag{8}$$

Thus, given the current setted stiffness $\boldsymbol{K}$ and the desired max allowed velocity $\boldsymbol{v}_{\max}$, $\Delta \boldsymbol{x}$ needs to respect

$$|\Delta \boldsymbol{x}| \leq \Delta \boldsymbol{x}_{\max} = \boldsymbol{K}^{-1}\boldsymbol{D}\boldsymbol{v}_{\max} = 2\boldsymbol{R}\tilde{\boldsymbol{K}}^{-\frac{1}{2}}\boldsymbol{R}^T\boldsymbol{v}_{\max} \tag{9}$$

obtained after using the definition of damping. Before sending to the robot, the $\Delta \boldsymbol{x}$ is *saturated* in order to respect the upper bound.

However, if taking into account the maximum static force ($\boldsymbol{F}_{\max}$) when $\dot{\boldsymbol{x}} = 0$ and $\ddot{\boldsymbol{x}} = 0$, an upper bound on the stiffness can be found, such that

$$\boldsymbol{K}\Delta \boldsymbol{x}_{\max} \leq \boldsymbol{F}_{\max} \tag{10}$$

$$\boldsymbol{R}\tilde{\boldsymbol{K}}\boldsymbol{R}^T\Delta \boldsymbol{x}_{\max} \leq \boldsymbol{F}_{\max} \tag{11}$$

$$2\tilde{\boldsymbol{K}}\boldsymbol{R}^T\boldsymbol{R}\tilde{\boldsymbol{K}}^{-1/2}\boldsymbol{R}^T\boldsymbol{v}_{\max} \leq \boldsymbol{R}^T\boldsymbol{F}_{\max} \tag{12}$$

$$2\tilde{\boldsymbol{K}}^{\frac{1}{2}}\boldsymbol{R}^T\boldsymbol{v}_{\max} \leq \boldsymbol{R}^T\boldsymbol{F}_{\max}. \tag{13}$$

Hence, since the matrix $\tilde{\boldsymbol{K}}$ is diagonal, we can find the upper bound of each element in the ith row and column ($\tilde{\boldsymbol{K}}_{ii}$) as

$$\tilde{\boldsymbol{K}}_{ii} \leq \left(\frac{(\boldsymbol{R}^T\boldsymbol{F}_{\max})_i}{2(\boldsymbol{R}^T\boldsymbol{v}_{\max})_i}\right)^2. \tag{14}$$

So, in every singular component, the value of the principal stiffness is *saturated* in order to respect the found inequality.

D. Stiffness Regulation

Regulating the stiffness can be used to incrementally increase the stiffness after each demonstration, reducing human control as the learned movement is interactively refined [36]. Alternatively, the stiffness can be regulated when perceiving strong external forces, as a disagreement detection [37]. Similarly, [32] proposed a variation of a DMP where the robot variable stiffness and the regressor phase are modulated to adapt to human kinesthetic demonstrations.

When more demonstrations are provided, the measure of *aleatoric* uncertainty, i.e., variability in the demonstration, can be used to regulate the tracking stiffness of the robot [38]. Differently, we propose to exploit the *epistemic* uncertainty quantification of the policy ($\boldsymbol{\sigma}$), enabling for automatically regulating the Cartesian impedance controller's stiffness, hence switching control between robot and human.

Mathematically

$$\hat{\boldsymbol{K}} = \text{saturate}(\boldsymbol{K})\frac{1-\boldsymbol{\sigma}(\boldsymbol{x})}{1-\boldsymbol{\sigma}_{tr}}, \text{ when } \boldsymbol{\sigma}(\boldsymbol{x}) > \boldsymbol{\sigma}_{tr} \tag{15}$$

where, the $\boldsymbol{\sigma}_{tr}$ is the uncertainty threshold that is used to detect the disagreement. Note that $\boldsymbol{\sigma}(\boldsymbol{x})$ goes from 0 when close to the trajectory, to 1 when at infinite distance from it. Thanks to this stiffness regulation, when the robot is dragged in regions of high uncertainties, it mitigates the external force applied to the user perturbing the trajectory. This behavior can be conceptualized as the robot's nonverbal teaching request or repositioning into regions closer to the demonstration.

E. Dual Cartesian Impedance Control

Differently from the execution of a single-arm, when a two-arms policy execution is performed, extra attention is required regarding the mechanical coupling of the movement. For example, when picking up a box with two hands and executing a reshelving operation, in case of a perturbation of one arm, the other arm must also follow the perturbed movement. In this case, both arms must be *mechanically* coupled, meaning that in the impedance control of each arm, we would add an extra coupling force defined as

$$\boldsymbol{F}_c^l = \boldsymbol{K}_c\left(\boldsymbol{x}_r - \boldsymbol{x}_l - \Delta\boldsymbol{x}_{\text{rel}}^{\text{des}}\right) + \boldsymbol{D}_c\left(\dot{\boldsymbol{x}}_r - \dot{\boldsymbol{x}}_l\right) \tag{16}$$

$$\boldsymbol{F}_c^r = \boldsymbol{K}_c\left(\boldsymbol{x}_l - \boldsymbol{x}_r + \Delta\boldsymbol{x}_{\text{rel}}^{\text{des}}\right) + \boldsymbol{D}_c\left(\dot{\boldsymbol{x}}_l - \dot{\boldsymbol{x}}_r\right) \tag{17}$$

where, $\Delta\boldsymbol{x}_{\text{rel}}^{\text{des}} = \boldsymbol{x}_r^{\text{des}} - \boldsymbol{x}_l^{\text{des}}$ is the desired distance from the two end-effectors controlled by the SIMPLe algorithm. A simple schematic visualization of the proposed bimanual impedance control is displayed in Fig. 6 where each end-effector is coupled with a stiffness (and damper) with respect to their goal but also with a relative stiffness (and damper) between them.

Note that the proposed safety saturation and regulation process described, respectively, in Section IV-C and (15) are applied on a per-arm basis, thus being applied to single-arm setups. For a bimanual setup, the displacement and stiffness for the coupling forces [$\boldsymbol{F}_c$, defined in (16) and (17)] are saturated and regulated similarly to (9), (14), and (15).

V. Real Robot Validation

We performed the experiments with two 7-DoF Franka–Emika Panda placed vertically on a table and with the same

[1] $\boldsymbol{R}$ is an orthogonal matrix, hence $\boldsymbol{R}^T = \boldsymbol{R}^{-1}$

orientation. The impedance control was implemented,[2] as described in Section IV. Each manipulator had a shared memory of their Cartesian poses, allowing the calculation of the mechanical coupling force. The experiments presented in Sections V-D, V-A, and V-B were performed using a custom 3-D-printed plate end-effector depicted in Fig. 10, which features a layer of soft form for reducing the interaction forces during impacts with objects as in [39]; the experiment presented in Section V-C was performed using the Franka gripper. The impedance control framework, written in C++ makes use of robot operating system to interface with Algorithm 1, written in Python.

We perform five experiments with the real robot setup as follows.

1) The interactive synchronization of the picking motion of a bottle crate when the demonstration is provided separately for each robot, showing how SIMPLe is used to learn a bimanual synchronization.
2) The interactive correction in picking a different crate compared to the one of the original demonstration, showing how to use the GGP formulation to modify the motion locally.
3) A handover task, where one robot picks and places an object and the other robot picks it from the other's goal location and places it at another position, showing the ability to restart the execution of a trajectory simply dragging the robot at the starting location.
4) A supplementary user study to compare teleoperation and KT, the two most common types of demonstration approaches.
5) A supplementary user study to compare giving interactive corrections to giving new demonstrations.

The first three are technical experiments to highlight and validate different functionalities of the proposed method. Each experiment was conducted in five trials, and for each of them, the final learned motion was performed five times after demonstration and correction(s). This approach allowed for the assessment of the reliability of the learned skill. The last two are supplementary user studies to evaluate the type of data input for the proposed by comparing two human demonstration approaches and to evaluate giving corrections compared to giving new demonstrations. These additional insights can provide a better understanding of the input data generation method and adjustments of the robot's skill for bimanual cases. For all the experiments, we used a position-time kernel for the GGP that computes the correlations and updates the time beliefs online. We use a negative exponential kernel, i.e., $k = \exp(-\frac{|x_i - x_j|}{\lambda})$, with a length scale of $0.05\,\mathrm{m}$ for the space correlation and $0.05\,\mathrm{s}$ for the time correlation. The sigma threshold is set to $\sigma(\lambda)$ which is the uncertainty when the closest point is at distance λ. The Cartesian stiffness is kept to $600\,\mathrm{N/m}$ for linear stiffness and $30\,\mathrm{Nm/rad}$ for rotational. The attractor distance is saturated at $0.05\,\mathrm{m}$, implying that the expected maximum applicable force is $30\,N$ in every linear Cartesian direction and the maximum expected linear velocity is $\approx 0.6\,\mathrm{m/s}$ in every linear direction. The rotation delta is saturated at $0.15\,\mathrm{rad}$, implying a maximum torque of $4.5\,\mathrm{Nm}$ in every rotational component and a maximum velocity of $\approx 0.4\,\mathrm{rad/s}$. The coupling stiffness is set to $800\,\mathrm{N/m}$ in the linear components and 0 for the rotational ones. The relative error is also saturated at $0.05\,\mathrm{m}$. A video of the experiments can be found at.[3]

A. Asynchronous Crate Picking

When a pianist approaches studying a new piece, they do it *one hand at a time*. After mastering the movement with each hand, they start learning how to successfully coordinate the combined execution. Inspired by this idea, in this validation experiment, the user is asked to demonstrate how to best pick a crate, first with the right and then with the left manipulator. However, when the independently learned behaviors were executed with SIMPLe the coordination was off, and the handling of the crate was not stable. In Section IV-B, we highlighted how user feedback can be used to reshape the trajectory and that the reactive formulation of SIMPLe makes the trajectory to "virtually" stop: this feature can be used to learn a bimanual task while simply coordinating the separately recorded policies, see Fig. 7.

The effect of the human input can be appreciated in Fig. 7. The original demonstrations are represented by dashed lines. Even if the movement of the two demonstrations looks correctly symmetric with respect to the y-plane, the right arm is slower. However, it can be noticed how, after only one correction round, the motion of the two demonstrations is synchronized, as depicted with a solid line. Given the perfect obtained synchronization, in the next round, the user focused on increasing the applied pressure on the side of the crate to increase the grasp reliability. In the five experiment repetitions, the user consistently provided necessary synchronization corrections. One trial had an additional correction round, and two trials had two extra correction rounds. After the interactive correction rounds, the robot always placed the crate correctly. The Cartesian error of the final crate position with respect to the final round of correction, considering 25 repetitions (five executions × five trials), has a mean of $0.021\,\mathrm{m}$ and a standard deviation of $0.009\,\mathrm{m}$.

B. Synchronous Crate Picking

In this experiment, we focused on successfully teaching the same task of picking a box but giving bimanual demonstrations and corrections. In particular, we showed that even giving only one bimanual demonstration with a few rounds of corrections, the task execution was successful. We also tested the possibility of locally modifying the original policy to pick a different box placed at a higher level. Fig. 8 highlights how the robot can be dragged higher sooner, at around 10 s, and how, after picking the crate, the robots follow the original policy, being able to place the crate and go to resting position autonomously. In the five experiment repetitions, in the first two trials, the user provided two rounds of correction, but only one in the last three. The final position error of the box has a mean of $0.005\,\mathrm{m}$ and a standard deviation of $0.004\,\mathrm{m}$.

[2] [Online]. Available: https://github.com/franzesegiovanni/franka_bimanual_controllers

[3] [Online]. Available: https://youtu.be/GasxgbJZHdQ

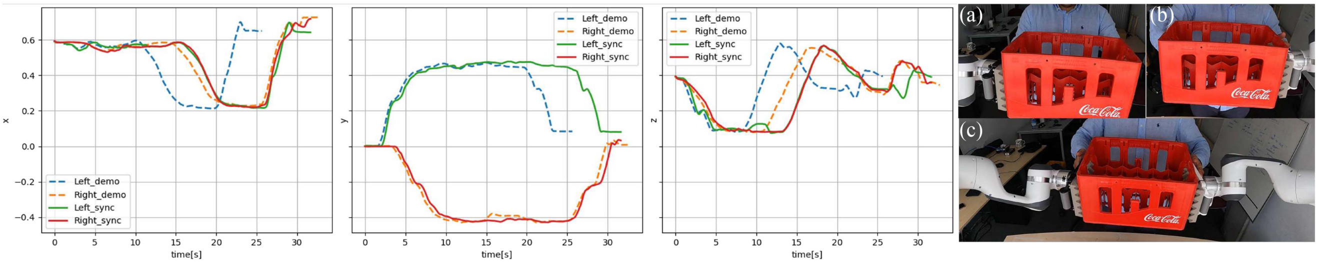

Fig. 7. Interactive synchronization of a bimanual picking task. The dashed lines are the demonstrations recorded in the independent demonstration phase (a) and (b). Since they are not perfectly synchronized, the autonomous execution would fail, hence, the human feedback in (c) allows a successful synchronization, depicted with solid lines.

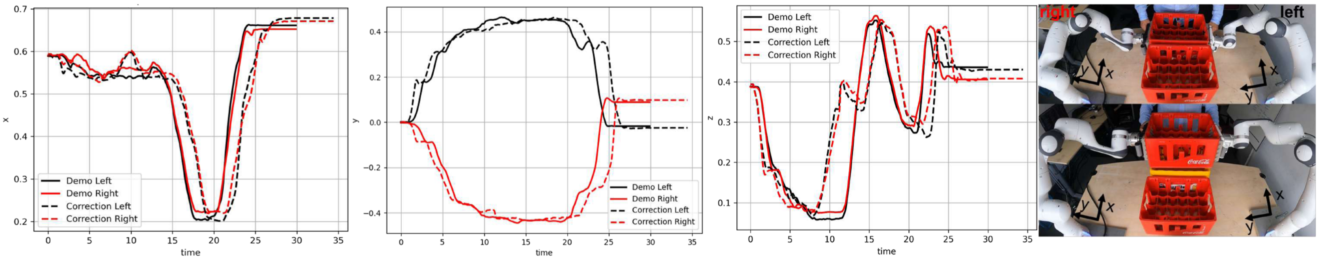

Fig. 8. Use interactive learning to teach the robot how to modify the original trajectory so the robot can learn how to pick a crate that is at a different height.

It is important to notice that even knowing the box's position, the motion's generalization in a task-parameterized approach is not trivial. In fact, the policy would have to move with respect to the picking frame and then, after a successful pick, switch with respect to the goal crate. This logic has shown to be successfully implemented in [33] but also to be a source of generalization ambiguities [40]. In general, performing a shared controlled teaching, with the user only taking control locally, can drastically reduce the burden of giving new complete demonstrations.

C. Object Handover

Another example of a tedious task is repetitive demonstrations: being able to demonstrate the task only once and then interactively assemble a long trajectory allows the teaching of complex bimanual coordination tasks, such as stirring a coffee mug [41] or learning a handover task. To validate SIMPLe in this circumstance, we taught the right arm to pick up a box and place it on the central separation line between the two robots. Then, the left arm would pick up the box and place it in its front. The goal is to show how dragging the robot around can be used for resynchronization or local trajectory reshaping and also as a movement "reset."

The original demonstrations are displayed with a dashed line in Fig. 9. When executing the motion with SIMPLe, the human can safely apply a force on the robot to stop its execution or drag it around to another desired position of the motion. At the beginning of Fig. 9, a force is applied to the left manipulator (highlighted by a red circle) to temporally stop it from moving, allowing the right arm to successfully pick a box and place it on the center line. At the moment that the user releases the robot, it is free to move and can pick up the box and reach its goal. To allow the repetition of the motion, the user applies a larger external force (observable with peaks), causing a drop in stiffness since the robot is probably dragged into a region of space with a lower correlation according to (15). Every time the robot finishes its pick and place task, if the user is willing to repeat it, they only have to drag the robot to the desired position of the trajectory. The user is teaching the motion multiple times, as reported with colored patches in the figure. We measure the final error in placing the box after the handover, executed five times in five different demonstration trials. The mean error and standard deviation are 0.011 and 0.008 m, respectively.

D. User Study: Teleoperation Versus Kinesthetic Teaching

The algorithm itself works with different data from different types of demonstrations. However, since obtained input data depends on the type of demonstration, the demonstration method is an essential part of the whole framework. Therefore, we conduct a supplementary user study to provide additional insight into the effects of the demonstration method to compare the two most common demonstration approaches: teleoperation and kinesthetic guidance. There are studies comparing both teaching approaches, but they were conducted for a single arm [42], [43]. The study in this article looks into this subject from a bimanual perspective.

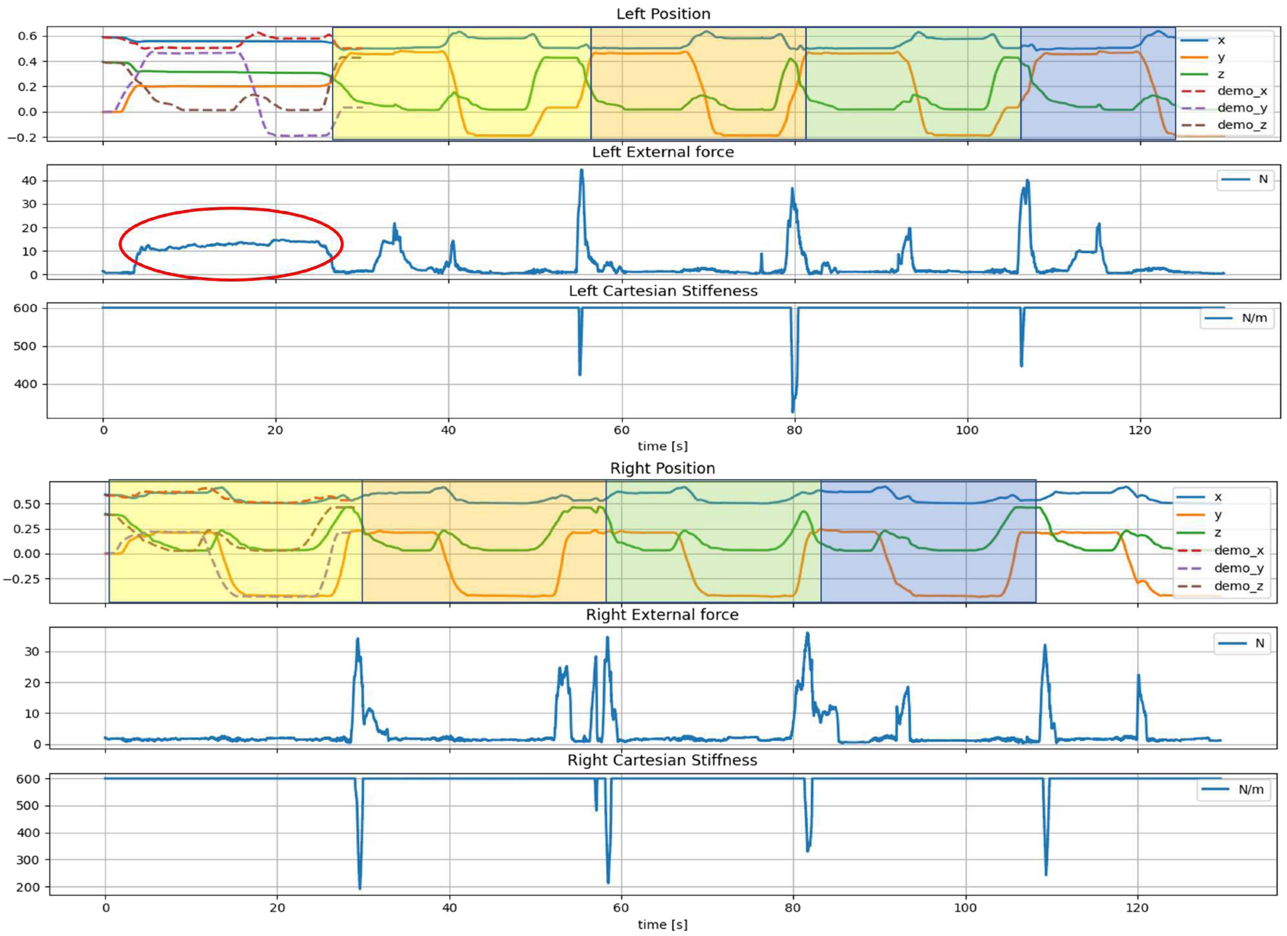

Fig. 9. Object handover. For both the left and the right arm, the Cartesian position (*x*,*y*,*z*) is depicted as a function of time, the Cartesian linear stiffness, and the norm of the Cartesian external force. The red circle indicates the region where an external force was applied to the left manipulator to correct the trajectories.

Section II highlighted how different works focus on enhancing the teleoperation ability of nonexpert users using assistive techniques, such as shared autonomy [9], [10], [11]. Since SIMPLe works with both teleoperated and kinesthetic demonstrations/corrections, we wanted to study which is more user-friendly. Although, getting the true answer is not easy: the teleoperation device can have a strong influence, as well as the dimension of the robot or the requested task. For the conducted user study, we asked seven nonexpert users to perform a relatively simple task: pick a box and stack it on top of another. These seven users were all male ranging from 23 to 40 years old.

In order to mitigate the learning bias from the results, participants had a familiarization phase for each teaching modality, in which they could restart the teaching session up to five times. For every new participant, the first teaching modality was alternated between teleoperated and kinesthetic, to remove the bias due to their familiarization with the task.

For metrics, we measured the success rate in solving the task and the total teaching time for each method. For subjective analysis, we asked the participants to complete a NASA TLX questionnaire. We conducted a paired samples t-test to verify if the time to do KT is significantly shorter than for teleoperation with the 6-D mice. However, three people out of seven failed to perform successful teleoperation, because they did not manage to coordinate well, making the robot self-collide or reach a joint limit. Therefore, we set as failure time the maximum time of the nonfailing ones. The test showed that KT requires less time compared to the teleoperation with the given hardware with the difference being statistically significant ($p < 0.05$).

Fig. 10 illustrates the average NASA TLX scores among the different users. We can observe that teleoperation resulted in being more mentally demanding and frustrating to perform. In general, we could observe that users tend to focus on teleoperating one arm at a time, making handling the box impossible. When providing KT, the physical contact with the robot helps them to understand the best trajectory better and to accomplish the task successfully.

E. User Study: Corrections Versus New Demonstration

Besides the input data generation method, another key factor related to bimanual manipulation teaching is how humans correct existing skills and what is their preference between correcting or giving a new demonstration. To test this, 12 nonexpert

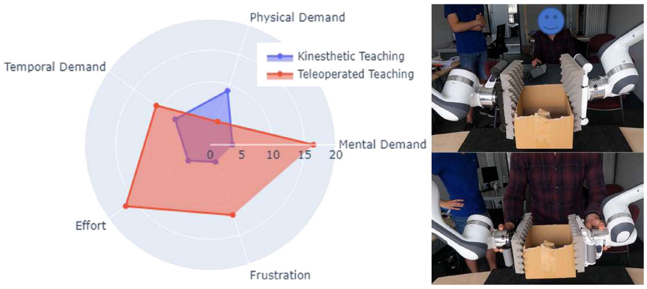

Fig. 10. User study to compare the performance of nonexpert users in performing bimanual teleoperation with two 6-D mice versus bimanual kinesthetic teaching (KT). On the left are scores of the NASA-TLX questionnaire, and on the right are the setups of teleoperating and performing KT.

TABLE I
LIKERT SCALE: CORRECTIONS VERSUS NEW DEMONSTRATIONS

Score	Q1	Q2	Q3	Q4	Q5
0 (strongly disagree)	0	0	0	4	0
1 (disagree)	0	2	2	4	2
2 (slightly disagree)	0	2	4	3	4
3 (slightly agree)	1	3	5	0	1
4 (agree)	2	4	0	1	5
5 (strongly agree)	9	1	1	0	0
mean	4,67	3.00	2,50	1,67	2,75
standard deviation	0,65	1,28	1,09	1,19	1,22

Q1: After giving Kinesthetic demonstration, I feel the robot is performing the task well,
Q2: After providing corrections, I feel the robot is adapting well to the novel situation,
Q3: I feel that giving a bimanual Kinesthetic Demonstration with two arms is tedious for a human teacher,
Q4: Performing Interactive Corrections is LESS physically tiring than giving a completely new demonstration,
Q5: Performing Interactive Corrections is LESS mentally tiring than giving a completely new demonstration

users participated in an experiment structured as follows. The user was asked to demonstrate the task of placing a box on the crate. The demonstration was then shown to the user after an offset was applied to the initial position of the box. The user was then tasked with kinesthetically correcting the initial policy to account for the change in the initial position. This was repeated two times for different initial positions of the box. The user now should have a sufficient understanding of what it means to give a demonstration or correction.

The second part of the experiment was designed to find the user preference for increasing lengths of the demonstration. The user was tasked with first demonstrating the task of placing the box on the crate. After the demonstration, an offset was applied to the box and the user was given the choice to either correct or redemonstrate given the new initial condition. For the second iteration, the task remained the same with the additional requirement that after picking up the box, before placing it on the crate, the user has to move the box through a different location as a waypoint. This was done to artificially lengthen the demonstration. Once again an offset was applied to the initial position of the box and the user was given the choice between correcting or redemonstrating. This was done one last time with two waypoints.

Given the choice, out of the 12 participants, 11, 8, and 10 chose to adjust the policy with the interactive corrections for the experiment with zero, one, and two waypoints, respectively, rather than providing new demonstrations. Thus, only in seven out of the 36 trials, a new demonstration was preferred, which indicates a strong preference for interactive corrections. Afterward, to evaluate their experience they were asked to answer several Likert scale questions related to user perception of corrected skill and their physical/mental load. The results can be seen in Table I, where the number in each cell represents the number of participants that choose a particular agreement on the Likert scale.

The users found that both new demonstrations and corrections were effective at improving the robot's task. The users were split on whether the bimanual demonstrations are tedious. In general, they found interactive corrections more physically demanding than providing new demonstrations, probably because the robots were already performing movements rather than being completely compliant during new demonstrations. During the experiments, it was observed that people that were shorter, had smaller hands, or were less muscular, tended to struggle more with correcting a policy. Those participants thus might have preferred giving a new demonstration over a correction. However, the users perceived interactive corrections as slightly less mentally demanding, probably because they needed to pay attention only to specific segments as opposed to the whole task.

VI. Conclusion

This article contributes to the field of bimanual manipulation with an interactive kinesthetic learning framework named SIMPLe. It uses a novel formulation of GP, named GGP, that is computationally efficient and ensures local and global stability of the motion while retaining an estimation of epistemic uncertainties. Thanks to the kernel formulation, the policy encoding can go from purely time-dependent to purely position-dependent or to a combination of both. At the same time, the graph representation of it allows an online update of the time *belief* that, differently from the robot position, cannot be directly measured. The study reports a comparison of a GP with the novel GGP, see Fig. 3 and an ablation study when the time dependence is considered or not, see Fig. 5. We conclude that considering the time and properly updating its beliefs allows dealing with more complex and possibly ambiguous demonstrations.

Various technical validation experiments were performed on a real bimanual setup to demonstrate the key functionalities and capabilities of the proposed method. The supplementary user studies gave interesting insights into how humans feel when teaching and correcting a robot with different modalities. Our study reported that users are faster and less stressed when performing KT compared to teleoperation. Furthermore, most users prefer giving corrections to completely new demonstrations.

References

[1] M. Garabini et al., "WRAPP-up: A dual-arm robot for intralogistics," *IEEE Robot. Automat. Mag.*, vol. 28, no. 3, pp. 50–66, Sep. 2021.
[2] C. Celemin et al., "Interactive imitation learning in robotics: A survey," *Found. Trends Robot.*, vol. 10, no. 1-2, pp. 1–197, 2022.
[3] T. Mukai et al., "Development of a nursing-care assistant robot RIBA that can lift a human in its arms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 5996–6001.

[4] J. Zhu, M. Gienger, G. Franzese, and J. Kober, "Do you need a hand?–An interactive robotic dressing assistance schemea," 2023, *arXiv:2301.02749*.

[5] P. Berthet-Rayne, M. Power, H. King, and G.-Z. Yang, "Hubot: A three state human-robot collaborative framework for bimanual surgical tasks based on learned models," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 715–722.

[6] X. Zhang, Y. Zheng, J. Ota, and Y. Huang, "Peg-in-hole assembly based on two-phase scheme and f/t sensor for dual-arm robot," *Sensors*, vol. 17, no. 9, 2017, Art. no. 2004.

[7] C. Smith et al., "Dual arm manipulation–A survey," *Robot. Auton. Syst.*, vol. 60, no. 10, pp. 1340–1353, 2012.

[8] Y. Yamada, S. Nagamatsu, and Y. Sato, "Development of multi-arm robots for automobile assembly," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1995, vol. 3, pp. 2224–2229.

[9] M. Laghi et al., "Shared-autonomy control for intuitive bimanual tele-manipulation," in *Proc. IEEE-RAS 18th Int. Conf. Humanoid Robots*, 2018, pp. 1–9.

[10] D. Rakita, B. Mutlu, M. Gleicher, and L. M. Hiatt, "Shared control-based bimanual robot manipulation," *Sci. Robot.*, vol. 4, no. 30, 2019, Art. no. eaaw0955.

[11] A. Tung et al., "Learning multi-arm manipulation through collaborative teleoperation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 9212–9219.

[12] R. Wen, Q. Rouxel, M. Mistry, Z. Li, and C. Tiseo, "Collaborative bimanual manipulation using optimal motion adaptation and interaction control: Retargetting human commands to feasible robot control references," *IEEE Robot. Automat. Mag.*, early access, May 16, 2023, doi: 10.1109/MRA.2023.3270222.

[13] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*, vol. 2. Cambridge, MA, USA: MIT Press, 2006.

[14] E. Gribovskaya and A. Billard, "Combining dynamical systems control and programming by demonstration for teaching discrete bimanual coordination tasks to a humanoid robot," in *Proc. IEEE/ACM 3rd Int. Conf. Hum.-Robot Interact.*, 2008, pp. 33–40.

[15] Y. Zhou, M. Do, and T. Asfour, "Coordinate change dynamic movement primitives–A leader-follower approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 5481–5488.

[16] Y. Zhou, M. Do, and T. Asfour, "Learning and force adaptation for interactive actions," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots (Humanoids)*, 2016, pp. 1129–1134.

[17] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," 2021, *arXiv:2102.03861*.

[18] Y. Fanger, J. Umlauft, and S. Hirche, "Gaussian processes for dynamic movement primitives with application in knowledge-based cooperation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 3913–3919.

[19] M. B. Bombile and A. Billard, "Dual-arm control for coordinated fast grabbing and tossing of an object: Proposing a new approach," *IEEE Robot. Automat. Mag.*, vol. 29, no. 3, pp. 127–138, Sep. 2022.

[20] O. Kroemer, C. Daniel, G. Neumann, H. van Hoof, and J. Peters, "Towards learning hierarchical skills for multi-phase manipulation tasks," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 1503–1510.

[21] S. Manschitz, J. Kober, M. Gienger, and J. Peters, "Probabilistic progress prediction and sequencing of concurrent movement primitives," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 449–455.

[22] R. Caccavale, M. Saveriano, G. A. Fontanelli, F. Ficuciello, D. Lee, and A. Finzi, "Imitation learning and attentional supervision of dual-arm structured tasks," in *Proc. IEEE Joint Int. Conf. Develop. Learn. Epigenetic Robot.*, 2017, pp. 66–71.

[23] F. Xie, A. Chowdhury, M. C. De Paolis Kaluza, L. Zhao, L. Wong, and R. Yu, "Deep Imitation Learning for Bimanual Robotic Manipulation," in *Advances in Neural Information Processing Systems,* vol. 33, H. Larochelle, M. Ranzato, R. M. Hadsell Balcan, and H. Lin, Eds. Curran Associates, Inc., 2020, pp. 2327–2337.

[24] S. S. M. Salehian, N. Figueroa, and A. Billard, "A unified framework for coordinated multi-arm motion planning," *Int. J. Robot. Res.*, vol. 37, no. 10, pp. 1205–1232, 2018. [Online]. Available: https://journals.sagepub.com/doi/full/10.1177/0278364918765952

[25] A. Gams, B. Nemec, A. J. Ijspeert, and A. Ude, "Coupling movement primitives: Interaction with the environment and bimanual tasks," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 816–830, Aug. 2014.

[26] N. Likar, B. Nemec, L. Žlajpah, S. Ando, and A. Ude, "Adaptation of bimanual assembly tasks using iterative learning framework," in *Proc. IEEE-RAS 15th Int. Conf. Humanoid Robots (Humanoids)*, 2015, pp. 771–776.

[27] G. Franzese, A. Mészáros, L. Peternel, and J. Kober, "ILoSA: Interactive learning of stiffness and attractors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 7778–7785.

[28] M. Schneider and W. Ertel, "Robot learning by demonstration with local Gaussian process regression," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 255–260.

[29] J. Hensman, A. Matthews, and Z. Ghahramani, "Scalable variational Gaussian process classification," *Artif. Intell. Statist.*, pp. 351–360, 2015.

[30] Y. Huang, L. Rozo, J. ao Silvério, and D. G. Caldwell, "Kernelized movement primitives," *Int. J. Robot. Res.*, vol. 38, no. 7, pp. 833–852, 2019.

[31] P. Valletta, R. Pérez-Dattari, and J. Kober, "Imitation learning with inconsistent demonstrations through uncertainty-based data manipulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 3655–3661.

[32] T. Kastritsi, F. Dimeas, and Z. Doulgeri, "Progressive automation with DMP synchronization and variable stiffness control," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3789–3796, Oct. 2018.

[33] A. Mészáros, G. Franzese, and J. Kober, "Learning to pick at non-zero-velocity from interactive demonstrations," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 6052–6059, Jul. 2022.

[34] T. Coleman, G. Franzese, and P. Borja, "Damping design for robot manipulators," in *Proc. Hum.-Friendly Robot. 2022: 15th Int. Workshop Human-Friendly Robot.*, 2023, pp. 74–89.

[35] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Requirements for safe robots: Measurements, analysis and new insights," *Int. J. Robot. Res.*, vol. 28, no. 11-12, pp. 1507–1527, 2009.

[36] M. Tykal, A. Montebelli, and V. Kyrki, "Incrementally assisted kinesthetic teaching for programming by demonstration," in *Proc. IEEE/ACM 11th Int. Conf. Hum.-Robot Interact.*, 2016, pp. 205–212.

[37] L. van der Spaa, G. Franzese, J. Kober, and M. Gienger, "Disagreement-aware variable impedance control for online learning of physical human-robot cooperation tasks," in *Proc. IEEE Int. Conf. Robotcs Automat.*, 2022, pp. 1–4. [Online]. Available: https://sites.google.com/view/saphri-icra2022/contributions

[38] N. Jaquier, D. Ginsbourger, and S. Calinon, "Learning from demonstration with model-based Gaussian process," in *Proc. Conf. Robot Learn.*, 2020, pp. 247–257.

[39] N. Dehio, J. Smith, D. L. Wigand, P. Mohammadi, M. Mistry, and J. J. Steil, "Enabling impedance-based physical human–multi-robot collaboration: Experiments with four torque-controlled manipulators," *Int. J. Robot. Res.*, vol. 41, no. 1, pp. 68–84, 2022.

[40] G. Franzese, C. Celemin, and J. Kober, "Learning interactively to resolve ambiguity in reference frame selection," in *Proc. Conf. Robot Learn.*, 2021, vol. 155, pp. 1298–1311.

[41] Z. Sun, Z. Wang, J. Liu, M. Li, and F. Chen, "Mixline: A hybrid reinforcement learning framework for long-horizon bimanual coffee stirring task," in *Proc. Int. Conf. Intell. Robot. Appl.*, 2022, pp. 627–636.

[42] K. Fischer et al., "A comparison of types of robot control for programming by demonstration," in *Proc. IEEE/ACM 11th Int. Conf. Hum.-Robot Interact.*, 2016, pp. 213–220.

[43] P. Gliesche, T. Krick, M. Pfingsthorn, S. Drolshagen, C. Kowalski, and A. Hein, "Kinesthetic device vs. keyboard/mouse: A comparison in home care telemanipulation," *Front. Robot. AI*, vol. 7, 2020, Art. no. 561015.

Giovanni Franzese (Graduate Student Member, IEEE) received the B.Sc. degree in mechanical engineering and the M.Sc. degree in mechatronics and robotics from Politecnico di Milano, Milan, Italy, in 2016 and 2018, respectively. He has been working toward the Ph.D. degree in robotics with the Department of Cognitive Robotics, TU Delft, Delft, The Netherlands, since 2019.

His research focuses on interactive imitation learning applied to robot manipulation.

Dr. Franzese has been an ELLIS Member for artificial intelligence, since 2022.

Leandro de Souza Rosa received the B.Sc. degree in computer engineering and the Ph.D. degree in computer science from the Institute of Mathematics and Computer Sciences, University of São Paulo, São Paulo, Brazil (which was partially developed at the Department of Electrical and Electronic Engineering, Imperial College London, U.K.), in 2013 and 2019, respectively.

He was a Postdoc Researcher with the Istituto Italiano di Tecnologia, Genova, Italy, and is currently with the Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, Delft, The Netherlands.

Tim Verburg received the B.Sc. degree in mechanical engineering with a minor in robotics from TU Delft, Delft, The Netherlands, in 2021, where he is currently working toward the M.Sc. degree in robotics.

His research interests include mechatronic design and robot learning.

Luka Peternel (Member, IEEE) received the Ph.D. in robotics from the University of Ljubljana, Ljubljana, Slovenia, in 2015.

He conducted Ph.D. studies with the Department for Automation, Biocybernetics and Robotics, Jožef Stefan Institute in Ljubljana, Ljubljana, from 2011 to 2015, and with the Department of Brain-Robot Interface, ATR Computational Neuroscience Laboratories, Kyoto, Japan, in 2013 and 2014, respectively. From 2015 to 2018, he was with the Human-Robot Interfaces and Physical Interaction Lab, Italian Institute of Technology, Genoa, Italy. Since 2019, he has been an Assistant Professor with Cognitive Robotics, TU Delft, Delft, The Netherlands.

Jens Kober (Senior Member, IEEE) received the Ph.D. degree in engineering from TU Darmstadt, Darmstadt, Germany, in 2012.

He is currently an Associate Professor with TU Delft, Delft, The Netherlands. He was a Postdoctoral Scholar jointly with CoR-Lab, Bielefeld University, Bielefeld, Germany, and with Honda Research Institute Europe, Offenbach, Germany.

Dr. Kober was a recipient of the annually awarded Georges Giralt PhD Award for the best PhD thesis in robotics in Europe, the 2018 IEEE RAS Early Academic Career Award, the 2022 RSS Early Career Award, and was a recipient of an ERC Starting grant.