Learning Physical Collaborative Robot Behaviors from Human Demonstrations

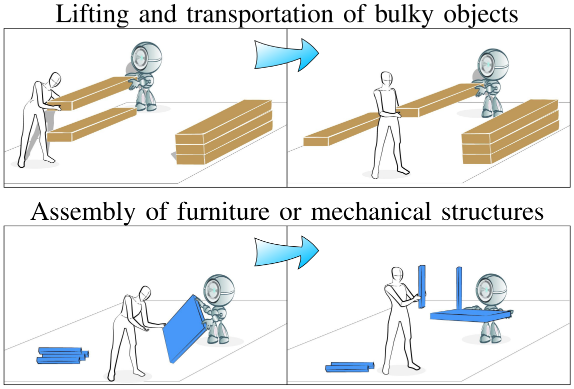
Leonel Rozo[1], *Member, IEEE,* Sylvain Calinon[2,1], Darwin G. Caldwell[1], Pablo Jiménez[3], and Carme Torras[3], *Senior member, IEEE,*

***Abstract*—Robots are becoming safe and smart enough to work alongside people not only on manufacturing production lines, but also in spaces such as houses, museums or hospitals. This can be significantly exploited in situations where a human needs the help of another person to perform a task, because a robot may take the role of the helper. In this sense, a human and the robotic assistant may cooperatively carry out a variety of tasks, therefore requiring the robot to communicate with the person, understand his/her needs and behave accordingly. To achieve this, we propose a framework for a user to teach a robot collaborative skills from demonstrations. We mainly focus on tasks involving physical contact with the user, where not only position, but also force sensing and compliance become highly relevant. Specifically, we present an approach that combines probabilistic learning, dynamical systems and stiffness estimation, to encode the robot behavior along the task. Our method allows a robot to learn not only trajectory following skills, but also impedance behaviors. To show the functionality and flexibility of our approach, two different testbeds are used: a transportation task and a collaborative table assembly.**

***Index Terms*—Robot learning, programming by demonstration, physical human-robot interaction, stiffness estimation.**

I. Introduction

RECENTLY, the fences in factories have started to disappear as a gentler breed of robots has entered the workplace, and new features have made even conventional robots more user-friendly and safer to be around. This shift will eventually alter the dynamics of labor in factories and workshops, allowing humans and robots to work together in efficient new ways. Some manufacturing tasks, such as the production of small mechanical components, require a robot to do the physical labor while a person performs quality-control inspections after each component is made. This requires the human and robot workers to operate side by side. For other jobs, like lifting an engine block so that it can be worked on, a user would want his/her robotic assistant to carry out the heavy lifting task (see Fig. 1). In such scenarios, the collaborative work can speed up production processes, improve manufacturing quality and reduce structural costs.


Lifting and transportation of bulky objects

Assembly of furniture or mechanical structures

Fig. 1. Illustration of human-robot cooperative tasks. In these kinds of scenarios, the robot needs to be easily and rapidly re-programmable, so that it can assist the user in a large variety of tasks, where different robot behaviors are required. Physical interaction between the participants occurs, and therefore the robot should be able to exploit the haptic information, as well as to show different compliance levels as needed.

On the other hand, service robots have as main premise to assist people in different environments such as houses, offices, hospitals, museums, etc. Here, some of the robot duties imply physical contact, for example, in hand-over tasks [3], or when a robot cooperatively transports an object with a human partner [2] (see Fig. 1). This physical interaction provokes a rich exchange of haptic information, and involves compliant robot movements. Additionally, the type of jobs carried out by the robot may frequently vary. Therefore, a robotic assistant requires to be easily and rapidly re-programmed several times according to specific needs. In this context, programming by demonstration (PbD) [4] emerges as a promising alternative solution allowing the natural and intuitive transfer of human knowledge about a task to a collaborative robot.

In this paper, we propose to use PbD to teach a robot different roles in human-robot collaboration (HRC) scenarios. Kinesthetic teaching[1] is used for the robot to learn, from demonstrations, an approximate model of the task along with its corresponding constraints. Specifically, our approach defines a virtual spring-damper system encapsulating the dynamics and constraints of the task, and in turn governing the robot behavior.[2] Such a system can act on different reference frames,

[1] Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova, Italy. name.surname@iit.it

[2] Idiap Research Institute, Rue Marconi 19, PO Box 592, CH-1920 Martigny, Switzerland. sylvain.calinon@idiap.ch

[3] Institut de Robòtica i Informàtica Industrial CSIC-UPC, C. Llorens i Artigas, 4-6, 08028 Barcelona, Spain. surname@iri.upc.edu

Initial results on the topic were published at the AAAI Conf. on Artificial Intelligence [1], and at the IEEE Intl. Symp. on Robot and Human Interactive Communication [2].

This work was supported by the European projects STIFF-FLOP (FP7-ICT-287728), DexROV (H2020-EU.3.2-635491), and the Spanish Ministry of Science and Innovation under project PAU+ DPI2011-27510.

The final version of this article is available at IEEE with DOI 10.1109/TRO.2016.2540623

[1] The term refers to the procedure where the user holds and moves the robot along the trajectories that need to be followed to accomplish the task, while the robot actively or passively compensates for the effect of gravity.

[2] We use the word *virtual* for clarifying that there is no real physical spring-damper system connected to the robot.

for instance, on coordinate systems representing the robot's base, a transported object, etc. We use a task-parametrized formulation of a Gaussian mixture model that allows us not only to encode the human demonstrations, but also to extract automatically varying constraints acting in different coordinate systems [5]. Moreover, by estimating the stiffness of the virtual system through convex optimization [6], the robot is able to interact with different compliance levels, thus extending its capabilities to impedance-based behaviors.

The contributions of the proposed framework are threefold: *(i)* exploitation of both position and force data in HRC; *(ii)* learning of compliance and position/force constraints; and *(iii)* modulation of the robot's behavior based on the user's actions as a result of the task-parametrized formulation. To show the flexibility of our approach, we test it in two different experiments. The first scenario consists of a collaborative transportation task, where the robot needs to simultaneously handle position and force constraints, while adapting its compliance level (Section IV-A). The second experimental setting considers the collaborative assembly of a wooden IKEA table, where the robot learns to adapt its compliance level based on position and haptic information (see Section IV-B).

The remainder of the paper is organized as follows: Section II reviews the related work. The interaction model, the learning algorithm and the stiffness estimation are described in Section III. Section IV presents the experimental settings and results. The pros and cons of the approach are discussed in Section V. Conclusions and future work are presented in Section VI.

II. Related Work

A. Control-based approaches

Human-robot collaboration has been investigated from the early nineties, when purely control-based approaches were dominant. Kosuge *et al.* [7], [8] proposed an admittance control based on the apparent mechanical impedance of an object manipulated by multiple robots and a human. The force applied by the human was transferred to the robot controllers, so that the human could command the motion of the object while the robots behaved as followers. Al-Jarrah and Zheng [9] introduced a two-level control scheme, where an admittance controller was driven by a higher level reflex control. The latter was triggered using a force-based threshold indicating that the robot acted as a load for the human. Force information was also exploited to estimate the human intention in cooperative tasks and change the robot control law accordingly [10]. The authors proposed to add the rate of change of the sensed force to the robot controller [11], while varying its damping as a function of the magnitude changes of the force.

Still based on admittance control, Bussy *et al.* [12] proposed to modify the robot control action according to the equilibrium trajectory hypothesis in a cooperative transportation setting. The equilibrium trajectory was computed from a desired dynamics of the object. The robot was then endowed with a velocity-dependent proactive behavior through a finite state machine whose states corresponded to hand-coded motion primitives, which were sequenced according to the user's intention. This approach is similar to the decomposition into non-holonomic motions represented by predefined virtual mechanisms [13], [14]. Agravante *et al.* [15] used a model of the task and visual servoing for defining the reference of an admittance controller (analogous to the equilibrium trajectory in [12]). The authors claimed that using exclusively force-based control might be insufficient in HRC, hence that visual information could improve the robot performance.

In contrast to [12], [14], our approach based on a virtual spring-damper system computes the equilibrium trajectory (which we also refer to as attractor) from a desired dynamics of the robot that is learned from kinesthetic demonstrations (see Section III-A). Here the robot motion depends not only on the interaction forces, but also on the task parameters influencing the collaborative behavior. It is worth noticing that the key limitation in the works reviewed above has been the need for a model of the task linked to an analysis of the possible robot movements, so that both the parameters and the structure of the controller can be designed accordingly. This significantly decreases the flexibility of these approaches in the sense that if a new robot skill is required or if an additional constraint needs to be considered, the controllers have to be redesigned. This limitation is here overcome by exploiting PbD.

B. Human performance-based approaches

Several works rely on human-human collaboration studies to assist in the design of the robot controllers. Ikeura *et al.* proposed to approximate human cooperation using variable admittance control (with zero stiffness). From data collected when two people jointly carried an object, the damping parameter was estimated according to the precision required by the task, either through least squares [11], or by minimizing a cost function that penalized high rates of change [16]. The approach was then improved by introducing stiffness into the controller, whose parameters were estimated from force and position data recorded when a single human completed the task following minimum jerk robot movements [17].

The minimum jerk model [18] has also been an inspiration for Maeda *et al.* [19] and Corteville *et al.* [20]. Such a model was used in [19] to estimate the human hand position in a human-robot carrying task. This estimation was then used as the reference for the robot controller. Similarly, in [20] an admittance-controlled robotic assistant set its reference based on the minimum jerk model. In order to improve the cooperation, the robot adjusted its speed profile based on Kalman estimations of the human motion. Human motion estimators were also adopted in [21], where a hidden Markov model (HMM) predicted the human intention based on haptic data and modified the robot controller reference accordingly.

Tsumugiwa *et al.* [22] used an admittance controller, where the damping varied according to the estimate of the human arm stiffness. Their approach assumed that a low velocity cooperative system remains stable if the robot's damping varies proportionally to the human stiffness. In [23] a robot controller emulating how humans compensate for interaction forces and instability was proposed. Composed of both feedforward and feedback terms, the controller iteratively learned to adapt to perturbations by minimizing motion errors and a metabolic cost, achieving variable impedance behaviors.

The works above suggest that both force sensory information and variable impedance are relevant for HRC, leading us to consider these aspects into our learning framework. The goal in most of these approaches is to *emulate* the way humans act in collaboration by shaping the parameters of a predefined controller using motion/force patterns sensed while a human-human pair carries out the task. The success of these methods mostly relies on how well the parameters of the robot controller are set to match the human collaborative behavior. Such approach may narrow the variety of collaborative behaviors that a robot could acquire, because characteristics like speed, power, precision, among others, may not be properly exploited. We propose to handle this by exploiting learning from demonstrations to transfer these characteristics.

C. Learning-based approaches

Evrard *et al.* [24] proposed the use of Gaussian mixture models (GMM) and Gaussian mixture regression (GMR) to respectively encode and reproduce robot collaborative behaviors. Leader and follower roles of a cooperative lifting task were demonstrated by teleoperation. The GMM encapsulated the robot motion and the sensed forces, while GMR generated the reference force during reproduction. Medina *et al.* [25] endowed a robot with a cognitive system providing segmentation, encoding and clustering capabilities for demonstrations of collaborative behavioral primitives. These were represented by a primitive graph and a primitive tree using HMMs that were incrementally updated during reproduction [26]. One of the main differences with respect to [24] was that the robot started behaving as a follower, but its role became more proactive as it acquired more knowledge about the task. Gribovskaya *et al.* [27] proposed a hybrid structure based on PbD and adaptive control. A model of the task was learned from demonstrations encoded by a GMM to generate feedforward control signals. Then, the impedance controller parameters were adapted as a function of kinematic and force errors generated in the task.

Dynamic movement primitives (DMP) have also been successfully used in HRC where interaction forces were considered. In [28], the learning problem was treated as that of finding an acceleration-based predictive reaction for coupled agents, in response to force signals indicating disagreements due to obstacle avoidance or different paths to follow. Gams *et al.* [29] argued that such adaptation should be done not only at acceleration, but also at velocity level, allowing for smoother interactions. Their approach learned coupled DMPs using iterative learning control that exploited the force feedback generated during several executions of the task. Note that our framework shares similarities with [24], [28], [29] in the sense that interaction forces are considered as additional variables influencing the collaborative robot behavior. Indeed, in our work, these forces affect not only the robot motion, but also its time-varying impedance.

Ben Amor *et al.* [30] proposed a probabilistic encoding of the DMP parameters that allowed for adaptation and correlation of the robot motion based on predictions of the human intention from partial observations. Their formulation used dynamic time warping for shaping the future robot actions according to the partner timing. Maeda *et al.* [31] extended this idea by modeling the collaborative interaction using the probabilistic motion primitives introduced in [32], where the correlation between the trajectories of both the human and the robot is exploited for carrying out coordinated tasks in which the robot action is entirely conditioned by the user motion. This approach has been recently applied to learning multiple collaborative tasks [33]. Our work differs from [31], [32] in that *(i)* our task-based parametrization allows the robot to automatically adapt to a broader range of situations, therefore augmenting its generalization capability; *(ii)* the proposed stiffness estimation does not necessarily depend on the demonstrations variability, and moreover, guarantees to find optimal gain matrices; and *(iii)* interaction forces were considered along the whole learning process.

Stiffness estimation: Variable impedance extends the robot learning capabilities beyond trajectory following tasks, where the robot is able to encode and reproduce more complex skills that involve, among others, contact with the environment and compliant bahaviors. In this context, several approaches have been proposed to estimate from collected data the stiffness and damping parameters to control robots. Erickson *et al.* [34] compared four different methods to estimate the robot impedance based on signal processing, adaptive control and recursive least squares. In [35], the authors estimated the nonlinear stiffness of robot joints with flexible transmissions by using dynamic residual signals along with least-squares and regressor-based techniques. From a different perspective, [36], [37] proposed to find a stiffness matrix using variability information extracted from human demonstrations in a PbD framework. The estimation was obtained from the inverse of the observed covariance encoded in a GMM [36], or from the conditional covariance in GMR [37]. Kronander and Billard [38] exploited the haptic information generated when the teacher shook or firmly grasped the robot for showing when high or low stiffness gains were required. Peternel *et al.* [39] used electromyography signals to teach a robot different compliance levels in a human-in-the-loop learning framework.

In [1] we proposed to encode an impedance-based collaborative behavior through a set of virtual springs that were activated according to the robot perceptions. Both the equilibrium point and the stiffness of the springs regulating robot motion were learned from demonstrations. We later focused on learning position and force constraints arising in a cooperative transportation task [2]. A time-driven trajectory following motion was encoded by a virtual attractor that allowed the robot to follow a desired path and to apply the force required to transport the load. The current paper takes inspiration from [2] to propose an interaction model that can encapsulate the robot dynamics for a large variety of collaborative skills. Such a model is exploited in this paper to shape the robot compliance by formulating the attractor's stiffness estimation as convex optimization, thus providing a more principled method than in [1] to estimate the stiffness gain matrices. The proposed task-parametrized encoding of the demonstrations permits the modulation of the robot actions as a function of task variables.

In summary, this paper builds on the insights obtained in [1], [2] to introduce a compact PbD model that is able: 1) to

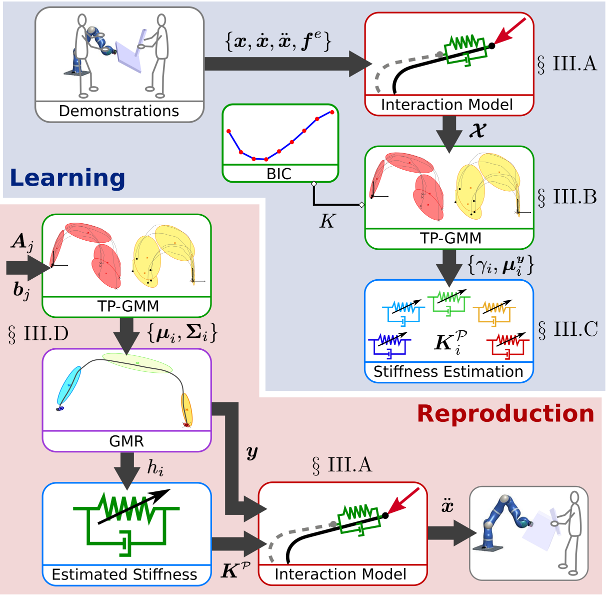

Demonstrations
$\{\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \boldsymbol{f}^e\}$
Interaction Model
§ III.A
$\boldsymbol{\mathcal{X}}$
BIC
Learning
K
TP-GMM
§ III.B
$\{\gamma_i, \boldsymbol{\mu}_i^{\boldsymbol{y}}\}$
$\boldsymbol{K}_i^{\mathcal{P}}$
Stiffness Estimation
§ III.C
$\boldsymbol{A}_j$
$\boldsymbol{b}_j$
TP-GMM
§ III.D
$\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}$
GMR
Reproduction
$\boldsymbol{y}$
§ III.A
h_i
$\ddot{\boldsymbol{x}}$
Estimated Stiffness
$\boldsymbol{K}^{\mathcal{P}}$
Interaction Model

Fig. 2. Diagram of the proposed framework. **Learning:** Demonstrations of the task are collected and then used to extract the desired attractor trajectories. Input vectors along with attractor trajectories are used to train a TP-GMM and to estimate stiffness matrices associated to every Gaussian component. **Reproduction:** Given a set of task parameters, a temporary GMM is generated and later used for reproducing the robot behavior by applying GMR.

learn different collaborative skills by exploiting both position and force information; 2) to optimally estimate the required time-varying compliance levels of the task; and 3) to modulate the robot's behavior based on both the user's actions and parameters of the task. A detailed description of the proposed approach is provided in the next section.

III. Proposed Approach

When a robot learns a collaborative role from teacher demonstrations, this needs a learning framework that encodes not only the raw data generated during the teaching process, but also an appropriate interaction model encapsulating the dynamics of the desired behavior. Moreover, some tasks may require the robot to manipulate objects, to pass them through specific locations in the robot workspace, and/or to apply specific forces to them. It is highly desirable that such task features are integrated in a joint model, which should be flexible enough, so that the robot can effortlessly be taught different collaborative skills as required. This section shows how we address these challenging aspects by proposing a novel PbD structure that combines a simple interaction model, probabilistic learning, and a stiffness estimation process for learning and reproducing collaborative tasks (see Fig. 2).

A. Interaction model

Let us represent the movement of the robot's end-effector as a single unit mass moving in Cartesian space under the effect of a control input $\boldsymbol{f}^m$ and interaction forces $\boldsymbol{f}^e$ as

$$\ddot{\boldsymbol{x}} = \boldsymbol{f}^m + \boldsymbol{f}^e, \tag{1}$$

where $\ddot{\boldsymbol{x}}$ is the acceleration of the mass, and the existence of the inverse dynamic model of the robot is assumed [40]. Hence, we formulate our problem as finding the motion control forces $\boldsymbol{f}^m$ to attain the desired task dynamics. To achieve this aim, we propose that the robot behavior during the interaction – at each time step t – is driven by a virtual spring-damper system. Specifically, the desired robot motion during interaction is given by

$$\ddot{\boldsymbol{x}}_t \;=\; \boldsymbol{K}_t^{\mathcal{P}}(\boldsymbol{y}_t - \boldsymbol{x}_t) - \boldsymbol{K}_t^{\mathcal{V}}\dot{\boldsymbol{x}}_t + \boldsymbol{f}_t^e, \tag{2}$$

where $\boldsymbol{K}_t^{\mathcal{P}}$, $\boldsymbol{K}_t^{\mathcal{V}}$ and $\boldsymbol{y}_t$ are the full stiffness matrix, the damping and the attractor point in the trajectory of the virtual system, respectively. By observing the evolution of the robot during the demonstrations of a task, with position $\boldsymbol{x}_t$, velocity $\dot{\boldsymbol{x}}_t$ and acceleration $\ddot{\boldsymbol{x}}_t$, and also the interaction forces $\boldsymbol{f}_t^e$ obtained by a sensor on the robot's end-effector, the evolution of the attractor $\boldsymbol{y}_t$ can be computed as

$$\boldsymbol{y}_t = \begin{bmatrix} \boldsymbol{I}, & (\boldsymbol{K}_t^{\mathcal{P}})^{-1}\boldsymbol{K}_t^{\mathcal{V}}, & (\boldsymbol{K}_t^{\mathcal{P}})^{-1}, & (\boldsymbol{K}_t^{\mathcal{P}})^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_t \\ \dot{\boldsymbol{x}}_t \\ \ddot{\boldsymbol{x}}_t \\ \boldsymbol{f}_t^e \end{bmatrix}, \tag{3}$$

corresponding to a simple linear transformation of the observed data. Note that (2) shares similarities with the DMPs formulation [41], where instead of encapsulating the robot motion by the attractor trajectory, a forcing term individually drives each variable of the robot movement.

This interaction model allows us to shape the robot behavior by varying both the stiffness and the attractor, based on the task requirements. Thus, we propose to tackle this problem from a robot learning perspective. Specifically, the variables $\boldsymbol{y}_t$ and $\boldsymbol{K}_t^{\mathcal{P}}$ will be learned from kinesthetic demonstrations provided by a human teacher who shows the robot its collaborative role. In the rest of this section, we explain how to extract these variables from examples of given tasks. Note that $\boldsymbol{K}_t^{\mathcal{V}}$ is not estimated here, but its values are pre-specified according to, for instance, a desired response of the linear system (2).

B. Learning

After having observed a set of demonstrations in some situations, we would like to generalize the skill to new situations. For instance, consider the scenario in which a human-robot dyad manipulates an object. The robot movements may largely depend on the initial and goal positions of the object, poses of obstacles populating the robot workspace, and additionally the robot may be required to react to human actions, that can be understood as position/orientation variations of specific parts of the human body. These variables influencing the robot behavior can be represented as reference frames, which we will refer to as *task parameters*. For generalization purposes, it is desirable for the robot to automatically adapt to new configurations of these parameters (e.g., unobserved positions and orientations of a manipulated object).

To address the aforementioned issue, we propose to probabilistically encode the demonstrations with a *task-parametrized* version of the Gaussian mixture model (TP-GMM) [42]. This model allows us to compactly capture the dependence of the robot motion on configuration changes of the task parameters. Formally, these parameters are represented as P coordinate systems, defined at time step t

by $\{\boldsymbol{b}_{t,j}, \boldsymbol{A}_{t,j}\}_{j=1}^{P}$, representing respectively the origin of the reference frame and a set of basis vectors $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots\}$ forming a transformation matrix $\boldsymbol{A} = [\boldsymbol{e}_1 \boldsymbol{e}_2 \cdots]$. Note that in this paper we focus on the special case in which the task parameters $\{\boldsymbol{b}_{t,j}, \boldsymbol{A}_{t,j}\}_{j=1}^{P}$ represent translations and rotations in the Cartesian space, but the model can be extended to any other affine transformation (including scaling and projections).

A demonstration $\boldsymbol{\xi} \in \mathbb{R}^{D\times T}$ is encoded in these different reference frames, forming a third order tensor dataset $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D\times T\times P}$, composed of P trajectory samples $\boldsymbol{X}^{(j)} \in \mathbb{R}^{D\times T}$ projected on P candidate frames, corresponding to matrices composed of D-dimensional observations at T time steps. The *model parameters* are defined by $\{\pi_i, \{\boldsymbol{\mu}_i^{(j)}, \boldsymbol{\Sigma}_i^{(j)}\}_{j=1}^{P}\}_{i=1}^{K}$, where π_i are the mixing coefficients, $\boldsymbol{\mu}_i^{(j)}$ and $\boldsymbol{\Sigma}_i^{(j)}$ are the center and covariance matrix of the i-th Gaussian component in frame j in a TP-GMM with K components.

Let us emphasize that the term *model parameters* refers here to the learned parameters of a model describing the movement or skill, whereas the external parameters representing the current situation (such as positions of objects or users) will be denoted *task parameters*. The latter are used as inputs to transform the learned *model parameters* in accordance to the situation. Learning of the model parameters is achieved by maximizing the log-likelihood under the constraint that the data in the different reference frames are generated from the same source, resulting in an Expectation Maximization (EM) process to iteratively update the model parameters until convergence.

E-step:

$$\gamma_{t,i} = \frac{\pi_i \prod\limits_{j=1}^{P} \mathcal{N}\big(\boldsymbol{X}_t^{(j)} |\, \boldsymbol{\mu}_i^{(j)}, \boldsymbol{\Sigma}_i^{(j)}\big)}{\sum_{k=1}^{K} \pi_k \prod\limits_{j=1}^{P} \mathcal{N}\big(\boldsymbol{X}_t^{(j)} |\, \boldsymbol{\mu}_k^{(j)}, \boldsymbol{\Sigma}_k^{(j)}\big)}. \tag{4}$$

M-step:

$$\pi_i = \frac{\sum_{t=1}^{T} \gamma_{t,i}}{T}, \quad \boldsymbol{\mu}_i^{(j)} = \frac{\sum_{t=1}^{T} \gamma_{t,i}\, \boldsymbol{X}_t^{(j)}}{\sum_{t=1}^{T} \gamma_{t,i}}, \tag{5}$$

$$\boldsymbol{\Sigma}_i^{(j)} = \frac{\sum_{t=1}^{T} \gamma_{t,i}\, (\boldsymbol{X}_t^{(j)} - \boldsymbol{\mu}_i^{(j)})(\boldsymbol{X}_t^{(j)} - \boldsymbol{\mu}_i^{(j)})^{\top}}{\sum_{t=1}^{T} \gamma_{t,i}}. \tag{6}$$

The model parameters are initialized with a *k-means* procedure (with five consecutive random clusters initialization) redefined through a similar process to that used for the above EM algorithm. The model selection is carried out by applying Bayesian Information Criterion (BIC) [43] to a set of TP-GMMs with different number of states. Note that other techniques such as Dirichlet processes for infinite Gaussian mixture models [44] can be alternatively used for computing the number of components in the model.

Notice that in a standard GMM, the role of EM is to estimate constant Gaussian parameters $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$. Here, EM is used to estimate task-parameterized model parameters $\boldsymbol{\mu}_i^{(j)}$ and $\boldsymbol{\Sigma}_i^{(j)}$ by incrementally modeling the local importance of the reference frames. In the proposed experiments, the overall learning process typically takes 1 to 5 seconds.

The learned model can further be used to reproduce movements in other situations (for new positions and orientations of reference frames). The model first retrieves a GMM at each time step t by computing a product of linearly transformed Gaussians

$$\mathcal{N}(\boldsymbol{\mu}_{t,i}, \boldsymbol{\Sigma}_{t,i}) \;\propto\; \prod_{j=1}^{P} \mathcal{N}\Big(\boldsymbol{A}_{t,j}\boldsymbol{\mu}_i^{(j)} + \boldsymbol{b}_{t,j},\; \boldsymbol{A}_{t,j}\boldsymbol{\Sigma}_i^{(j)}\boldsymbol{A}_{t,j}^{\top}\Big). \tag{7}$$

By using the product properties of multivariate normal distributions, the above distribution is evaluated with

$$\boldsymbol{\Sigma}_{t,i} = \Big(\sum_{j=1}^{P} \big(\boldsymbol{A}_{t,j}\boldsymbol{\Sigma}_i^{(j)}\boldsymbol{A}_{t,j}^{\top}\big)^{-1}\Big)^{-1}, \tag{8}$$

$$\boldsymbol{\mu}_{t,i} = \boldsymbol{\Sigma}_{t,i} \sum_{j=1}^{P} (\boldsymbol{A}_{t,j}\boldsymbol{\Sigma}_i^{(j)}\boldsymbol{A}_{t,j}^{\top})^{-1}(\boldsymbol{A}_{t,j}\boldsymbol{\mu}_i^{(j)} + \boldsymbol{b}_{t,j}). \tag{9}$$

C. *Stiffness estimation*

In our framework, we exploit the interaction model (2) that encapsulates the desired robot dynamics extracted from demonstrations to estimate a varying stiffness matrix. Such estimation can be carried out in a local fashion by taking advantage of the probabilistic encoding of the data. In other words, after having encoded the demonstrations with TP-GMM, we estimate local stiffness matrices $\boldsymbol{K}_i^{\mathcal{P}}$ associated to each Gaussian component i of the model. Each model component is locally encoding part of the desired robot dynamics, and thus each $\boldsymbol{K}_i^{\mathcal{P}}$ should fulfill some local dynamics.

The robot motion during interaction defined by (2) can be rewritten to express the local dynamics encoded by a Gaussian component i as

$$\gamma_{t,i}\,(\ddot{\boldsymbol{x}}_t + \boldsymbol{K}^{\mathcal{V}}\dot{\boldsymbol{x}}_t - \boldsymbol{f}_t^{e}) = \gamma_{t,i}\boldsymbol{K}_i^{\mathcal{P}}\,\big(\boldsymbol{\mu}_{t,i}^{\boldsymbol{y}} - \boldsymbol{x}_t\big), \tag{10}$$

with $\boldsymbol{K}_i^{\mathcal{P}} \in \boldsymbol{S}_{+}^{m}$ being the stiffness matrix of the component i, where $\boldsymbol{S}_{+}^{m}$ denotes the set of symmetric positive semi-definite $m \times m$ matrices. The weights $\gamma_{t,i}$ computed from (4) allow us to determine a region of validity in which the estimated stiffness matrix $\boldsymbol{K}_i^{\mathcal{P}}$ is optimal for the demonstrated dynamics. Observe that $\boldsymbol{\mu}_{t,i}^{\boldsymbol{y}}$ is the subvector of the center $\boldsymbol{\mu}_{t,i}$ spanning the attractor data dimensions. We treat here stiffness matrix estimation as a weighted norm approximation problem. In other words, we want to minimize the Euclidean norm of the residuals

$$\boldsymbol{r}_t\,(\boldsymbol{K}_i^{\mathcal{P}}) = \gamma_{t,i}\,(\boldsymbol{K}_i^{\mathcal{P}}\tilde{\boldsymbol{x}}_{t,i} - \boldsymbol{v}_t), \tag{11}$$

subject to the matrix inequality constraint introduced by the positive semidefiniteness of the stiffness matrix $\boldsymbol{K}_i^{\mathcal{P}}$. Note that $\tilde{\boldsymbol{x}}_{t,i} = (\boldsymbol{\mu}_{t,i}^{\boldsymbol{y}} - \boldsymbol{x}_t)$ and $\boldsymbol{v}_t = \ddot{\boldsymbol{x}}_t + \boldsymbol{K}^{\mathcal{V}}\dot{\boldsymbol{x}}_t - \boldsymbol{f}_t^{e}$. Here the weights $\gamma_{t,i}$ define the estimation neighborhood, so that only the datapoints belonging to the component i will be considered to estimate the matrix $\boldsymbol{K}_i^{\mathcal{P}}$. The above objective function and the positive semi-definite constraint on the stiffness matrix can be formulated as the convex optimization problem,

$$\underset{\boldsymbol{K}_i^{\mathcal{P}}}{\text{minimize}}\ \|\boldsymbol{r}_t\,(\boldsymbol{K}_i^{\mathcal{P}})\|_2\,, \quad \text{subject to} \quad \boldsymbol{K}_i^{\mathcal{P}} \succeq \boldsymbol{0}, \tag{12}$$

which is a semidefinite program (SDP). As a result of the convexity of both the objective function and the inequality constraint, standard techniques such as interior-point methods [6] can be used to efficiently solve this SDP problem. In this manner, optimal stiffness matrices are obtained to locally satisfy the dynamics observed during the demonstrations. This estimation process is carried out once for each Gaussian component i. Then, the demonstrations are discarded, which means that during reproduction, only the TP-GMM and associated stiffness matrices are needed.

D. Skill reproduction

With the temporary GMM computed in (7), the reproduction of a reference movement or behavior can be formalized as a regression problem [45]. *Gaussian mixture regression* (GMR) offers a simple solution to handle encoding, recognition, prediction and reproduction in robot learning [46]. In contrast to other regression methods such as *Locally Weighted Projection Regression* [47], or *Gaussian Process Regression* [48], GMR exploits the joint probability density function of the data modeled by TP-GMM for deriving the regression.[3]

Let us define the superscripts $\mathcal{I}$ and $\mathcal{O}$ as the sets of dimensions that span the input and output variables. At each iteration step t, the datapoint $\boldsymbol{\xi}_t$ can be decomposed into two subvectors $\boldsymbol{\xi}_t^{I}$ and $\boldsymbol{\xi}_t^{O}$ spanning the input and output variables, respectively. With this notation, a block decomposition of the datapoint $\boldsymbol{\xi}_t$, vectors $\boldsymbol{\mu}_i$ and matrices $\boldsymbol{\Sigma}_i$ can be written as

$$\boldsymbol{\xi}_t = \begin{bmatrix} \boldsymbol{\xi}_t^{I} \\ \boldsymbol{\xi}_t^{O} \end{bmatrix}, \quad \boldsymbol{\mu}_i = \begin{bmatrix} \boldsymbol{\mu}_i^{I} \\ \boldsymbol{\mu}_i^{O} \end{bmatrix}, \quad \boldsymbol{\Sigma}_i = \begin{bmatrix} \boldsymbol{\Sigma}_i^{I} \, \boldsymbol{\Sigma}_i^{IO} \\ \boldsymbol{\Sigma}_i^{OI} \boldsymbol{\Sigma}_i^{O} \end{bmatrix}. \tag{13}$$

The temporary GMM estimated in (7) encodes the joint distribution $\mathcal{P}(\boldsymbol{\xi}^{I}, \boldsymbol{\xi}^{O}) \sim \sum_{i=1}^{K} \pi_i \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ of the dataset $\boldsymbol{\xi}$. At each reproduction step t, $\mathcal{P}(\boldsymbol{\xi}_t^{O}|\boldsymbol{\xi}_t^{I})$ is computed as the conditional distribution

$$\begin{aligned} \mathcal{P}(\boldsymbol{\xi}_t^{O}|\boldsymbol{\xi}_t^{I}) &\sim \sum_{i=1}^{K} h_i(\boldsymbol{\xi}_t^{I})\, \mathcal{N}\Big(\hat{\boldsymbol{\mu}}_i^{O}(\boldsymbol{\xi}_t^{I}), \hat{\boldsymbol{\Sigma}}_i^{O}\Big), && (14) \\ \text{with} \quad \hat{\boldsymbol{\mu}}_i^{O}(\boldsymbol{\xi}_t^{I}) &= \boldsymbol{\mu}_i^{O} + \boldsymbol{\Sigma}_i^{OI} \boldsymbol{\Sigma}_i^{I}{}^{-1} (\boldsymbol{\xi}_t^{I} - \boldsymbol{\mu}_i^{I}), && (15) \\ \hat{\boldsymbol{\Sigma}}_i^{O} &= \boldsymbol{\Sigma}_i^{O} - \boldsymbol{\Sigma}_i^{OI} \boldsymbol{\Sigma}_i^{I}{}^{-1} \boldsymbol{\Sigma}_i^{IO}, && (16) \\ \text{and} \quad h_i(\boldsymbol{\xi}_t^{I}) &= \frac{\pi_i \mathcal{N}(\boldsymbol{\xi}_t^{I}|\, \boldsymbol{\mu}_i^{I}, \boldsymbol{\Sigma}_i^{I})}{\sum_{k}^{K} \pi_k \mathcal{N}(\boldsymbol{\xi}_t^{I}|\, \boldsymbol{\mu}_k^{I}, \boldsymbol{\Sigma}_k^{I})}. && (17) \end{aligned}$$

Note that according to our proposed interaction model, the robot behavior is driven by a virtual attractor $\boldsymbol{y}$ (see (3)), which corresponds to the output vector $\boldsymbol{\xi}^{O}$ of our regression model, while the input vector $\boldsymbol{\xi}^{I}$ will depend on the characteristics of the problem at hand, as shown in Section IV. Moreover, the stiffness matrices previously estimated define the stiffness of the virtual attractor as $\boldsymbol{K}_t^{\mathcal{P}} = \sum_{i=1}^{K} h_i(\boldsymbol{\xi}_t^{I}) \boldsymbol{K}_i^{\mathcal{P}}$, at each time step t. In the proposed experiments, the GMR computation at each time step took less than 1 millisecond. Fig. 2 shows an illustrative diagram of the proposed framework, while Table I summarizes our approach and its different stages.

TABLE I
SUMMARY OF THE PROPOSED APPROACH

```
1. Task demonstrations
   - Determine $P$ (number of frames or task parameters).
   - $\forall t \in \{1, \ldots, T\}$, collect $\boldsymbol{\xi}_t$ observed from the $P$
   different reference frames and form the third-order tensor $\boldsymbol{\mathcal{X}}$.
2. Model fitting (see Section III-B)
   - Determine $K$ (number of components of the model).
   - Use (4)-(6) to learn $\{\pi_i, \{\boldsymbol{\mu}_i^{(j)}, \boldsymbol{\Sigma}_i^{(j)}\}_{j=1}^{P}\}_{i=1}^{K}$ .
3. Stiffness estimation (see Section III-C)
   - Find $\boldsymbol{K}_i^{\mathcal{P}}$ for each Gaussian component $i$ through convex
   optimization using (11) and (12).
4. Reproduction (see Section III-D)
   - Set the input $\mathcal{I}$ and output $\mathcal{O}$ elements.
   for $t \leftarrow 1$ to $T$ (for each reproduction time step)
      - Collect/select $\boldsymbol{\xi}_t^{\mathcal{I}}$ and $\{\boldsymbol{b}_{t,j}, \boldsymbol{A}_{t,j}\}_{j=1}^{P}$.
      - Use (7) to estimate temporary GMM parameters
        $\{\boldsymbol{\mu}_{t,i}, \boldsymbol{\Sigma}_{t,i}\}_{i=1}^{K}$ modeling $\boldsymbol{\xi}_t^{\mathcal{I}}$ and $\boldsymbol{\xi}_t^{\mathcal{O}}$ as
        $\boldsymbol{\xi}_t^{\mathcal{I}}, \boldsymbol{\xi}_t^{\mathcal{O}} \sim \sum_{i=1}^{K} \pi_i \mathcal{N}(\boldsymbol{\mu}_{t,i}, \boldsymbol{\Sigma}_{t,i})$.
      - Use (14) to retrieve $\boldsymbol{\xi}_t^{\mathcal{O}}$ through GMR.
      - Compute the attractor's stiffness matrix as
        $\boldsymbol{K}_t^{\mathcal{P}} = \sum_{i=1}^{K} h_i(\boldsymbol{\xi}_t^{I}) \boldsymbol{K}_i^{\mathcal{P}}$.
      - Compute the reference acceleration $\ddot{\boldsymbol{x}}_t$ from (2).
   end
```

It is worth highlighting that the foregoing probabilistic formulation of the problem models not only local correlations among outputs, but also among inputs, and between inputs-outputs (see (13)), in contrast to standard DMPs in which a movement is considered as a set of univariate outputs synchronized by a decay term. In our application, the correlation information held by the joint inputs-outputs is crucial, not only to determine the relevance of the task parameters, but also to discover and re-use local sensorimotor patterns or synergies.

IV. EXPERIMENTS

This section introduces the two experimental settings that were used to test the performance of the proposed learning framework and to show its flexibility and generalization capability in HRC scenarios. For each task, we show and explain the obtained results for each module of the whole learning approach previously described.

A. Transportation Task

1) Description: the first task consists of teaching a robot to simultaneously handle the position and force constraints arising when an object is cooperatively manipulated or transported (see Fig. 3). At the beginning of the transportation task, the two participants simultaneously reach for the object. Once they make contact with the object, they start jointly transporting it along a given path to reach the target location. When the object gets to the final position, the human-human pair releases it and moves away from the object. Both the starting and goal object position/orientation may vary across repetitions. Here, the aim is to automatize such part of this collaborative task by replacing one of the human participants by a robot.

For this experiment, we used a torque-controlled 7-DoF WAM robot equipped with a 6-axis force/torque sensor. In the demonstration phase, the gravity-compensated robot is kinesthetically guided by the teacher while cooperatively achieving the task with the other user, as shown in Fig. 3. The teacher

[3] Note that GPR with Gaussian kernels also exploits the conditioning property of normal distributions, but in a different way. The input and output variables of the joint distribution in GPR represent respectively the new and previous observations, where a GP is used for each movement dimension. In contrast, the input and output variables of the joint distribution in GMR represent respectively the multivariate inputs driving the behavior and the multivariate movement.

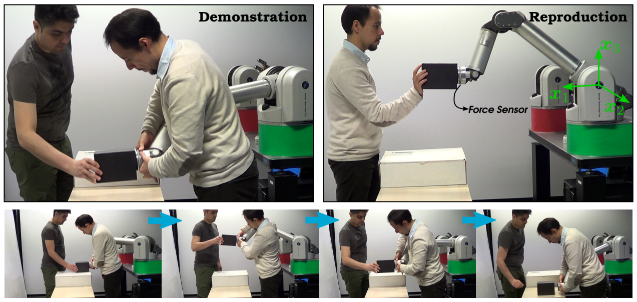


Fig. 3. Experimental setting of the human-robot transportation task. The *top row* shows the kinesthetic demonstration (left), and reproduction phase (right). A snapshots sequence of a demonstrations is shown in the *bottom row*.

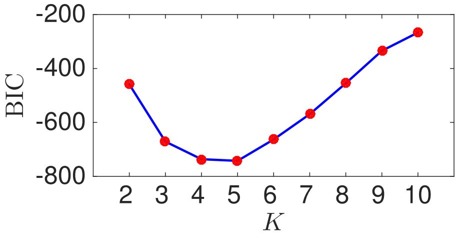


Fig. 4. Transportation task: Bayesian Information Criterion (BIC) for TP-GMMs with different number of states K.

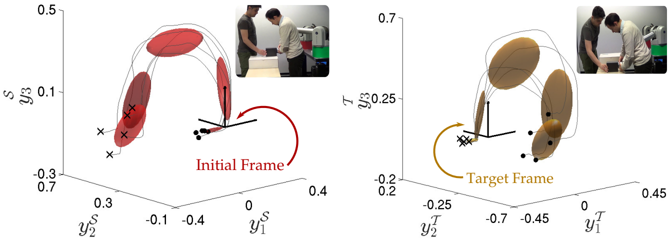


Fig. 5. Transportation task: Local models in the *initial* frame $\mathcal{S}$ (where the object is picked up), and in the *target* frame $\mathcal{T}$ (where the object is released). The gray lines depict the attractor trajectories projected on the corresponding reference frame. The ellipsoids represent the components of the local models. The black dots and crosses depict the beginning and the end of the demonstrations. The small and narrow ellipsoids reflect the high local invariance of the trajectories observed from the different frames. The big ellipsoids represent regions where precision is not required. Units are given in meters.

shows the robot both the path to be followed and the force pattern it should apply while transporting the object. Hence, the learned attractor will be directly associated to the desired force to be applied to the transported object, as computed in (3). In other words, the proposed formulation of the interaction process encodes the specific position and force requirements to be fulfilled during reproduction.

In this experiment three task parameters ($P = 3$) are considered, namely, the frames representing the initial $\mathcal{S}$ and target $\mathcal{T}$ locations of the object, and a third frame $\mathcal{I}$ attached to an object that is not relevant for the task, whose position randomly varies across demonstrations. The purpose of introducing the frame $\mathcal{I}$ is to exhibit how the TP-GMM handles task parameters that are irrelevant for the problem at hand. The task parameters are defined as

$$\begin{aligned}
\boldsymbol{A}_{t,1} &= \begin{bmatrix} 1 & \boldsymbol{0}_{1\times 3} \\ \boldsymbol{0}_{3\times 1} & \boldsymbol{R}^{\mathcal{S}} \end{bmatrix}, \boldsymbol{b}_{n,1} = \begin{bmatrix} 0 \\ \boldsymbol{x}_o^{\mathcal{S}} \end{bmatrix}, \\
\boldsymbol{A}_{t,2} &= \begin{bmatrix} 1 & \boldsymbol{0}_{1\times 3} \\ \boldsymbol{0}_{3\times 1} & \boldsymbol{R}^{\mathcal{T}} \end{bmatrix}, \boldsymbol{b}_{n,2} = \begin{bmatrix} 0 \\ \boldsymbol{x}_o^{\mathcal{T}} \end{bmatrix}, \text{and} \\
\boldsymbol{A}_{t,3} &= \begin{bmatrix} 1 & \boldsymbol{0}_{1\times 3} \\ \boldsymbol{0}_{3\times 1} & \boldsymbol{R}^{\mathcal{I}} \end{bmatrix}, \boldsymbol{b}_{n,3} = \begin{bmatrix} 0 \\ \boldsymbol{x}^{\mathcal{I}} \end{bmatrix}.
\end{aligned}$$

Here, $\boldsymbol{x}_o^{\mathcal{S}}$ and $\boldsymbol{x}_o^{\mathcal{T}}$ are the Cartesian positions where the object is picked up and then released. Similarly, $\boldsymbol{R}^{\mathcal{S}}$ and $\boldsymbol{R}^{\mathcal{T}}$ respectively represent the initial and final orientation of the object with orientation matrices. Finally, $\boldsymbol{x}^{\mathcal{I}}$ and $\boldsymbol{R}^{\mathcal{I}}$ are the position and orientation of the irrelevant object.[4] The datapoint $\boldsymbol{\xi}$ is defined as $\boldsymbol{\xi}^{I} = t$ and $\boldsymbol{\xi}^{O} = \boldsymbol{y}$, where t and $\boldsymbol{y}$ are time and the attractor's path (see (3)), respectively.

During reproduction of the task, the initial and target frames are given to the model in order to obtain the temporary GMM parameters using (7). Then, the robot and the user transport the object towards the target location. Here, for each time step t, the robot obtains a new attractor location (see (14)) along with an estimated stiffness matrix, which generate a new desired acceleration in the operational space of the robot. For simplicity of the experiment, the orientation of the robot end-effector was kept constant. However, the approach also supports tasks with variable end-effector orientations [49].

2) Results: Five examples of the collaborative behavior are given to the robot. The demonstrations are then used for training several TP-GMMs with different number of components. Then, we used a Bayesian Information Criterion (BIC) [43] to automatically select the TP-GMM providing the best trade-off between the data fitting and the number of Gaussian components. According to the values shown in Fig. 4, the model with five components has the lowest BIC value, thus offering the best compromise. Fig. 5 shows the resulting encoding of the attractor trajectories computed from (3) and observed from the perspective of the frames $\mathcal{S}$ and $\mathcal{T}$. Notice that the multiple demonstrations are locally consistent when the robot approaches the initial location of the object (frame $\mathcal{S}$), and when the manipulator moves away from it once it reached its target position (frame $\mathcal{T}$). This is reflected by the small and narrow ellipsoids in these parts of the task. The attractor trajectories observed from the frame $\mathcal{I}$ do not show any consistent or useful information regarding the task, and therefore they are not shown.

Handling irrelevant task parameters: Fig. 6 shows how the TP-GMM adapts to variations on the task parameters. The experiments consisted of varying the position of one frame at once while all others remain unchanged. we can see that when the frames $\mathcal{S}$ and $\mathcal{T}$ move, the resulting (temporary) model adapts its components accordingly. In contrast, when the irrelevant frame $\mathcal{I}$ moves, the model remains unaltered. This is explained by the fact that TP-GMM exploits the variability of the demonstrations observed from every frame to automatically discover which task parameters are relevant for the task. Such characteristic permits the selection of a common list of candidate task parameters without carrying out a thorough analysis about their importance for the skill or

[4] The pose of the objects was predefined in this experiment, but these can alternatively be obtained using an optical tracking system.

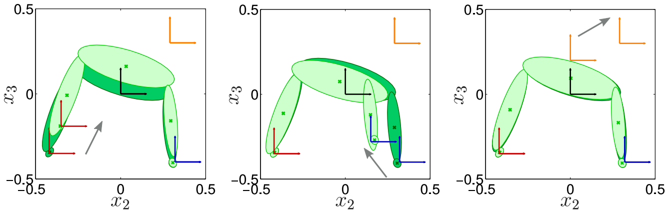


Fig. 6. Transportation task: The first, second, and third graphs respectively show the TP-GMM obtained when the *initial* frame $\mathcal{S}$ (red), the *target* frame $\mathcal{T}$ (blue), and the *irrelevant* frame $\mathcal{I}$ (orange) are moved on the plane x_2x_3. The model automatically discovers that the frame $\mathcal{I}$ is irrelevant for the task, and thus remains unchanged when this moves. Units are given in meters.

behavior to be learned.

Reproduction of the task: After learning, the obtained model was used to test the reproduction and generalization of the task on the real platform. Three types of tests were carried out to evaluate the performance. First, the human and robot cooperatively transported the load for new configurations of the task parameters while the force applied to the load was similar to those observed during the demonstrations. Fig. 7 (*top*) displays three successful reproductions under the aforementioned conditions where both the starting and target locations varied, and the robot adapted to the change of situation accordingly. Fig. 7 (*middle*) shows one of these reproduction attempts where the sensed force profile remains nearly constant throughout the whole reproduction. It is worth highlighting that the observed offset between the end-effector position and the attractor path allows the robot to apply the desired force to the object while transporting it. Fig. 8 shows the corresponding TP-GMM for the same reproduction, along with the stiffness profiles and the activation weights. Note that, in this case, time is the variable governing the influence of the model components on the stiffness estimation. Regarding the compliance level of the robot, notice that the robot behaves compliantly along x_1, while being stiffer along the axes x_2 and x_3. This behavior is reproduced both at the beginning and at the end of the execution of the task, when the robot does not allow high variations on the plane x_2x_3, guaranteeing that the object is picked up and released by following trajectories consistent with the demonstrations. As expected, the retrieved behavior is the opposite when the robot is cooperatively transporting the load in the middle of the path. In this part of the task, the robot behaves stiffly along x_1, while allowing deviations on the plane x_2x_3.

The second test consisted of applying a varying force to evaluate how the robot reacted to force variations not observed during learning. The human operator started the task pushing the object with a force higher than those taught during the demonstrations. Then, the applied force was significantly reduced, and finally it reached values similar to the demonstrations, as shown in Fig. 7 (*bottom*). As it can be observed, the robot could successfully adapt to these force variations. When the force along the x_1 axis was high, the robot allowed small deviations from the path, still ensuring that the position constraint remained within a feasible range determined by the observed variability in the demonstrations

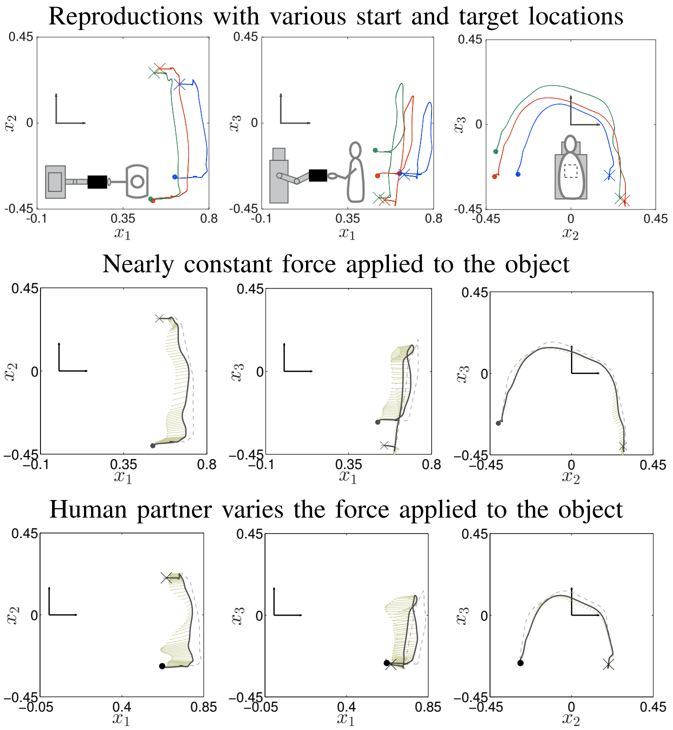


Fig. 7. Transportation task: The robot performs the task for new task-parameters configurations, as well as successfully adapts to variations on the force applied to the object. The black and light gray lines respectively represent the robot's trajectory and the attractor $\boldsymbol{y}$. The light brown arrows display the sensed force at the end-effector. The dots and crosses respectively display the start and end of the reproduction. Units are given in meters.

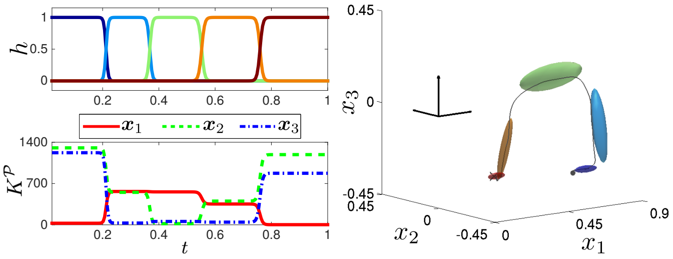
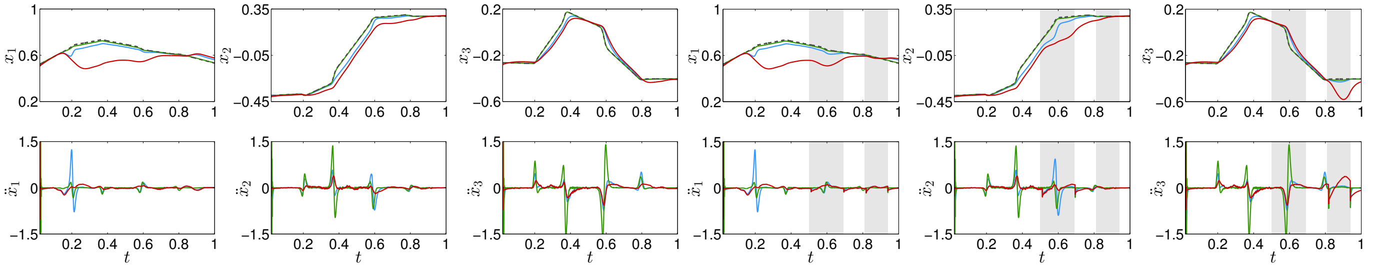


Fig. 8. Transportation task: (*left*) The top plot shows the influence of the learning model components along the reproduction, where the color matches that of the ellipsoids. The bottom plot displays the stiffness profile (Newton per meter) along the main axes of motion. (*right*) The resulting TP-GMM for the set of task parameters given at the beginning of the reproduction. The trajectory followed by the robot (in meters) is shown in solid gray line. Note that the model components determine the varying stiffness matrix used by the robot during reproduction.

and the impedance parameters. In contrast, when the force was very low (i.e., the human nearly losing contact with the object), the robot moved to try to compensate for the reduced force and prevent the object from being dropped. Note that despite the force variations, the robot was able to transport the object along a similar path in the other dimensions, by showing a collaborative behavior that is an appropriate compromise between force and position constraints, automatically extracted from the statistical representation of the demonstrations and the interaction model. A video of the experiment is available at `http://programming-by-demonstration.org/TRO2016/`



(a) No perturbation

(b) With perturbations

Fig. 9. Transportation task: Comparison of collaborative behavior reproductions where the stiffness matrix of the virtual attractor was learned with the proposed approach (blue line), or manually set to high and low constant stiffness (green and red lines, respectively). The equilibrium trajectory of the attractor is depicted by the black dashed line. The top row shows the trajectories followed by the end-effector for the foregoing cases. The bottom graphs display the desired accelerations computed from (2). Gray areas represent periods of time when a perturbation was applied to the robot. Units are given in meters.

Comparison with baseline executions: Notice that the aforementioned impedance adaptation capabilities are crucial for both successfully tracking the desired trajectories and interacting with the human user safely. In order to show this, we compared the robot reproduction using our approach with baseline executions. Specifically, for a given set of task parameters, we carried out the reproduction of the transportation task with (*i*) stiffness matrices computed as described in Section III-C, (*ii*) high stiffness (i.e., $\boldsymbol{K}_t^{\mathcal{P}} = 2000\boldsymbol{I}_{3\times3}$), and (*iii*) high compliance (i.e, $\boldsymbol{K}_t^{\mathcal{P}} = 100\boldsymbol{I}_{3\times1}$). Fig. 9 shows the results for these cases under two different conditions, namely when the human interacts with the robot similarly to the demonstrations, and when the robot faces perturbations during the execution of the task. Observe that when the robot uses a high compliance, it is not able to follow the desired trajectory once the user is in contact with the object (around 0.2 in the time axis). This means the robot does not compensate for the forces applied by the user, as required to satisfactorily transport the object. Moreover, when a perturbation occurs, the robot is significantly moved away from the desired trajectory, unable to respond accordingly even in the parts where a high precision trajectory tracking is required (see gray areas in Fig. 9b).

The use of high constant stiffness also showed some important drawbacks in this experiment. Firstly, despite the robot was able to precisely follow the attractor trajectory, it did not respond to interaction force variations during reproduction, because the high stiffness is only used to compensate position errors. Secondly, the desired end-effector accelerations are considerably higher than those computed from the proposed adaptive stiffness approach, as observed in Fig. 9. This may lead the controller to apply very high torque commands, thus inducing an unsafe human-robot interaction.

B. Table Assembly Task

1) Description: We consider a human-robot collaborative task where the robot's role is to hold a wooden table while the human screws the four legs to it, similarly as two persons would collaborate to carry out the same task (as illustratively shown in Fig. 10, middle row). Fig. 10 (*top*) presents an example of assembly instructions that can be found in "do it yourself" furniture catalogs. Here, two small tables require specific sequences of force and movement to get assembled.

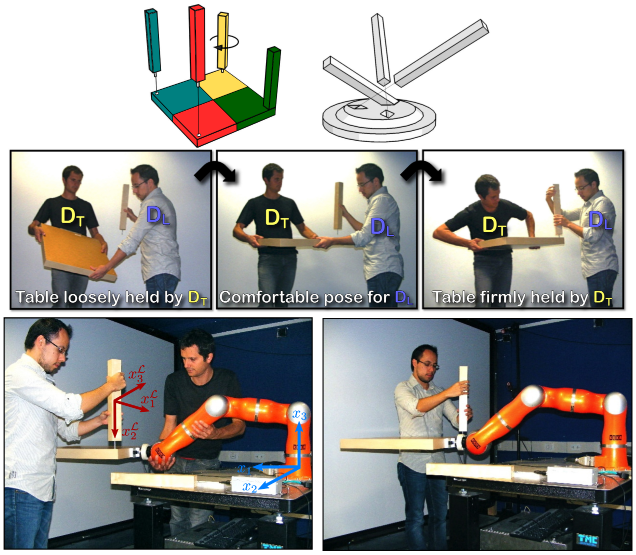


Fig. 10. *Top*: Examples of table assemblies characterized by different sequences, positions and orientations of components, with haptic and movement patterns that are specific for each item. *Middle*: Illustration of a table assembly task carried out by two persons. *Bottom*: demonstration (left) and reproduction (right) of the table assembly task in the experimental setup.

Learning such specificities is required for an efficient collaborative assembly. Instead of manually programming those specificities for each item, we would like the robot to extract them automatically from demonstrations provided by two persons collaborating to assemble the different parts of the table (see Fig. 10, bottom row). After learning, the task can be reproduced by a single user, with the robot interacting appropriately with respect to the preferences of the user and the specificities of the item being assembled. We thus do not need to provide the robot with information about the points of assembly, the different options, orientation of table legs, etc. The robot instead learns these specificities from demonstrations.

In the learning phase two persons perform the task, where one is kinesthetically guiding the robot to demonstrate the robot's role. The compliance behavior of the person holding the table changes to allow the other person to perform the corresponding sub-task more easily (see Fig. 10, bottom row). During reproduction, the robot replaces the person holding

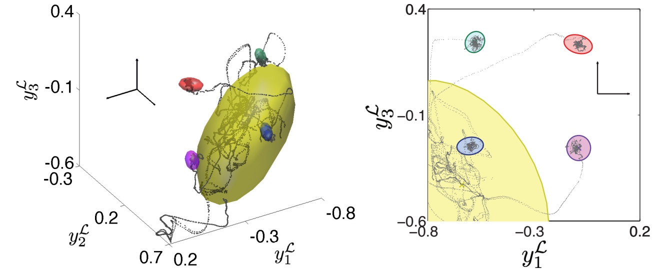


Fig. 11. Table assembly task: resulting TP-GMM in the leg's reference frame $\mathcal{L}$. The ellipsoids represent the Gaussian components of the learning model while the gray dots depict the trajectory followed by the attractor. The stiff phase of the four assembled legs was automatically encoded by the four small ellipsoids. The compliant behavior was encapsulated in the yellow component. Units are given in meters.

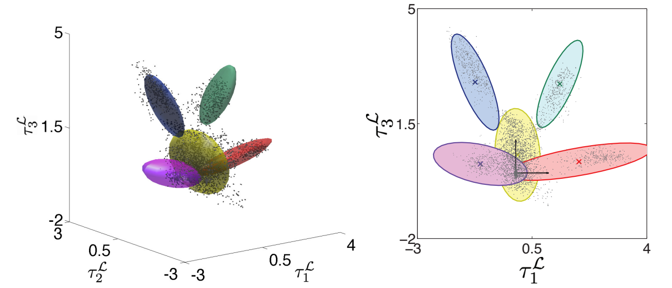
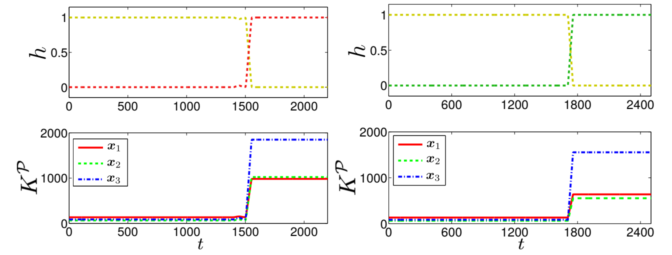


Fig. 12. Table assembly task: resulting TP-GMM projected onto the torques subspace of the leg's reference frame $\mathcal{L}$. The torques uniquely represent the four different assembly processes and the compliant behavior. Units are given in Newton meter.



(a) 1st leg assembly (b) 2nd leg assembly

Fig. 13. Table assembly task: the first row shows the influence of the model components during reproduction, where the color matches that of the ellipsoids in Fig. 11 and Fig. 12. The second row displays the corresponding stiffness profile (in Newton per meter) of the attractor along the main axes of motion.

the table by automatically estimating the attractor point and stiffness matrix to fulfill the required dynamics during the interaction. The collaborative scenario consists of screwing the legs at the four threads on the table. The user first requires the robot to behave compliantly, allowing him/her to move the table freely (compliant phase) until a comfortable pose is found for the work to be performed next. When the user grasps a leg and starts inserting it into the thread of the table, the robot adopts a stiff posture, holding the table to facilitate its partner's part of the task (stiff phase).

In this setting, we used a 7-DoF KUKA lightweight robot (LWR) [50], that is equipped with a six-axis force-torque sensor (ATI Mini45) attached between its wrist and the wooden table, measuring the interaction forces generated while moving the table and screwing the legs.

The position and orientation of the table legs are tracked with a marker-based NaturalPoint OptiTrack motion capture system, composed of 12 cameras working at a rate of 30 fps. The datapoint $\boldsymbol{\xi}$ is defined as $\boldsymbol{\xi}^{\mathcal{I}} = \boldsymbol{f}_e$ and $\boldsymbol{\xi}^{\mathcal{O}} = \boldsymbol{y}$, where $\boldsymbol{f}_e$ and $\boldsymbol{y}$ are the sensed forces/torques at the robot's end-effector, and the attractor path (see (3)), respectively. A transformation matrix and offset vector are computed to represent the first task parameter as

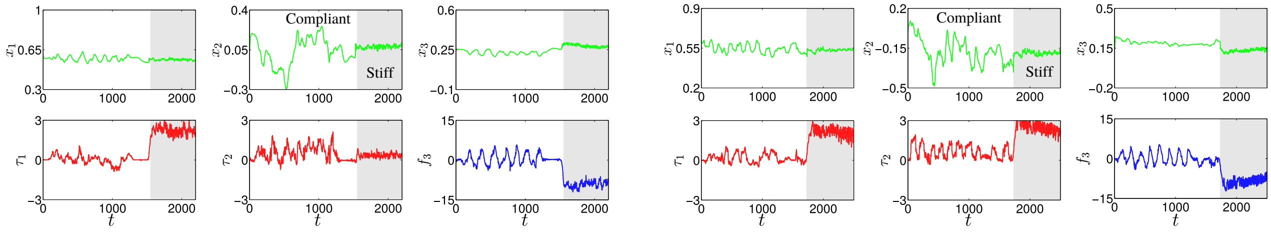
$$\boldsymbol{A}_{t,1} = \begin{bmatrix} \boldsymbol{R}_t^{\mathcal{L}} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{R}_t^{\mathcal{L}} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{R}_t^{\mathcal{L}} \end{bmatrix}, \boldsymbol{b}_{t,1} = \begin{bmatrix} \boldsymbol{0}_{6\times1} \\ \boldsymbol{x}_t^{\mathcal{L}} \end{bmatrix},$$

namely the leg configuration in the fixed robot frame, where $\boldsymbol{x}_t^{\mathcal{L}}$ and $\boldsymbol{R}_t^{\mathcal{L}}$ respectively stand for the Cartesian position as a vector and the orientation of the leg as a rotation matrix. The other task parameters $\{\boldsymbol{b}_{t,2}, \boldsymbol{A}_{t,2}\}$ define the robot's fixed frame of reference. Thus, two reference frames ($P = 2$) are used to describe the task parameters in this experiment.

It is worth highlighting that the combination of position and haptic information is fundamental for this task. If only position trajectories were used, the robot could not distinguish the phase during which the user aligns the screw with the thread. Here, the robot must regulate its stiffness in accordance with the sensed force pattern. If its behavior was based only on forces, the collaboration could fail because the robot could not distinguish which forces correspond to interactions with the user and which are produced by the assembly of the table. This can be problematic because these patterns might be similar in some situations. Both perception channels are thus needed to properly learn how the impedance behavior should be shaped.

2) Results: A model of five components ($K = 5$) was trained with twelve demonstrations (i.e., each leg was assembled three times to its corresponding thread with specific position and force patterns). The number of components K was found by choosing the TP-GMM with the lowest BIC value out of a set of models with different number of Gaussians. The resulting model automatically discovered four stiff components corresponding to the four screwing phases, with the remaining component representing the compliant phase. Each "stiff component" is characterized by the force-torque pattern and the relative position of the leg with respect to the robot's tool frame, which are different for each leg (see Fig. 11 and Fig. 12). The "compliant component" encodes the remaining points in the data space, i.e., the interaction forces-torques and the varying robot end-effector and leg positions.

Stiffness estimation: Once the model was learned, the stiffness was estimated as described in Section III-C. The proposed approach successfully estimated high stiffness matrices for the four Gaussian components representing the stiff phase of the task, and a stiffness matrix with very low values for the component encoding the compliant behavior. Fig. 13 shows both the components influence and stiffness profiles for a couple of reproductions. Notice that our approach allows the system to learn different stiffness values along the main axes of motion according to the task dynamics observed during the demonstrations. Specifically, in this experiment, the stiffness value along the x_3 axis is the highest. We attribute this to the fact that, during the stiff phase of some demonstrations,



(a) 1st leg assembly (b) 2nd leg assembly

Fig. 14. Table assembly task: reproductions for two different legs. The robot trajectory (in meters) is shown in the first row, where $\boldsymbol{x}_3$ is the vertical axis in the robot frame (see Fig. 10). The sensed vertical force (Newtons) and torques (Newton meters) around $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are shown in the second row. At the beginning of the reproductions, the robot can be freely moved by the user. However, when the robot senses a force/torque pattern associated with the assembly of one of the legs, its stiffness significantly increases and the robot behaves stiffly (gray areas in the plots), maintaining a nearly constant position to help the user during the screwing phase. Each leg generates a different force/torque pattern during the screwing, which determines the model component(s) driving the robot motion and compliance. The corresponding stiffness profiles are shown in Fig. 13.

the table was placed parallel to the x_1x_2 plane of the robot reference frame, and sometimes it was slightly moved along this plane. Not surprisingly, the teacher concentrated on keeping a stable pose along the vertical axis, which coincides with the direction of the force applied by the other person while screwing the legs into the table.

Reproduction of the task: We tested the reproduction and generalization capabilities of the system by carrying out the assembly process for all the legs. Fig. 14 shows the movement followed by the robot end-effector according to the force/torque profile sensed at its wrist. The stiffness of the robot is depicted in Fig. 13. At the beginning of the reproductions, the robot can be freely moved by the user. However, when the robot senses a force/torque pattern associated with the assembly of one of the legs (as supported by the relative position of table threads with respect to the end-effector), its stiffness significantly increases and the robot behaves stiffly, so that its partner can assemble the leg easily. Fig. 15 shows how the learning model varies as a function of the task parameters, where the distribution of the components in the dataspace changes according to both the leg position and orientation (see (7)), and the sensed force/torque pattern. Therefore, the model components and sensed forces governing the robot motion will also define the attractor stiffness.

We can observe in Fig. 13 that the compliant component (yellow dashed line) is influential during the first part of the reproduction, dominating the other components. After this, the robot becomes stiff, with specific patterns depending on which leg is being screwed. This means that not all the components influence the robot's impedance during the stiff phase, but mostly the Gaussian encoding the stiff behavior for the corresponding leg (as observed from the different colors representing the different stiff components), while the remaining activation weights stay close to zero. The proposed approach does not only learn when to change the compliance in a binary fashion, but also the manner in which to switch between the two behaviors. In this experiment, the sharp stiff/compliant switch is a specific characteristic of this collaborative task (fast but with continuous transitions between the two compliance levels), which is correctly learned and retrieved by the proposed approach.

Additionally, two situations that did not appear in the demonstrations were presented to the robot. First, the user tried to screw the leg at the center of the table, which means that the leg was placed at an incorrect position. In the second situation, the user positioned the leg in one of the table threads but the leg was wrongly oriented, making the screwing process unfeasible. In both cases, the robot behaved compliantly as expected, because neither corresponded to a correct screwing phase. A video of the experiment is available at `http://programming-by-demonstration.org/TRO2016/`.

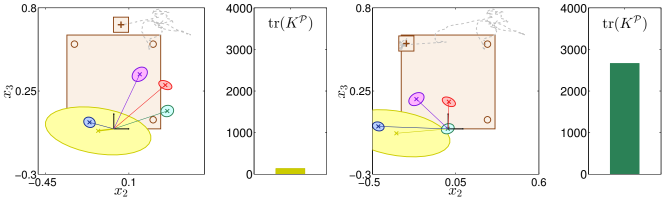


Fig. 15. Table assembly task: Assembly of the 2nd leg at different phases of the interaction. The first two graphs show an instant of the compliant phase, while the other two depict the time when the robot behaves stiffly. The projection of the Gaussian components in the tool's frame (as ellipses) is shown in the *first* and *third* graphs, while the bars show the trace of the resulting stiffness matrix (in Newton per meter). The centers and covariances in the TP-GMM vary as the leg is moved around the robot workspace. The gray dashed line represents the leg's trajectory (in meters). Both the table (with its 4 threads) and the legs are depicted in brown color, where the cross marks the current position of the leg.

Comparison with baseline executions: Automatic impedance adaptation is crucial in this task. Indeed, this collaborative assembly behavior cannot be performed by manually setting a specific stiffness gain matrix in advance. In the case of predefining a high stiffness behavior, the robot would simply stay at a fixed initial position, hence impeding the human partner from moving the table around and finding comfortable poses to screw each of the legs. On the contrary, if a very low stiffness matrix was given, the user could effortlessly move the table, but the screwing process would become infeasible.

V. Discussion

Interaction model

Our approach to represent the robot motion through a virtual attractor shares similarities with [12], [14], [15], where equilibrium points or reference trajectories of the controllers are set according to desired motion primitives. In this context, we would like to highlight two aspects. First, our attractor depends not only on the given input vector during the reproduction phase (see (14)), but also on the resulting TP-GMM for the given set of task parameters. If one or more of these parameters are related to the human partner actions, and are time-dependent, the attractor's trajectory will vary accordingly. This fact can be understood as the attractor depends on the user preferences or actions, which is highly relevant in scenarios where a robot interacts with a human. Second, the proposed definition of robot attractor (see (3)) requires a crude initial estimate of the stiffness matrix $\boldsymbol{K}^{\mathcal{P}}$, which is then refined through the estimation process presented in Section III-C. Despite such first estimate might affect the robot execution of the task, this allows us to predefine a specific response of the dynamical system, for instance, a critically damped behavior.

Selection of input variables and task parameters

The TP-GMM structure can handle a set of several candidate task parameters (or candidate projections), which will have a higher/lower influence on the robot behavior according to the variability of the task observed from each of them, as described in Section IV-A. In this sense, the teacher can specify the set of candidate frames that could potentially be relevant for the task, without an evaluation of their importance, since this will be automatically discovered by the algorithm. Note that the number of frames can be over-specified by the experimenter (e.g., by providing an exhaustive list), but it comes at the expense of requiring more demonstrations to obtain sufficient statistics to discard the frames that have no role in the task. In practice, the experimenter selects objects or locations in the robot kinematic chain that might be relevant for the task, and that remain valid for a wide range of manipulation tasks.

Regarding the input variable in the regression process, the selection procedure can be achieved in two ways. On the one hand, similar to the task parameters selection, a set of candidate input variables can be chosen, and after an automatic selection process based on how each candidate input affects the outputs, the most relevant inputs can be selected as those driving the regression [51] (this issue has strong links with the well-known *what to imitate?* problem). Alternatively, the teacher can take advantage of the his/her prior knowledge to tell the robot which inputs it needs to use for the task.

Stiffness estimation

The differences between the proposed stiffness estimation method and others used in PbD are worth emphasizing. Methods like [36], [37] estimate the stiffness of the robot by directly exploiting the variability observed in the demonstrations. These approaches have the disadvantage that they take only into account the positional information from the data, whose variability can sometimes be too weak if only a few number of demonstrations are considered. In these experiments, the users covered various portions of the workspace. In a more realistic scenario, the users might not be aware of this scaffolding teaching procedure, and a smaller number of less spread datapoints might be acquired. In such a situation, the kinematic variability information may not be sufficient to estimate the stiffness. In a similar context, the stochastic feedback controller obtained in [32] also depends on the trajectories variability, hence being prone to the aforementioned situation.

In contrast, we proposed in [1] a method that took into consideration haptic inputs in a two-step estimation process. A first estimate was obtained from weighted least squares (WLS) applied to the interaction model of the task. After, the algorithm found the nearest positive semi-definite matrix to such estimate. The estimation technique proposed in this paper is based both on a complete interaction model that considers the interaction forces (see (2) and (11)), and on the learning model that allows us to obtain local stiffness. The proposed formulation permits taking into account the positive semi-definite constraint directly into a convex optimization problem, which can be reliably and efficiently solved with widely available techniques to provide optimal estimates. As a drawback, the proposed estimation method requires several iterations to converge, while the other techniques provide algebraic closed-form solutions. Nevertheless, the first estimation obtained from WLS in [1] did not consider the positive semi-definite constraint, and therefore the resulting approximation may lie in $\boldsymbol{S}_{+}^{m}$ away from the optimal matrix minimizing (11), leading to suboptimal reproductions.

Handling more than one skill with TP-GMM

The proposed approach did not consider the problem of handling more than one collaborative behavior at the same time. The current formulation learns a different TP-GMM for each collaborative skill. Nevertheless, the probabilistic nature of the model can be exploited in further work to merge or sequence different skills with certain degree of similarity. Moreover, a higher level encoding may also be possible by training a mixture of task-parametrized models, similarly to [33]. All experiments reported in this paper were characterized by frames of reference constructed from translation and rotation operators. However, the affine transformation formulated in (7) does not impose specific restrictions on the task parameters, therefore allowing for a larger variety of relationships among the local models to be further explored, such as joint space and task space constraints, nullspace projections or force-based parameters conditioning collaborative behaviors with contact.

VI. Conclusions and Future Work

We have introduced a learning framework for HRC tasks in which the partners physically interact with a manipulated object. The framework exploits both position and force data, estimates time-varying compliance levels, encodes position and force constraints, and modulates the robot's behavior based on the user actions as a result of the task-parametrized formulation of the learning model. The flexibility and performance of the proposed methods were evaluated in two experiments with different robotic platforms, where the robot was able to learn either time-dependent or time-independent behaviors.

The experimental results showed that the approach successfully encoded the task dynamics while handling the different constraints observed in the demonstrations. Moreover, the resulting stiffness profiles obtained along the reproductions allowed the robot to fulfill the required dynamics of the different tasks. In this context, the local encoding of the data in the TP-GMM is crucial when different compliance levels are needed to successfully perform the task. This aspect becomes crucial when the robot physically interacts with a human, where safety requirements might also be included into the stiffness learning process. Generalization capabilities were evaluated for both new configurations of the task parameters and unobserved situations.

We plan to extend the estimation of the impedance parameters to the estimation of the damping matrix. We will also explore in which manner the variability of the demonstrations encapsulated in the covariance matrices of the model could be exploited to detect if the robot reaches an unexpected situation that is too far from the demonstrations (e.g., in case of failures). This could be exploited as a signal for the user to provide new demonstrations, possible refinements or corrections, which will require us to extend the current approach to incremental learning and estimation techniques. Additionally, we will study how interaction forces can be exploited to reshape the robot collaborative behaviors in an online manner. Further work will also investigate the automatic detection of redundant or irrelevant frames, in order to automatically determine in which manner the frames are coordinated with each other and locally contribute to the achievement of the task.

References

[1] L. Rozo, S. Calinon, D. G. Caldwell, P. Jiménez, and C. Torras, “Learning collaborative impedance-based robot behaviors,” in *AAAI*, 2013, pp. 1422–1428.

[2] L. Rozo, S. Calinon, and D. G. Caldwell, “Learning force and position constraints in human-robot cooperative transportation,” in *IEEE RO-MAN*, 2014, pp. 619–624.

[3] H. Admoni, A. Dragan, S. Srinivasa, and B. Scassellati, “Deliberate delays during robot-to-human handovers improve compliance with gaze communication,” in *ACM/IEEE HRI*, 2014, pp. 49–56.

[4] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, “Robot programming by demonstration,” in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 1371–1394.

[5] S. Calinon, Z. Li, T. Alizadeh, N. Tsagarakis, and D. Caldwell, “Statistical dynamical systems for skills acquisition in humanoids,” in *IEEE/RAS Humanoids*, 2012, pp. 323–329.

[6] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[7] K. Kosuge, H. Yoshida, and T. Fukuda, “Dynamic control for robot-human collaboration,” in *IEEE Intl. Workshop on Robot and Human Communication*, 1993, pp. 398–401.

[8] K. Kosuge and N. Kazamura, “Control of a robot handling an object in cooperation with a human,” in *IEEE Intl. Workshop on Robot and Human Communication*, 1997, pp. 142–147.

[9] O. Al-Jarrah and Y. Zheng, “Arm-manipulator coordination for load sharing using reflexive motion control,” in *IEEE ICRA*, 1997, pp. 2326–2331.

[10] V. Duchaine and C. Gosselin, “General model of human-robot cooperation using a novel velocity based variable impedance control,” in *IEEE EuroHaptics*, 2007, pp. 446–451.

[11] R. Ikeura and H. Inooka, “Variable impedance control of a robot for cooperation with a human,” in *IEEE ICRA*, 1995, pp. 3097–3102.

[12] A. Bussy, P. Gergondet, A. Kheddar, F. Keith, and A. Crosnier, “Proactive behavior of a humanoid robot in a haptic transportation task with a human partner,” in *IEEE RO-MAN*, 2012, pp. 962–967.

[13] J. Dumora, F. Geffard, C. Bidard, T. Brouillet, and P. Fraisse, “Experimental study on haptic communication of a human in a shared human-robot collaborative task,” in *IEEE/RSJ IROS*, 2012, pp. 5137–5144.

[14] J. Dumora, F. Geffard, C. Bidard, and P. Fraisse, “Towards a robotic partner for collaborative manipulation,” in *ACM/IEEE HRI - Workshop on Collaborative Manipulation*, 2013, pp. 1–6.

[15] D. Agravante, A. Cherubini, A. Bussy, and A. Kheddar, “Human-humanoid joint haptic table carrying task with height stabilization using vision,” in *IEEE/RSJ IROS*, 2013, pp. 4609–4614.

[16] R. Ikeura, T. Moriguchi, and K. Mizutani, “Optimal variable impedance control for a robot and its application to lifting an object with a human,” in *IEEE Intl. Workshop on Robot and Human Interactive Communication*, 2002, pp. 500–505.

[17] M. Rahman, R. Ikeura, and K. Mizutani, “Investigating the impedance characteristic of human arm for development of robots to cooperate with human operators,” in *IEEE SMC*, 1999, pp. 676–681.

[18] T. Flash and N. Hogan, “The coordination of arm movements: An experimentally confirmed mathematical model,” *Journal of Neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.

[19] Y. Maeda, T. Hara, and T. Arai, “Human-robot cooperative manipulation with motion estimation,” in *IEEE/RSJ IROS*, 2001, pp. 2240–2245.

[20] B. Corteville, E. Aertbelien, H. Bruyninckx, J. D. Schutter, and H. V. Brussel, “Human-inspired robot assistant for fast point-to-point movements,” in *IEEE ICRA*, 2007, pp. 3639–3644.

[21] Z. Wang, A. Peer, and M. Buss, “An HMM approach to realistic haptic human-robot interaction,” in *IEEE EuroHaptics*, 2009, pp. 374–379.

[22] T. Tsumugiwa, R. Yokogawa, and K. Hara, “Variable impedance control based on estimation of human arm stiffness for human-robot cooperative calligraphic task,” in *IEEE ICRA*, 2002, pp. 644–650.

[23] C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schäffer, and E. Burdet, “Human-like adaptation of force and impedance in stable and unstable interactions,” *IEEE T-RO*, vol. 27, no. 5, pp. 918–930, 2011.

[24] S. Calinon, P. Evrard, E. Gribovskaya, A. Billard, and A. Kheddar, “Learning collaborative manipulation tasks by demonstration using a haptic interface,” in *IEEE ICAR*, 2009, pp. 1–6.

[25] J. Medina, M. Lawitzky, A. Mortl, D. Lee, and S. Hirche, “An experience-driven robotic assistant acquiring human knowledge to improve haptic cooperation,” in *IEEE/RSJ IROS*, 2011, pp. 2416–2422.

[26] D. Kulić, W. Takano, and Y. Nakamura, “Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden Markov chains,” *IJRR*, vol. 27, no. 7, pp. 761–784, 2008.

[27] E. Gribovskaya, A. Kheddar, and A. Billard, “Motion learning and adaptive impedance for robot control during physical interaction with humans,” in *IEEE ICRA*, 2011, pp. 4326–4332.

[28] T. Kulvicius, M. Biehlc, M. J. Aein, M. Tamosiunaite, and F. Wörgötter, “Interaction learning for dynamic movement primitives used in cooperative robotic tasks,” *Rob. Auton. Syst.*, vol. 61, no. 12, pp. 1450–1459, 2013.

[29] A. Gams, B. Nemec, A. J. Ijspeert, and A. Ude, “Coupling movement primitives: Interaction with the environment and bimanual tasks,” *IEEE T-RO*, vol. 30, no. 4, pp. 816–830, 2014.

[30] H. B. Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, “Interaction primitives for human-robot cooperation tasks,” in *IEEE ICRA*, 2014, pp. 2831–2837.

[31] G. Maeda, M. Ewerton, R. Lioutikov, H. B. Amor, J. Peters, and G. Neumann, “Learning interaction for collaborative tasks with probabilistic movement primitives,” in *IEEE/RAS Humanoids*, 2014, pp. 527–534.

[32] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, “Probabilistic movement primitives,” in *NIPS*, 2013, pp. 2616–2624.

[33] M. Ewerton, G. Neumann, R. Lioutikov, H. B. Amor, J. Peters, and G. Maeda, “Learning multiple collaborative tasks with a mixture of interaction primitives,” in *IEEE ICRA*, 2015, pp. 1–6.

[34] D. Erickson, M. Weber, and I. Sharf, “Contact stiffness and damping estimation for robotic systems,” *IJRR*, vol. 22, no. 1, pp. 41–57, 2003.

[35] F. Flacco and A. D. Luca, “Residual-based stiffness estimation in robots with flexible transmissions,” in *IEEE ICRA*, 2011, pp. 5541–5547.

[36] S. Calinon, I. Sardellitti, and D. Caldwell, “Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies,” in *IEEE/RSJ IROS*, 2010, pp. 249–254.

[37] D. Lee and C. Ott, “Incremental kinesthetic teaching of motion primitves using the motion refinement tube,” *Auton. Robot.*, vol. 31, pp. 115–131, 2011.

[38] K. Kronander and A. Billard, “Learning compliant manipulation through kinesthetic and tactile human-robot interaction,” *IEEE TOH*, vol. 7, no. 3, pp. 367–380, 2014.

[39] L. Peternel, T. Petrič, E. Oztop, and J. Babič, “Teaching robots to cooperate with humans in dynamic manipulation tasks based on multi-modal human-in-the-loop approach,” *Auton. Robot.*, vol. 36, no. 1–2, pp. 123–136, 2014.

[40] O. Khatib, “A unified approach for motion and force control of robot manipulators: The operational space formulation,” *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[41] A. Ijspeert, J. N. and. H. Hoffmann, P. Pastor, and S. Schaal, “Dynamical movement primitives: Learning attractor models for motor behaviors,” *Neural Comput.*, vol. 25, no. 2, pp. 328–373, 2013.

[42] S. Calinon, D. Bruno, and D. G. Caldwell, “A task-parameterized probabilistic model with minimal intervention control,” in *IEEE ICRA*, Hong Kong, China, May-June 2014, pp. 3339–3344.

[43] K. Murphy, *Machine Learning - A Probabilistic Perspective*. MIT Press, 2012.

[44] V. Krüger, V. Tikhanoff, L. Natale, and G. Sandini, “Imitation learning of non-linear point-to-point robot motions using Dirichlet processes,” in *IEEE ICRA*, 2012, pp. 2029–2034.

[45] Z. Ghahramani and M. Jordan, “Supervised learning from incomplete data via EM approach,” in *NIPS*, 1994, pp. 120–127.

[46] S. Calinon, F. Guenter, and A. Billard, “On learning, representing and generalizing a task in a humanoid robot,” *IEEE T-SMC, Part B*, vol. 37, no. 2, pp. 286–298, 2007.

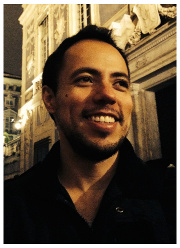
[47] S. Vijayakumar, A. D’Souza, and S. Schaal, “Incremental online learning in high dimensions,” *Neural Comput.*, vol. 12, no. 11, pp. 2602–2634, 2005.

[48] D. Nguyen-Tuong, M. Seeger, and J. Peters, “Model learning with local Gaussian process regression,” *Advanced Robotics*, vol. 23, no. 15, pp. 2015–2034, 2009.

[49] J. Silvério, L. Rozo, S. Calinon, and D. G. Caldwell, “Learning full end-effector poses from demonstrations using task-parameterized dynamical systems,” in *IEEE/RSJ IROS*, 2015, pp. 464–470.

[50] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger, “The DLR lightweight robot - design and control concepts for robots in human environments,” *Industrial Robot: An Intl. Journal*, vol. 34, no. 5, pp. 376–385, 2007.

[51] L. Rozo, P. Jiménez, and C. Torras, “A robot learning from demonstration framework to perform force-based manipulation tasks,” *Intel. Serv. Robot., Special Issue on Artificial Intelligence Techniques for Robotics, Part 2*, vol. 6, no. 1, pp. 33–51, 2013.



Leonel Rozo is a senior postdoctoral researcher at the Department of Advanced Robotics (ADVR), Istituto Italiano di Tecnologia since 2013. He received his B.Sc on Mechatronics Engineering from the ”Nueva Granada” Military University (Colombia, 2005), his M.Sc in Automatic Control and Robotics (2007), and Ph.D in Robotics (2013) from the Polytechnical University of Catalonia (Barcelona, Spain). From 2007 to 2012 he carried out his research on force-based manipulation tasks learning at the Institut de Robòtica i Informàtica Industrial (CSIC-UPC). His research interests cover robot programming by demonstration, physical human-robot interaction, machine learning and optimal control for robotics.

Sylvain Calinon is a researcher at the Idiap Research Institute, heading the Robot Learning & Interaction Group. He is also a Lecturer at the École Polytechnique Fédérale de Lausanne and an External Collaborator at the Department of Advanced Robotics (ADVR), Italian Institute of Technology (IIT). From 2009 to 2014, he was a Team Leader at IIT. From 2007 to 2009, he was a Postdoc at EPFL. He holds a PhD from EPFL (2007), awarded by Robotdalen, ABB and EPFL-Press Awards. He is the author of about 70 publications and a book in the field of robot learning by imitation and human-robot interaction, with recognition including Best Paper Award at Ro-Man’2007 and Best Paper Award Finalist at ICIRA’2015, IROS’2013 and Humanoids’2009. He currently serves in the Organizing Committee of IROS’2016 and as Associate Editor in IEEE Robotics and Automation Letters, Springer Intelligent Service Robotics, Frontiers in Robotics and AI, and the International Journal of Advanced Robotic Systems. Personal webpage: `http://calinon.ch`

Darwin G. Caldwell is a founding Director at the Italian Institute of Technology in Genoa, Italy, and a Honorary Professor at the Universities of Sheffield, Manchester, Bangor, Kings College, London and Tianjin University China. His research interests include innovative actuators, humanoid and quadrupedal robotics and locomotion (iCub, cCub, HyQ and COMAN), haptic feedback, force augmentation exoskeletons, dexterous manipulators, biomimetic systems, rehabilitation and surgical robotics, telepresence and teleoperation procedures. He is the author or co-author of over 450 academic papers, and 17 patents and has received awards and nominations from several international journals and conference including; IMechE Best Paper Award 2009, Ind. Robot Journal 2010, ICRA (2007), IROS (2007, 2012, 2013), ICAR (2003), Humanoids (2008, 2012), CASE (2008), ICMA (2011), Robio (2013) IFAC IAV, MMVR (2011), ACHI (2010), WorldHaptics (2007) and Virtual Concepts (2006). He is Editor for Frontiers in Robotics and AI, secretary of the IEEE/ASME Trans. on Mechatronics and on the editorial board of the International Journal of Social Robotics and Industrial Robot.

Pablo Jiménez is an Associate Researcher at IRI-CSIC since 2002. He received the Ph.D. degree in Robotics and Automation in 1998 from the Technical University of Catalonia (UPC), with a work on orientation-based geometric pruning for collision detection. From 1993 to 1997 he worked first at the Institut de Cibernètica (Barcelona) and later at IRI with different contracts and scholarships. In 1999-2000 he made a 9-month research stage at the Institut fr Produktionsanlagen und Konstruktionstechnik (IPK, Fraunhofer Gesellschaft - TU Berlin). His research is on Robot Planning and Learning. His principal contributions in the past have been on basic collision detection algorithms for robot motion planning, graph search procedures in cyclic AND/OR graphs, and currently on force-based skills learning from demonstration. His research interests include also high-level or symbolic task learning and planning.

Carme Torras is Research Professor at the Spanish Scientific Research Council (CSIC). She received M.Sc. degrees in Mathematics and Computer Science from the Universitat de Barcelona and the University of Massachusetts, Amherst, respectively, and a Ph.D. degree in Computer Science from the Technical University of Catalonia (UPC). Prof. Torras has published five books and about two hundred papers in the areas of robot kinematics, neurocomputing, machine learning, computer vision and geometric reasoning. She has been local project leader of several European projects, among which the FP6 IP project Perception, Action and COgnition through Learning of Object-Action Complexes (PACO-PLUS), and the FP7 STREP projects GARdeNIng with a Cognitive System (GARNICS) and Intelligent observation and execution of Actions and manipulations (IntellAct). She was awarded the Narcís Monturiol Medal of the Generalitat de Catalunya in 2000, and she became ECCAI Fellow in 2007, member of Academia Europaea in 2010, and member of the Royal Academy of Sciences and Arts of Barcelona in 2013.