

A Learning-based Shared Control Architecture for Interactive Task Execution

Firas Abi-Farraj, Takayuki Osa, Nicolò Pedemonte, Jan Peters, Gerhard Neumann, Paolo Robuffo Giordano

▶ **To cite this version:**

Firas Abi-Farraj, Takayuki Osa, Nicolò Pedemonte, Jan Peters, Gerhard Neumann, et al.. A Learning-based Shared Control Architecture for Interactive Task Execution. IEEE Int. Conf. on Robotics and Automation, ICRA'17, May 2017, Singapore, Singapore. hal-01482137

HAL Id: hal-01482137
https://inria.hal.science/hal-01482137

Submitted on 3 Mar 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Learning-based Shared Control Architecture for Interactive Task Execution

Firas Abi-Farraj[1], Takayuki Osa[2], Nicoló Pedemonte [1], Jan Peters[2,3], Gerhard Neumann[2], Paolo Robuffo Giordano[1]

***Abstract*— Shared control is a key technology for various robotic applications in which a robotic system and a human operator are meant to collaborate efficiently. In order to achieve efficient task execution in shared control, it is essential to predict the desired behavior for a given situation or context in order to simplify the control task for the human operator. This prediction is obtained by exploiting Learning from Demonstration (LfD), which is a popular approach for transferring human skills to robots. We encode the demonstrated behavior as trajectory distributions and generalize the learned distributions to new situations. The goal of this paper is to present a shared control framework that uses learned expert distributions to gain more autonomy. Our approach controls the balance between the controller's autonomy and the human preference based on the distributions of the demonstrated trajectories. Moreover, the learned distributions are autonomously refined from collaborative task executions, resulting in a master-slave system with increasing autonomy that requires less user input with an increasing number of task executions. We experimentally validated that our shared control approach enables efficient task executions. Moreover, the conducted experiments demonstrated that the developed system improves its performances through interactive task executions with our shared control.**

I. INTRODUCTION

Robotic bilateral teleoperation is a key technology for allowing human operators to be in (partial) control of a robotic system in, e.g., hazardous/remote environments such as nuclear sites, deep oceans or the outer space by exploiting a combination of visual and force feedback. However, controlling a complex robotic system using human inputs alone is often difficult and may require special skills or dedicated training [1], [2]. Assisted teleoperation frameworks, in which the human inputs are 'mediated' by the (partial) autonomy of the controlled robotic systems, have thus been proposed for reducing the workload requirements of the human operator and improving her/his performance [2]–[4]. Examples in this sense range from virtual fixtures [5], [6], which are task-dependent and require continuous human input, to more complicated shared control schemes aiming at profiting from the human's supervisory capabilities in guiding an autonomous system [4], [7].

[1]F. A. Farraj, Nicoló Pedemonte, and P. Robuffo Giordano are with CNRS at Irisa and Inria Rennes Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France. {firas.abi-farraj, nicolo.pedemonte, prg}@irisa.fr

[2]T. Osa, J. Peters, and G. Neumann are with Institut für Intelligente Autonome Systeme, Technische Universität Darmstadt, 64289, Germany.

[3]J. Peters is with Max-Planck-Institut für Intelligente Systeme, Tübingen, 72076, Germany. {osa, neumann, peters}@ias.tu-darmstadt.de

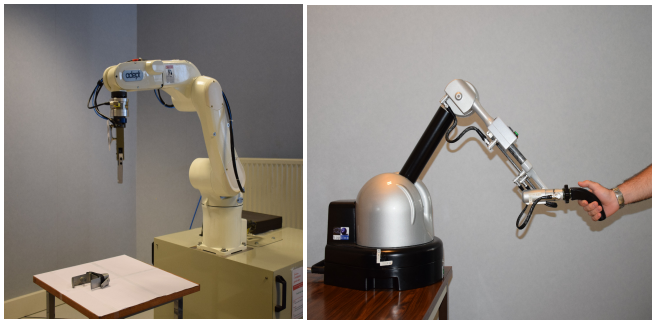


Fig. 1. The experimental setup showing the slave robotic arm on the left and the master haptic arm on the right.

Finding the proper balance between the level of human intervention and that of robot autonomy remains, however, in most cases an open (and task-dependent) problem. This balance is indeed often tuned heuristically and, moreover, is not adapted during the task execution, a limitation that can lead to an increase of the operator's workload and a decrease of her/his efficiency. On the other hand, a shared control architecture could greatly benefit from the ability to adapt *online* the operator/autonomy balance as a function of the 'confidence' of the robot autonomy in realizing the shared task. This would (i) allow the operator to intervene only when strictly needed (when the robot autonomy is more likely to fail in fulfilling the task), and (ii) provide the operator with an informative feedback for guiding her/his actions only when the autonomous component is expected to perform well. However, to the best of our knowledge, no established way has yet been proposed for taking a step in this direction.

On the other hand, learning from demonstration (LfD) or programming by demonstration (PbD) have been proposed as an intuitive way to program robotic motions [8], [9]. In LfD methods, the distribution of the demonstrated trajectories is often modeled, and the learned distribution is leveraged for generalizing the demonstrated behaviors [10], [11]. Recent work on LfD showed that the variance of the demonstrated trajectories can be used to adaptively control the robot behaviors [12]. Leveraging the distribution of the demonstrated trajectories into the design of shared control frameworks seems, therefore, a meaningful/promising approach for obtaining a shared control framework able to adjust *online* the balance between the robot autonomy and the human preference.

To this end, in this paper we consider that the distribution of the demonstrated trajectories indicates the preference of an expert operator on the robot motion: a high variance in

the demonstrated trajectories is assumed to indicate a weak preference of the operator, whereas a low variance is taken as a strong preference. Based on this assumption, we use the variance of the trajectory to control the balance between the controllers autonomy and the human intervention. In particular, when the variance of the demonstrated trajectories is low, the user is fed back with strong force cues meant to minimize any deviation from the 'nominal' (learned) trajectory. On the contrary, as the variance increases, the force cues are suitably attenuated, thus providing the operator with the possibility (and feeling) of freely moving the robotic system along any direction of interest. Through interactive task execution using our shared control framework, we can obtain additional trajectories that are executed under the supervision of the human operator. By aggregating the newly obtained data, we refine the learned trajectory distribution. Therefore, the performance of our shared control system improves through interactive task executions.

This work was conducted in the context of the European H2020 Robotic Manipulation for Nuclear Sort and segregation (RoMaNS) project[1]. In the project scenario, a human operator teleoperates a robotic arm with the aim of approaching and grasping nuclear waste for sort and segregation purposes. The paper particularly tackles the pre-grasp approach phase in the project providing an efficient LfD-based shared control architecture for trajectory following.

The rest of the paper is organized as follows. In Sect. II, an overview on the state of the art is presented. Sec III introduces the general shared control framework and Sec IV details the algorithm for the generation of conditional trajectory distributions from demonstrations, while the details of the shared control architecture are presented in Sect. V. Trajectory refinement from executions is described in Section V-A after which section VI reports the experimental results, and Sect. VII concludes the paper discussing some future directions.

II. Related Work

Shared control appears in various applications ranging from robotic surgery [13], to space robotics [14] and maintenance of nuclear sites [15]. Moreover, the use of haptic feedback in assisted teleoperation systems has been proved to be efficient and intuitive for the user. However, as explained, the definition of these haptic cues and the level of the systems autonomy w.r.t. the users authority remain open points in most applications.

The level of autonomy of the assistive system can be defined as a fixed or a variable entity. In fixed assistance systems, the tasks are distributed between the operator and the autonomous controller and cannot be re-assigned [16] whereas in variable assistance systems the task assignment is changing throughout the task execution [17], [18]. This variation is mostly pre-determined and does not take into account the 'confidence' of the autonomous component in achieving the task at hand. Moreover, the inputs from the human operator are often isolated from those of the autonomous controller, with each one in charge of controlling different independent variables. In [19], Enes and Book presented a blended shared control algorithm which they tested on a 1-dof navigation problem. The users commands were evaluated w.r.t. the optimal solution of the problem and a virtual spring with a constant stiffness was implemented for the haptic feedback. Another blended pseudo-admittance shared control framework for trajectory following has been presented in [20]. The virtual fixture was defined around the planned trajectory and a constant-stiffness virtual mass-spring-damper system provided haptic feedback to the operator.

On the other hand, in LfD methods, the distribution of the demonstrated trajectory is often modeled with statistical methods. The framework called ProMP models the distribution of the trajectories in the parameter space [10] while the method in [11] modeled the distribution of the demonstrated trajectories at each time step using Gaussian Processes. These studies showed that the demonstrated behaviors can be generalized to new situation by modeling the distribution of the demonstrated trajectories. The learned distribution of demonstrations can then be used for designing control schemes. For instance, in [12] the robot learns collaborative skills and adapts its impedance behavior from demonstrations. However, to the best of our knowledge, no prior work has been proposed for leveraging the distribution of the demonstrated trajectories for controlling the balance between the controllers autonomy and the human inputs.

Moreover, although we assume that the demonstrations from human experts are available, it is often expensive to obtain enough demonstrations for building an initial dataset rich enough for capturing all the possible conditions. However, in a shared control framework the trajectory obtained through the actual execution of the task can be also reused to update the model of the trajectory distribution. Calinon et al. described the incremental learning of the trajectory distribution in [21]. The method presented in [21] uses kinesthetic teaching for modifying the trajectory, which can be cumbersome if a manipulator has many degrees of freedom. In contrast, we will use our shared control architecture so that human operator can intuitively modify the trajectory. Regarding online learning, Ross et al. proposed a data aggregation approach and clarified the no-regret property of their approach [22]. This approach can be used in various applications of imitation learning where additional data of experts' demonstration is available. Chernova et al. proposed to refine the policy by actively requesting additional demonstrations [23]. In this method, the confidence of the autonomous agent is computed and if the agent's confidence is low, additional demonstrations are requested. By using this kind of incremental aggregation of the demonstrated data, the policy for generating actions can be improved.

III. PRELIMINARIES

In this work, we consider a bilateral teleoperation system consisting of a 6-dof torque-controlled master device and

[1]http://www.h2020romans.eu/

a velocity-controlled 6-dof slave manipulator arm. Figure 1 illustrates the experimental setup.

Let $\boldsymbol{x}_M = (\boldsymbol{p}_M, \boldsymbol{\phi}_M) \in \mathbb{R}^6$ and $\boldsymbol{x}_S = (\boldsymbol{p}_S, \boldsymbol{\phi}_S) \in \mathbb{R}^6$ be the position and (a minimal representation of) the orientation of the master/slave end-effectors in their respective base frames. The master device is modeled as a generic (gravity pre-compensated) mechanical system

$$\boldsymbol{M}(\boldsymbol{x}_M)\ddot{\boldsymbol{x}}_M + \boldsymbol{C}(\boldsymbol{x}_M, \dot{\boldsymbol{x}}_M)\dot{\boldsymbol{x}}_M = \boldsymbol{\tau} + \boldsymbol{\tau}_h, \tag{1}$$

where $\boldsymbol{M}(\boldsymbol{x}_M) \in \mathbb{R}^{6\times6}$ is the positive-definite and symmetric inertia matrix (in the 'task' coordinates $\boldsymbol{x}_M$), $\boldsymbol{C}(\boldsymbol{x}_M, \dot{\boldsymbol{x}}_M) \in \mathbb{R}^{6\times6}$ accounts for Coriolis/centrifugal terms, and $\boldsymbol{\tau}, \boldsymbol{\tau}_h \in \mathbb{R}^6$ are the control and human forces, respectively, which are applied at the master end-effector. Note that the control term $\boldsymbol{\tau}$ will be exploited for providing force feedback cues to the user.

As for the slave, we assume instead that its end-effector (where the gripper is mounted, see Fig. 1) can be controlled at the velocity level (as in most industrial manipulator arms) so that

$$\dot{\boldsymbol{x}}_S = \begin{bmatrix} \dot{\boldsymbol{p}}_S \\ \dot{\boldsymbol{\phi}}_S \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T}^{-1}(\boldsymbol{\phi}_S) \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_S \\ \boldsymbol{\omega}_S \end{bmatrix} \tag{2}$$

$$= \boldsymbol{L}(\boldsymbol{\phi}_S) \begin{bmatrix} \boldsymbol{v}_S \\ \boldsymbol{\omega}_S \end{bmatrix}, \tag{3}$$

where $(\boldsymbol{v}_S, \boldsymbol{\omega}_S) \in \mathbb{R}^6$ are the available linear/angular end-effector velocity commands and $\boldsymbol{T}(\boldsymbol{\phi}_S) \in \mathbb{R}^{3\times3}$ is the usual transformation matrix such that

$$\boldsymbol{\omega}_S = \boldsymbol{T}(\boldsymbol{\phi}_S)\dot{\boldsymbol{\phi}}_S. \tag{4}$$

Finally, in the context of this work, we assume that the slave pose $\boldsymbol{x}_S$ is directly coupled to the master pose $\boldsymbol{x}_M$ (actuated by the human operator via $\boldsymbol{\tau}_h$) modulo a possible constant roto-translation/scaling factor which, w.l.o.g., is here neglected for simplicity. The slave velocity commands are then chosen as

$$\begin{bmatrix} \boldsymbol{v}_S \\ \boldsymbol{\omega}_S \end{bmatrix} = \boldsymbol{L}^{-1}(\boldsymbol{\phi}_S)(\lambda(\boldsymbol{x}_M - \boldsymbol{x}_S) + \dot{\boldsymbol{x}}_M) \tag{5}$$

where $\lambda > 0$ is a control gain.

We also introduce a context vector $\boldsymbol{s}$ which characterizes the task. For example, the context vector could contain the position of the object we want to grasp. We will use the context vector to adapt the desired trajectories of the master-slave system.

In the next sections, we will discuss how to obtain representative statistics from a set of demonstrated master trajectories $\boldsymbol{x}_M(t)$ and how this statistics can then be exploited for designing the force cues $\boldsymbol{\tau}$ in (1).

IV. MODELLING DEMONSTRATED TRAJECTORY DISTRIBUTIONS

During the learning phase, we assume that a skilled human operator demonstrates a number of feasible trajectories for letting the slave robot approaching a target object of interest. During this phase the slave is commanded to follow the master position $\boldsymbol{x}_M(t)$ as in (5), but no force feedback is provided to the user ($\boldsymbol{\tau} = \boldsymbol{0}$). We then model a distribution of these demonstration trajectories by using indepdent Gaussian distributions for each time step t, i.e.,

$$p(\boldsymbol{x}_M(t)) \sim \mathcal{N}(\boldsymbol{\mu}(t), \boldsymbol{\Sigma}(t)) \tag{6}$$

where $\boldsymbol{\mu}(t)$ and $\boldsymbol{\Sigma}(t)$ are the mean and variance of $\boldsymbol{x}_M(t)$. Assuming that the trajectory is given as a sequence of states of the system, i.e., $\boldsymbol{\xi} = [\boldsymbol{x}_M(t_0), \ldots, \boldsymbol{x}_M(T)]$, the distribution of the trajectory $\boldsymbol{\xi}$ can be modeled as

$$p(\boldsymbol{\xi}) = \prod_{t=0}^{T} \mathcal{N}(\boldsymbol{\mu}(t), \boldsymbol{\Sigma}(t)), \tag{7}$$

where T is the number of the time steps of the trajectory.

We assume that the demonstrations are available under various contexts $\boldsymbol{s}^i$. In this case, we can model the conditional distribution of the demonstrated trajectories given the context in order to generalize the demonstrated trajectories to new situations [11], [24], [25]. Here, we use Locally Weighted Regression (LWR) to model this distribution [26], [27]. Although we use LWR in this work, our approach is not limited to specific regression methods. Other regression methods such as Gaussian Mixture Regression can be also used to model the distribution.

We assume a dataset $\mathcal{D} = \{\boldsymbol{\xi}^i, \boldsymbol{s}^i\}_{i=1}^N$ of trajectories and context vectors is available. Given a new query context $\boldsymbol{s}_{\text{test}}$, the locality weight for the ith sample can be computed as

$$w_i = \exp\left(-\frac{(\boldsymbol{s}_i - \boldsymbol{s}_{\text{test}})^\top(\boldsymbol{s}_i - \boldsymbol{s}_{\text{test}})}{h}\right). \tag{8}$$

where h is a constant that determines the bandwidth of the locality kernel. As in LWR, we can now compute a local linear model by using a weighted linear ridge regression, i.e.,

$$\begin{bmatrix} \boldsymbol{a}_t^\top \\ \boldsymbol{A}_t^\top \end{bmatrix} = (\boldsymbol{S}^\top \boldsymbol{W} \boldsymbol{S} + \lambda \boldsymbol{I})^{-1} \boldsymbol{S}^\top \boldsymbol{W} \boldsymbol{X}_t,$$

where the matrix $\boldsymbol{S} = [\tilde{\boldsymbol{s}}^1, \cdots, \tilde{\boldsymbol{s}}^N]^\top$ contains all training context vectors $\tilde{\boldsymbol{s}}^i = [1; \boldsymbol{s}^i]$ which have been extended by a bias term, the matrix $\boldsymbol{W} = \text{diag}([w_i])$ is a diagonal matrix containing the weightings w_i and the matrix $\boldsymbol{X}_t = [\boldsymbol{x}_M^1(t), \cdots, \boldsymbol{x}_M^N(t)]^\top$ contains all state samples obtained for time step t.

The estimated mean state for time t is then given by

$$\mathbb{E}[\boldsymbol{x}_M(t)|\boldsymbol{s}_{\text{test}}] = \boldsymbol{A}_t \boldsymbol{s}_{\text{test}} + \boldsymbol{a}_t. \tag{9}$$

Similarly, we can compute the conditional covariance over $\boldsymbol{x}_M(t)$ given our query context $\boldsymbol{s}_{\text{test}}$ using a weighted maximum likelihood estimate, i.e.,

$$\boldsymbol{\Sigma}_{\boldsymbol{x}_M|\boldsymbol{s}_{\text{test}}}(t) = \frac{\sum_{i=1}^N w_i (\boldsymbol{x}_M^i(t) - \boldsymbol{\mu}^i)(\boldsymbol{x}_M^i(t) - \boldsymbol{\mu}^i)^\top}{\sum_{i=1}^N w_i}, \tag{10}$$

where $\boldsymbol{\mu}^i = \boldsymbol{A}_t \boldsymbol{s}^i + \boldsymbol{a}_t$ is the estimated mean for sample i.

V. SHARED CONTROL GUIDED BY LEARNED TRAJECTORY DISTRIBUTIONS

In our application, we consider the position of the target object $\boldsymbol{p}_O = (x_O, y_O, z_O)$ as the context $\boldsymbol{s}$ upon which the conditional distribution of the trajectory is generated in (9) and (10). To simplify the notation, we will refer to $\mathbb{E}[\boldsymbol{x}_M(t)|\boldsymbol{s}_{\text{test}}]$ by $\boldsymbol{x}_{M,d}(t) = (\boldsymbol{p}_{M,d}(t), \boldsymbol{\phi}_{M,d}(t))$ in the following section. Moreover, we will denote by $\boldsymbol{\Sigma}_p(t) \in \mathbb{R}^{3\times3}$ and $\boldsymbol{\Sigma}_\phi(t) \in \mathbb{R}^{3\times3}$ the 3×3 block diagonal elements of the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{x}_M|\boldsymbol{s}_{\text{test}}}(t) \in \mathbb{R}^{6\times6}$ computed in (10).

Having estimated the distribution from demonstrations, we now present the design of the force cues $\boldsymbol{\tau}$ in (1). Our approach is as follows: we treat the mean of the conditional distribution $\boldsymbol{x}_{M,d}(t)$ as a *desired* trajectory for the master device. Indeed, this trajectory represents, in some sense, the 'best/nominal' approaching trajectory (to the considered target location) from the expert user's demonstrations collected during the learning phase. The force cues $\boldsymbol{\tau}$ will then attempt to steer the master device along the desired $\boldsymbol{x}_{M,d}(t)$ with a 'stiffness', roughly speaking, inversely proportional to the variance of the generated conditional distribution. This will effectively tune the degree of maneuverability of the master device around the nominal trajectory $\boldsymbol{x}_{M,d}(t)$, and, thus, provide the operator with an increased situational awareness (more freedom to deviate from $\boldsymbol{x}_{M,d}(t)$ for large variances, and the converse for small variances).

Let $\boldsymbol{\tau} = (\boldsymbol{\tau}_p, \boldsymbol{\tau}_\phi)$, where $\boldsymbol{\tau}_p \in \mathbb{R}^3$ and $\boldsymbol{\tau}_\phi \in \mathbb{R}^3$ are the forces/torques acting on the position/orientation $\boldsymbol{p}_M$, $\boldsymbol{\phi}_M$ of the master end-effector, and define $\boldsymbol{e}_p = \boldsymbol{p}_{M,d} - \boldsymbol{p}_M$ and $\boldsymbol{e}_\phi = \boldsymbol{\phi}_{M,d} - \boldsymbol{\phi}_M$ as the position/orientation errors. Following the classical literature on task-space impedance control [28], we then design

$$\begin{bmatrix} \boldsymbol{\tau}_p \\ \boldsymbol{T}^\top(\boldsymbol{\phi})\boldsymbol{\tau}_\phi \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}_p & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_\phi \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{p}}_{M,d} \\ \ddot{\boldsymbol{\phi}}_{M,d} \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_p & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B}_\phi \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{e}}_p \\ \dot{\boldsymbol{e}}_\phi \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_p & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_\phi \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_p \\ \boldsymbol{e}_\phi \end{bmatrix} \tag{11}$$

where $\boldsymbol{M}_p(\boldsymbol{p})$ and $\boldsymbol{M}_\phi(\boldsymbol{\phi})$ are the 3×3 block diagonal elements of the master inertia matrix $\boldsymbol{M}(\boldsymbol{x}_M)$ in (1) associated to the coordinates $\boldsymbol{p}$ and $\boldsymbol{\phi}$, ($\boldsymbol{B}_p$, $\boldsymbol{B}_\phi$) and ($\boldsymbol{K}_p$, $\boldsymbol{K}_\phi$) are 3×3 damping and stiffness matrix terms, and $T(\boldsymbol{\phi})$ is the transformation matrix defined in (4).

As mentioned earlier, a decrease in the variance of the generated distribution is assumed to indicate more confidence in the generated trajectory. This confidence is to be reflected as an increase in the stiffness of the virtual spring and the force cues fed to the human operator and vice versa. To this end, we consider the eigenvalue decomposition of the (symmetric and positive definite) covariance matrixes

$$\begin{cases} \boldsymbol{\Sigma}_p &= \mathbf{V}_p \mathbf{S}_p \mathbf{V}_p^\top, \\ \boldsymbol{\Sigma}_\phi &= \mathbf{V}_\phi \mathbf{S}_\phi \mathbf{V}_\phi^\top, \end{cases} \tag{12}$$

with $\mathbf{S}_p = \text{diag}(\sigma_{p_i})$ and $\mathbf{S}_\phi = \text{diag}(\sigma_{\phi_i})$. The desired stiffness matrices ($\boldsymbol{K}_p$, $\boldsymbol{K}_\phi$) are then defined as

$$\begin{cases} \boldsymbol{K}_p &= \mathbf{V}_p \boldsymbol{K}_{p,0} e^{-\alpha_p \mathbf{S}_p} \mathbf{V}_p^\top \\ \boldsymbol{K}_\phi &= \mathbf{V}_\phi \boldsymbol{K}_{\phi,0} e^{-\alpha_\phi \mathbf{S}_\phi} \mathbf{V}_\phi^\top \end{cases} \tag{13}$$

where $\boldsymbol{K}_{p,0} = \text{diag}(k_{p,i}) > 0$, $\boldsymbol{K}_{\phi,0} = \text{diag}(k_{\phi,i}) > 0$, $\alpha_p > 0$, and $\alpha_\phi > 0$.

This stiffness design achieves the desired behavior: indeed, by focusing on the first position term (the second one being equivalent), the chosen $\boldsymbol{K}_p$ will implement a virtual spring of value $k_{p,i}e^{-\alpha\sigma_{p,i}}$ on each of the principal axes of $\boldsymbol{\Sigma}_p(t)$. The stifness will then range from the maximum values $k_{p,i}$ for small variances $\sigma_{p,i} \approx 0$, to negligible values for large variances $\sigma_{p,i}$, with the parameter α_p governing the decrease rate.

Finally, the damping terms are designed, as usual, in order to obtain a critically-damped closed-loop behavior [29]

$$\begin{cases} \boldsymbol{B}_p(t) &= 2(M_p^{-1/2} K_p(t) M_p^{-1/2})^{1/2} \\ \boldsymbol{B}_\phi(t) &= 2(M_\phi^{-1/2} K_\phi(t) M_\phi^{-1/2})^{1/2} \end{cases}. \tag{14}$$

A. Trajectory Refinement through Interactive Task Executions

By using the shared control architecture described in Section V, we can obtain a new sample of the trajectory and the context. The obtained sample can be used to refine the model of the trajectory distribution by simply aggregating it to the dataset. When a new trajectory is obtained, we examine the information gain which can be obtained by adding it to the dataset. Using the dataset $\mathcal{D}$, we model the joint distribution of the context and the state of the system at time t as a Gaussian distribution

$$p_{\mathcal{D}}\left(\begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{x}_M(t) \end{bmatrix}\right) \sim \mathcal{N}_{\mathcal{D}}(\boldsymbol{\mu}_{\mathcal{D}}(t), \boldsymbol{\Sigma}_{\mathcal{D}}(t)). \tag{15}$$

Let $\mathcal{D}'$ denote the dataset which can be obtained by aggregating the data as $\mathcal{D}' = \mathcal{D} \cup \mathcal{D}_{\text{new}}$. The information gain from the newly obtained data is given by Kullback-Leibler divergence [30] as

$$\begin{aligned} \mathcal{I}(\mathcal{D}, \mathcal{D}_{\text{new}}) &= \sum_{t=0}^{T} D_{\text{KL}}(p_{\mathcal{D}'} || p_{\mathcal{D}}) \\ &= \sum_{t=0}^{T} D_{\text{KL}}(\mathcal{N}_{\mathcal{D}'} || \mathcal{N}_{\mathcal{D}}) \\ &= \frac{1}{2} \sum_{t=0}^{T} \left(\log\left(\frac{\det \Sigma_{\mathcal{D}}(t)}{\det \Sigma_{\mathcal{D}'}(t)}\right) - n + \text{tr}(\Sigma_{\mathcal{D}}^{-1}(t)\Sigma_{\mathcal{D}'}(t)) \right. \\ &\quad \left. + (\mu_{\mathcal{D}}(t) - \mu_{\mathcal{D}'(t)})^\top \Sigma_{\mathcal{D}}^{-1}(t)(\mu_{\mathcal{D}}(t) - \mu_{\mathcal{D}'}(t)) \right), \end{aligned} \tag{16}$$

where n is the dimension of $[\boldsymbol{s}^\top, \boldsymbol{x}_M^\top(t)]$. If the information gain $\mathcal{I}(\mathcal{D}, \mathcal{D}_{\text{new}})$ is larger than the threshold $\mathcal{I}_0$, the newly obtained data is aggregated to the dataset as $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_{\text{new}}$. Using the information criterion, we can keep the dataset as compact as possible. Our approach for learning shared control from interactive task executions is summarized in Algorithm 1.

Through interactive learning, the trajectory distribution is adapted by the operator's preference. If the operator's

Algorithm 1 Learning Trajectory Distribution through Interactive Task Executions

Input: dataset of the trajectories demonstrated by experts and the contexts of each demonstration $\mathcal{D}$, information gain threshold $\mathcal{I}_0$
repeat
  Model the trajectory distribution $p(\boldsymbol{\xi}|\boldsymbol{s})$
  Update the parameters of the shared control
  Perform the task under the context s_{new} using the shared control
  Record the obtained data $\mathcal{D}_{\text{new}} = \{\boldsymbol{\xi}, \boldsymbol{s}_{\text{new}}\}$
  Evaluate the information gain from the new data $\mathcal{I}(\mathcal{D}, \mathcal{D}_{\text{new}})$
  if $\mathcal{I}(\mathcal{D}, \mathcal{D}_{\text{new}}) > \mathcal{I}_0$ **then**
    Aggregate the dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_{\text{new}}$
  end if
until the trajectory distribution learned

preference is stationary, the trajectory distribution induced by the learned model converges to the trajectory distribution induced by the operator as the number of the trajectory samples increase. Therefore, the required amount of control input from the operator is expected to decrease as the learned trajectory distribution is more and more refined.

VI. EXPERIMENTAL RESULTS

We evaluated our approach using the experimental test-bed shown in Fig. 1. The slave side is an Adept Viper s850 6-dof serial manipulator equipped with a gripper whereas the master consists of the Haption VIRTUOSE 6D haptic device[2].

In the considered task, the human operator is required to steer the slave arm towards a target object with the intention of grasping it. The target object is assumed to be fixed and the slave and the master arms go back to their respective pre-defined initial positions after every iteration. A video is attached to this paper showing the proposed approach, the test bed and the performed experiments. A human operator demonstrated the described task 20 times using the master-slave system. During the demonstrations, the master manipulator was totally compliant and the user did not receive any force feedback.

To evaluate our algorithm, we used data from demonstrations performed under biased conditions. This bias leads to the generation of 'non-optimal' trajectory distributions given certain contexts which is essential in order to induce a significant analysable intervention from the human operator. To introduce the bias, initial demonstrations were performed with the target object placed 16 cm lower than in the testing phase (Fig. 2). A human operator performed the task using our shared control framework, and the learned trajectory distribution was refined by aggregating the executed trajectory to the dataset after every iteration. This procedure was repeated five times.

[2]www.haption.com

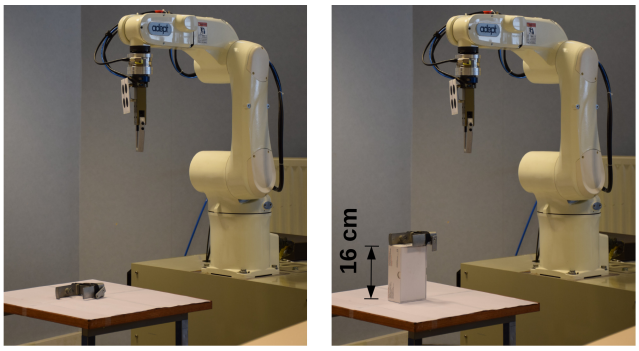


Fig. 2. The experimental setup. The initial demonstrations were performed with an object on a table. The learned trajectory distribution was then tested with the object on a box.

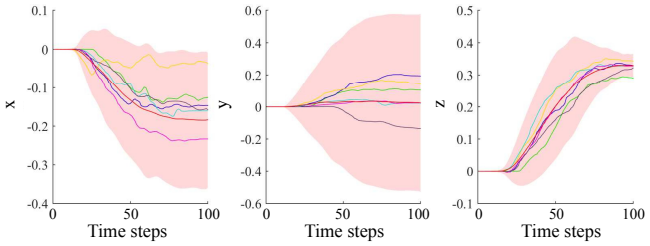


Fig. 3. The three positional components of the pose of the end effector of the master arm at every time step t during each of the 20 demonstrations

The initially demonstrated trajectories are shown in Fig. 3. To avoid redundancy, the displayed plots correspond only to the translational position of the end effector although both, positions and orientations of the trajectories, were planned and executed as described in the previous sections.

Fig. 4 shows the planned trajectory distribution (plotted in red with the shaded region reflecting the value of the variance) versus the executed trajectory (in blue) for the first, third and last iterations of the experiments. The difference between the planned trajectory distribution and the trajectory executed under the human operator's supervision was large in the first iteration as shown in Fig. 4(a) where the executed trajectory is mainly monotone while the planned distribution shifts from decreasing to increasing as in x and z. Moreover, the planned trajectory was jerky with lots of vibrations due to the prediction uncertainty. Meanwhile, Fig. 4(b), which corresponds to the third iteration of the experiment, shows a notable improvement in the behavior of the planned trajectory distribution which is now more in line with the executed trajectory. The planned trajectories are, as expected, even better in the fifth iteration (Fig. 4(c)). Although the offset between the planned and executed trajectories still remained after five task executions, we think these are acceptable results. As long as the operator holds the master manipulator, an unintentional force is applied due to the inertia of the operator's hand. However, since the operator did not try to modify the trajectory in the final two iterations, we consider that the offset is acceptable. On the other hand, the smoothness of the trajectory has significantly improved when comparing the planned trajectory generated during the fifth

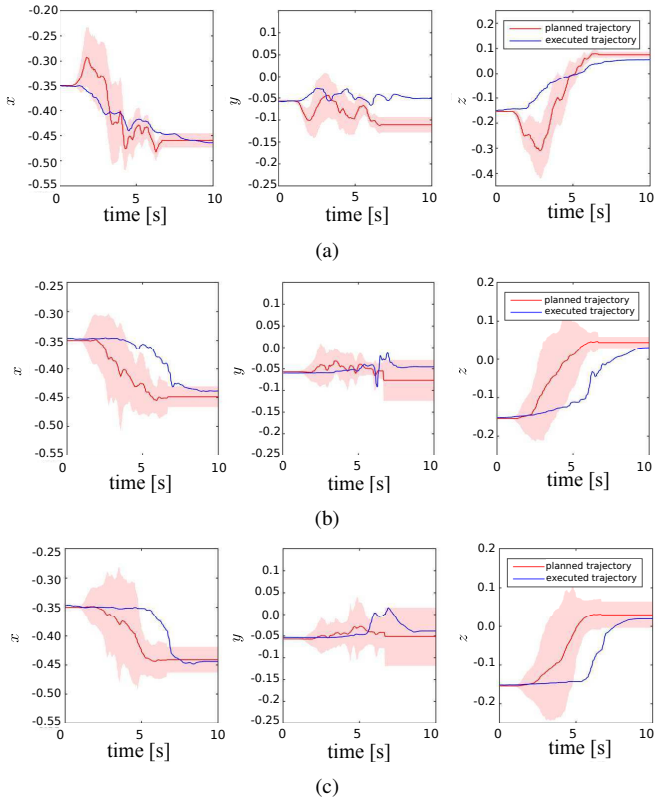


Fig. 4. Improvement of the trajectory prediction. Blue lines represent the executed trajectory, and red lines represent the distribution of the predicted trajectory. (a)The executed trajectory and the predicted trajectory based on the initial dataset. (b)The executed trajectory and the predicted trajectory after the aggregation of three additional trajectories obtained from the interactive task executions. (c)The executed trajectory and the predicted trajectory after the aggregation of five additional trajectories obtained from the interactive task executions.

iteration with the one generated in the first. This is mostly visible in x and z since the displacement in the y-axis was limited in the described experiment.

Metrics of the intervention of the human operator in each of the 5 iterations of the experiment are shown in Fig. 5. The magnitude of the forces and the torques exerted by the human operator at the master side is clearly decreasing with the progression of the experiment. However, as a result of the continuous interaction between the human operator and the end effector of the master arm, these forces will converge to a range but not to zero. The slight increase in the linear force for iteration 4 is noise resulting from the mentioned interaction.

This result illustrates the efficiency of the shared control architecture in providing the human operator with the needed informative force cues reflecting his deviation from the generated trajectory distribution. Moreover, it reflects the adaptability of the learning routine and its swift compliance over iterations with the inputs dictated by the human operator resulting in less intervention from the human operator over subsequent task executions.

In addition, the information gain from each executed trajectory (defined in (16)) was analyzed in this experiment

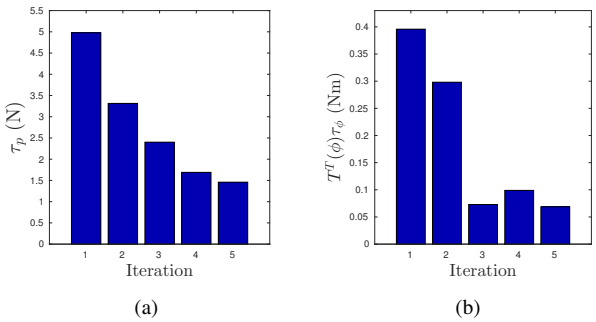


Fig. 5. Metrics of the applied human intervention during task execution. The intervention from the human operator decreased after interactive task executions. (a)Mean force exerted at the master manipulator. (b)Mean torque exerted at the master manipulator.

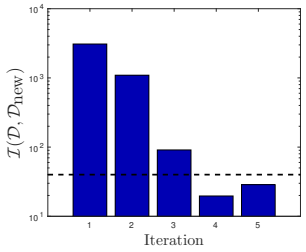


Fig. 6. Information gain of the additional data. The information gain decreases as the number of the data samples increases.

as well to check its effectiveness as a token in deciding whether to aggregate the newly executed trajectory to the learning dataset or not. Fig. 6 recounts the behavior of the indicator over time showing a decrease in the information gain with every iteration. The horizontal line signifies the threshold above which an execution is aggregated to the dataset. However, for the sake of testing the impact of this aggregation, all the executions were exceptionally aggregated to the dataset in this experiment.

The mentioned figure shows that the most significant information gain was achieved during the first three iterations after which it dropped significantly. This result is in line with Fig. 4 which depicts the planned and executed trajectories during the first, third and fifth iterations. The figure shows that the planned trajectory changed significantly between the first and the third iterations. In contrast, the change was more limited between the third and the fifth. We can conclude that the aggregation of the last two executions (for which the information gain was below the threshold in Fig. 6) into the dataset had only little impact on the final results. The observed behavior thus validates the effectiveness of the chosen measure for informative training samples.

VII. CONCLUSION AND DISCUSSION

We presented a shared control framework that is guided by a trajectory distribution learned from human experts. By exploiting the learned distributions, we can adaptively control the balance between autonomy of the controller and the human intervention. We used the variance of the demonstrated

trajectories as an indicator of the human experts' preference, and the haptic feedback exerted at the master system was controlled accordingly. Experimental results show that the learned trajectory distribution was refined and the performance of our shared control improved through interactive task executions. In future work, we will analyze the stability of the shared control architecture and evaluate the efficacy of our system by performing tests with subjects.

In the experiments, we showed that our shared control framework learns the demonstrated behaviors through interactive task executions. However, to further evaluate the efficacy of our shared control framework we will perform tests with subjects on more complex tasks in future work. Moreover, although we used Euler angles to represent the orientation, unit quaternions or other singularity-free representations can be used. However, the representation with unit quaternion requires modeling the distribution with unit norm constraints, which is not well-investigated in literature and would require an extension of the learning method.

So far, we addressed the shared control for a single task, e.g. approaching objects. It is interesting to address the problem for more complicated tasks consisting, for example, of a sequence of primitive motions. Another potential extension of the shared control algorithm is the incorporation of visual information, as in [3], to improve the robustness of the system towards loose joints and miscalibrations. The stability of the teloperation loop is to be analyzed as well. In particular, the presence of a time-varying stiffness needs to be analysed even though no instabilities were reported during the experiments.

ACKNOWLEDGMENT

This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No #645582 (RoMaNS).

REFERENCES

[1] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1608–1639, 2010.

[2] A. Franchi, C. Secchi, M. Ryll, H. H. Bulthoff, and P. Robuffo Giordano, "Shared control : Balancing autonomy and human assistance with a group of quadrotor uavs," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 57–68, Sept 2012.

[3] F. Farraj, N. Pedemonte, and P. Robuffo Giordano, "A visual-based shared control architecture for remote telemanipulation," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[4] C. Masone, P. Robuffo Giordano, H. H. Bülthoff, and A. Franchi, "Semi-autonomous Trajectory Generation for Mobile Robots with Integral Haptic Shared Control," in *2014*, 2014, pp. 6468–6475.

[5] J. J. Abbott, P. Marayong, and A. M. Okamura, "Haptic virtual fixtures for robot-assisted manipulation," in *Robotics research*. Springer, 2007, pp. 49–64.

[6] L. B. Rosenberg, "Virtual fixtures: Perceptual tools for telerobotic manipulation," in *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*. IEEE, 1993, pp. 76–82.

[7] C. Masone, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano, "Interactive Planning of Persistent Trajectories for Human-Assisted Navigation of Mobile Robots," in *2012*, 2012, pp. 2641–2648.

[8] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, *Springer Handbook of Robotics*. Springer Berlin Heidelberg, 2008, ch. Robot Programming by Demonstration, pp. 1371–1394.

[9] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[10] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Probabilistic movement primitives," in *Proceedings of Advances in Neural Information Processing Systems 26*, 2013.

[11] T. Osa, N. Sugita, and M. Mitsuishi, "Online trajectory planning in dynamic environments for surgical task automation," in *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

[12] L. Rozo, S. Calinon, D. G. Caldwell, P. Jimenéz, and C. Torras, "Learning physical collaborative robot behaviors from human demonstrations," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 513–527, 2016.

[13] T. Osa, S. Uchida, N. Sugita, and M. Mitsuishi, "Hybrid rate-admittance control with force reflection for safe teleoperated surgery," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 5, pp. 2379–2390, Oct 2015.

[14] W. S. Kim, B. Hannaford, and A. K. Fejczy, "Force-reflection and shared compliant control in operating telemanipulators with time delay," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 2, pp. 176–185, Apr 1992.

[15] R. Schilling, "Telerobots in the nuclear industry: a manufacturer's view," *Industrial Robots*, vol. 19, no. 2, pp. 3–4, 1992.

[16] T. Debus, J. Stoll, R. D. Howe, and P. Dupont, "Cooperative human and machine perception in teleoperated assembly," in *Experimental Robotics VII*. Springer, 2001, pp. 51–60.

[17] A. D. Dragan and S. S. Srinivasa, *Formalizing assistive teleoperation*. MIT Press, July, 2012.

[18] D. Kortenkamp, D. Keirn-Schreckenghost, and R. P. Bonasso, "Adjustable control autonomy for manned space flight," in *Aerospace Conference Proceedings, 2000 IEEE*, vol. 7. IEEE, 2000, pp. 629–640.

[19] A. Enes and W. Book, "Blended shared control of zermelo's navigation problem," in *American Control Conference (ACC), 2010*. IEEE, 2010, pp. 4307–4312.

[20] J. J. Abbott and A. M. Okamura, "Pseudo-admittance bilateral telemanipulation with guidance virtualfixtures," *International Journal of Robotics Research*, vol. 26, no. 8, pp. 865–884, 2007.

[21] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2007.

[22] S. Ross, G. J. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011, pp. 627–635.

[23] S. Chernova and M. Veloso, "Interactive policy learning through confidence-based autonomy," *Journal of Artificial Intelligence Research*, vol. 34, no. 1, pp. 1–25, 2009.

[24] B. C. da Silva, G. Baldassarre, G. Konidaris, and A. Barto, "Learning parameterized motor skills on a humanoid robot," in *Proceedings of IEEE International conference on Robotics and Automation (ICRA)*, 2014.

[25] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent Service Robotics*, 2016.

[26] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, pp. 129–145, 1996.

[27] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning for control," *Artificial Intelligence Review*, vol. 11, no. 1, pp. 75–113, 1997.

[28] F. Caccavale, C. Natale, B. Siciliano, and L. Villani, "Six-dof impedance control based on angle/axis representations," *IEEE Trans. on Robotics*, vol. 15, no. 2, pp. 289–300, 1999.

[29] J. A. Gray and A. N. Andry, "A simple calculation for the critical damping matrix of a linear multidegree of freedom system," vol. 9, no. 6, pp. 379–380. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0093641382900350

[30] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.