Learning bimanual end-effector poses from demonstrations using task-parameterized dynamical systems

João Silvério[1], Leonel Rozo[1], Sylvain Calinon[1,2], Darwin G. Caldwell[1]

***Abstract*— Very often, when addressing the problem of human-robot skill transfer in task space, only the Cartesian position of the end-effector is encoded by the learning algorithms, instead of the full pose. However, orientation is just as important as position, if not more, when it comes to successfully performing a manipulation task. In this paper, we present a framework that allows robots to learn the full poses of their end-effectors in a task-parameterized manner. Our approach permits the encoding of complex skills, such as those found in bimanual manipulation scenarios, where the generalized coordination patterns between end-effectors (i.e. position and orientation patterns) need to be considered. The proposed framework combines a dynamical systems formulation of the demonstrated trajectories, both in $\mathbb{R}^3$ and $SO(3)$, and task-parameterized probabilistic models that build local task representations in both spaces, based on which it is possible to extract the relevant features of the demonstrated skill. We validate our approach with an experiment in which two 7-DoF WAM robots learn to perform a bimanual sweeping task.**

I. Introduction

Programming by demonstration (PbD) has gained high interest in Robotics as an intuitive and user-friendly means to transfer skills from humans to robots [1], [2]. Several learning algorithms have been proposed to encode human demonstrations in a compact way, while encapsulating the relevant information of the task at hand [3], [4], [5]. Most of the proposed methods have mainly focused on learning skills where the end-effector position in Cartesian space is exclusively considered, while orientation is ignored or kept constant. This can be attributed to the fact that, unlike position, that can be uniquely represented by a vector in $\mathbb{R}^3$, minimal parameterizations of $SO(3)$ contain singularities, which make robot learning a more difficult problem, especially when the movement to be learned requires to be adapted to external task parameters describing the current context, situation or configuration of the workspace. However, if the end-effector orientation is considered, more complex tasks can be learned, therefore increasing the dexterity of robots.

Bimanual robotic manipulation (Fig. 1) is a good example of a scenario where complex movements at the level of the end-effectors are needed for performing successfully [6]. In this context, learning generalized motions (i.e., position and orientation) is crucial for achieving dexterous and autonomous dual-arm skills. In a PbD framework, we posit

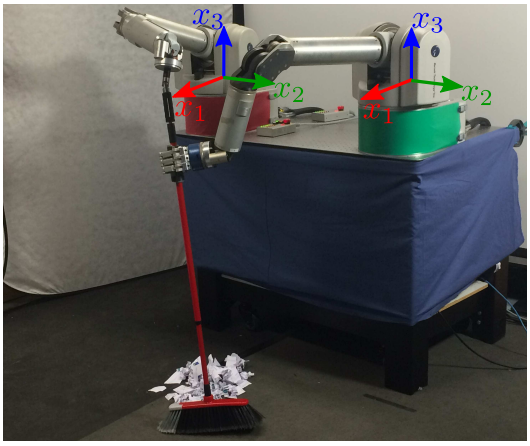

Fig. 1. Two WAM 7-DoF robots performing a bimanual sweeping task.

that a bimanual manipulation skill is successfully learned when a robot can: 1) extract the pose constraints between the two hands, 2) maintain the learned poses when perturbed, and 3) know when the formation needs to be maintained (because a bimanual robot may also be performing two separate uni-manual tasks at the same time). In this paper, we present a framework for learning generalized end-effector motions by encoding the desired dynamics of a demonstrated skill through a set of virtual mass-spring-damper systems. The approach is also able to automatically discover the constraints of the movement by exploiting the task-parameterized formulation of a Gaussian mixture model (TP-GMM) [7]. The foregoing aspects allow robots to build compact models of bimanual tasks, where full end-effector poses can be modulated by external task parameters in a probabilistic manner, leading to increased generalization capabilities. Formation constraints between end-effectors are thus automatically identified from a set of human demonstrations, which is highly relevant in joint object manipulation [6].

The contributions of this paper are twofold: 1) a method that permits including the end-effector orientation in the formulation of TP-GMM, allowing the robot to learn complete end-effector poses; 2) a quaternion-based dynamical systems formulation which makes it possible to encode the dynamics of the task in $SO(3)$ through a virtual attractor, and therefore to select the desired impedance of the orientation controller.

The remainder of the paper is organized as follows: Section II covers related work in both bimanual skill transfer and learning of orientations, Section III describes the proposed framework for learning position and orientation constraints from demonstrations, and Section IV introduces the

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

[2]Idiap Research Institute, Rue Marconi 19, CH-1920 Martigny, Switzerland.

This work was supported by the STIFF-FLOP European project (FP7-ICT-287728) and by the DexROV European project (H2020-EU.3.2-635491).

generalized statistical dynamical systems approach. Section V presents the experimental setup used for validating the proposed approach, as well as the corresponding results, and, finally, conclusions and future work are drawn in Section VI.

II. Related Work

The transfer of bimanual skills to robots is a growing topic of research in robot learning. As the number of humanoid robots available in the market grows, there is an increasing need to exploit their two hands, in order to increase the repertoire of tasks they can perform [6].

Gams *et al.* [8] extend dynamic movement primitives (DMPs) [4] with a method that modulates their velocity using a force term computed based on a virtual spring connecting the end-effectors. This spring has an equilibrium point that is the desired distance between the end-effectors, therefore forcing them to maintain a desired formation. Umlauft *et al.* [9] define a cooperation term inspired by artificial potential fields, that is a function of the distance between the end-effectors. This term is then incorporated as feedback in DMPs allowing the robots to keep a demonstrated formation. Even though these approaches successfully encode demonstrated trajectories, they present two main limitations: 1) the neglect of orientation in the pose of the end-effectors, and 2) the absence of statistical information about the task to be reproduced, which makes it impossible to abstract the range of possible variations of the skill. Other works like [10], focus on the design of optimal controllers with the purpose of minimizing internal forces (when cooperatively handling objects) and deviations from desired formations. In this case, the desired target configurations were pre-programmed. Nevertheless, such approaches have the potential to be combined with PbD, which would allow users to demonstrate the desired formations to the dual-arm setup.

Concerning statistical modeling and regression of orientation data, Lang *et al.* [11] propose a method based on Gaussian processes (GP) to model orientations as quaternions. The approach, however, is not applicable in robot learning scenarios where predicted end-effector orientations need to adapt to task variations. We showed in [12] that treating task adaptation as standard regression (with GP implementation as example) has limited generalization capabilities. We also showed that task-parameterized modeling approaches, on the other hand, introduce a structure based on affine transformations to the task adaptation problem that results in better generalization. This is achieved at the expense of being less generic (due to the introduced structure), but still covering a wide range of problems in robotics. It is thus convenient that, when using task-parameterized models to achieve task adaptation at the level of orientation, we choose a parameterization of orientation that abides by such structure, i.e. that allows for composing orientations using an affine operation. Note that the axis-angle representation is not appropriate for composing orientations, as pointed out in [11]. Hence, in our approach, we have chosen to use quaternions.

Pastor *et al.* [13] and Ude *et al.* [14] extended the DMP framework to allow the encoding of orientations from demonstrations using quaternions. The latter work improves on the former by providing a formulation that ensures faster convergence to attractor points. We take inspiration from [14] to develop a quaternion-based statistical dynamical systems formulation of orientation trajectories (Section IV).

We have proposed, in [5], a method based on task-parameterized Gaussian mixture models (TP-GMM) to transfer bimanual skills to a humanoid robot. TP-GMM (Section III) builds representations of demonstrations in a given set of candidate frames, which can represent any coordinate system or rigid body in the scene, including objects and robot body parts. The local information about the variability and coordination of a movement, encoded in each frame, is used to automatically coordinate and use the most relevant frames for the execution of the task, a feature that is not offered by methods such as DMPs. In addition, TP-GMM increases the generalization capabilities of standard GMM in that, by encoding information locally, a movement can be regenerated online even if the frames are moving during reproduction. In bimanual manipulation scenarios, we select the frame of each end-effector as potentially relevant frames, with the objective of encoding bimanual coordination patterns. With respect to [5], the work we propose in this paper is innovative in that: 1) it builds local representations not only of end-effector position but also of orientation; 2) it uses a single TP-GMM for both arms, as opposed to one per arm, becoming a more compact model formulation; 3) the dynamics of the demonstrated skill in $SO(3)$ are taken into account through the computation of a quaternion-based virtual attractor. In [7], a pan-tilt parameterization of orientation was employed in TP-GMM, for a peg-in-hole experiment. However, this parameterization is not appropriate for composing orientations. In that paper, only the initial and final orientations of the end-effector were required to match those of the start and end holes, without orientation offsets between frames and end-effector. In this paper we go beyond this limitation by enabling the encoding of full orientation trajectories. The present work aims to fill a gap in PbD, which is that of transferring bimanual skills to robots while considering task variations with full end-effector poses.

III. Learning full pose constraints using a task-parameterized Gaussian mixture model

In this paper we make use of TP-GMM to encode both positions and orientations of end-effectors in multiple reference frames described by a set of task parameters. Formally, the task parameters correspond to P coordinate systems, defined at time step n by $\{\boldsymbol{b}_{n,j}, \boldsymbol{A}_{n,j}\}_{j=1}^{P}$, representing respectively the origin of the frame and a set of basis vectors $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots\}$ forming a transformation matrix $\boldsymbol{A}=[\boldsymbol{e}_1\boldsymbol{e}_2\cdots]$.

A movement $\boldsymbol{\xi} \in \mathbb{R}^{D\times N}$ is demonstrated in a global frame and projected onto these different viewpoints, forming P trajectory samples $\boldsymbol{X}^{(j)} \in \mathbb{R}^{D\times N}$. The projection is implemented by means of a linear transformation using the task parameters as $\boldsymbol{X}_n^{(j)} = \boldsymbol{A}_{n,j}^{-1}(\boldsymbol{\xi}_n - \boldsymbol{b}_{n,j})$, for any given

time step n. Every $\boldsymbol{X}^{(j)}$ corresponds to a matrix composed of D-dimensional observations during N time steps. The parameters of the model with K components 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 and $\boldsymbol{\mu}_i^{(j)}$, $\boldsymbol{\Sigma}_i^{(j)}$ are the center and covariance matrix of the i-th Gaussian component at frame j.

Learning of the parameters is achieved by maximizing the log-likelihood under the constraint that the data in the different frames are generated from the same source, resulting in an Expectation-Maximization (EM) process to iteratively update the model parameters until convergence (see details in [7]).

A. Task Parameters for learning orientation constraints

Parameterizations of $SO(3)$ such as Euler angles (minimal, 3-dimensional) and axis-angles (non-minimal, 4-dimensional) contain representation singularities [15]. In addition, composing orientations using these parameterizations is not straightforward (adding two pairs of Euler angles or axis-angles does not result in the same orientation as the one that results from applying the two rotations consecutively). Alternatively, quaternions and rotation matrices provide a non-minimal, singularity-free, parameterization of $SO(3)$. The (non-commutative) product between two quaternions or two rotation matrices performs the rotation operation, providing an appropriate way of composing orientations. It is in our interest to keep the model dimensionality low, since the cost of estimating a GMM grows quadratically with the number of dimensions. We therefore opt for using unit quaternions since they represent orientations using only 4 parameters against 9 of rotation matrices (which, additionally, have orthonormality constraints).

A unit quaternion, $\boldsymbol{\epsilon} \in \mathbb{S}^3$, with $\mathbb{S}^3$ denoting the unit hypersphere of $\mathbb{R}^4$, is given by:

$$\boldsymbol{\epsilon} = \begin{bmatrix} v \\ \boldsymbol{u} \end{bmatrix} = \begin{bmatrix} cos\left(\frac{\theta}{2}\right) \\ sin\left(\frac{\theta}{2}\right)\boldsymbol{n} \end{bmatrix}, \tag{1}$$

where $v \in \mathbb{R}$ and $\boldsymbol{u} \in \mathbb{R}^3$, following the notation used by [14]. Let us also define $\bar{\boldsymbol{\epsilon}} = \begin{bmatrix} v & -\boldsymbol{u}^\top \end{bmatrix}^\top$ as the conjugate quaternion. Note that unit quaternions are related to the axis-angle parameterization of $SO(3)$ through the rightmost term of (1). Consequently, it is possible to obtain, from a unit quaternion, a 3D unit vector $\boldsymbol{n}$ and an angle θ, which are the two components of the axis-angle parameterization. The mapping between these two parameterizations of orientation plays an important role in our approach and is covered with greater detail in Section IV.

The product between two quaternions (that preserves membership of $\mathbb{S}^3$) is given by:

$$\boldsymbol{\epsilon}_1 * \boldsymbol{\epsilon}_2 = \begin{bmatrix} v_1 v_2 - \boldsymbol{u}_1^\top \boldsymbol{u}_2 \\ v_1\boldsymbol{u}_2 + v_2\boldsymbol{u}_1 + \boldsymbol{u}_1 \times \boldsymbol{u}_2 \end{bmatrix}, \tag{2}$$

and can be interpreted as a rotation operator: when $\boldsymbol{\epsilon}_1$ and $\boldsymbol{\epsilon}_2$ are defined with respect to the same reference frame, the quaternion product rotates the frame whose orientation is described by $\boldsymbol{\epsilon}_2$ by the rotation defined by $\boldsymbol{\epsilon}_1$. Alternatively, the quaternion product can also be seen as an operator that maps orientations between frames. Thus, if $\boldsymbol{\epsilon}_2$ describes the orientation of a frame C with respect to another frame B, and $\boldsymbol{\epsilon}_1$ describes the rotation of B with respect to a frame A, then the quaternion product (2) gives the orientation of C with respect to A.

Moreover, the product between two quaternions can alternatively be represented by:

$$\boldsymbol{\epsilon} = \boldsymbol{\mathcal{E}}_1\boldsymbol{\epsilon}_2, \text{with } \boldsymbol{\mathcal{E}}_1 = \begin{bmatrix} v_1 & -u_{1,1} & -u_{1,2} & -u_{1,3} \\ u_{1,1} & v_1 & -u_{1,3} & u_{1,2} \\ u_{1,2} & u_{1,3} & v_1 & -u_{1,1} \\ u_{1,3} & -u_{1,2} & u_{1,1} & v_1 \end{bmatrix}, \tag{3}$$

where $\boldsymbol{\mathcal{E}}_1 \in \mathbb{R}^{4\times4}$ (to which we will refer as *quaternion matrix*) is built from the quaternion elements to implement the quaternion product through matrix-vector multiplication.

By taking advantage of the matrix representation of quaternions, we can build compact representations of demonstrated full end-effector poses using TP-GMM, in particular by setting $\boldsymbol{\xi}_n = \hat{\boldsymbol{\epsilon}}_n$, $\boldsymbol{b}_{n,j} = \boldsymbol{0}$ and $\boldsymbol{A}_{n,j} = \boldsymbol{\mathcal{E}}_{n,j}$ as the orientation elements of the task-parameters (the complete structure including position is given in section V). Here, $\boldsymbol{\mathcal{E}}_{n,j}$ represents the orientation of frame j at time step n, expressed as a quaternion matrix, $\boldsymbol{0}$ is a 4×1 vector and $\hat{\boldsymbol{\epsilon}}_n$ is a quaternion obtained from the demonstrations.

When the demonstrated trajectories are projected on the frames, the operation $\boldsymbol{A}_{n,j}^{-1}(\boldsymbol{\xi}_n - \boldsymbol{b}_{n,j})$ yields $\boldsymbol{\mathcal{E}}_{n,j}^{-1}\hat{\boldsymbol{\epsilon}}_n$, for the quaternion part of the task-parameters. This product is equivalent to the quaternion product $\bar{\boldsymbol{\epsilon}}_{n,j} * \hat{\boldsymbol{\epsilon}}_n$, mapping the reference orientation $\hat{\boldsymbol{\epsilon}}_n$ to frame j. Therefore, the proposed formulation allows TP-GMM to build local representations of the demonstrated skill, not only at the level of position, as in [5], but also orientation. As we will see next, this feature allows the generalization of the demonstrated orientations to new frame orientations.

B. Reproduction using products of linearly transformed Gaussians

The learned model can be used to reproduce movements in new situations, that is, generalizing to new frame positions and orientations. The model first retrieves, at each time step n, a temporary GMM by computing a product of linearly transformed Gaussians [7]:

$$\mathcal{N}(\boldsymbol{\mu}_{n,i}, \boldsymbol{\Sigma}_{n,i}) \propto \prod_{j=1}^{P} \mathcal{N}\Big(\boldsymbol{A}_{n,j}\boldsymbol{\mu}_i^{(j)} + \boldsymbol{b}_{n,j},\ \boldsymbol{A}_{n,j}\boldsymbol{\Sigma}_i^{(j)}\boldsymbol{A}_{n,j}^\top\Big), \tag{4}$$

For the elements that concern the quaternion data, the term $\boldsymbol{A}_{n,j}\boldsymbol{\mu}_i^{(j)} + \boldsymbol{b}_{n,j}$ becomes $\boldsymbol{\mathcal{E}}_{n,j}\boldsymbol{\mu}_i^{\boldsymbol{\epsilon}(j)}$, where $\boldsymbol{\mu}_i^{\boldsymbol{\epsilon}(j)}$ denotes the quaternion elements of the local i-th Gaussian center in frame j. Note that $\boldsymbol{\mu}_i^{\boldsymbol{\epsilon}(j)}$ contains the orientation of the end-effector with respect to frame j, while $\boldsymbol{\mathcal{E}}_{n,j}$ represents the orientation of frame j with respect to the robot's base. Their product gives the desired orientation of the end-effector with respect to the robot's base for any value of $\boldsymbol{\mathcal{E}}_{n,j}$ (i.e., generalization to new frame orientations).

By using the temporary GMM computed in (4) for a given set of task parameters, we resort to Gaussian mixture regression (GMR) to retrieve, at each time step, a

position and orientation reference. Specifically, GMR relies on the joint distribution $\mathcal{P}(\boldsymbol{\xi}^{\mathcal{I}}, \boldsymbol{\xi}^{\mathcal{O}})$ learned by the task-parameterized GMM. The conditional probability $\mathcal{P}(\boldsymbol{\xi}_n^{\mathcal{O}}|\boldsymbol{\xi}_n^{\mathcal{I}})$ is then estimated as an output distribution $\mathcal{N}(\hat{\boldsymbol{\mu}}_n^{\mathcal{O}}, \hat{\boldsymbol{\Sigma}}_n^{\mathcal{O}})$ that is also Gaussian [7].

Note that we encode unit quaternions in a TP-GMM as 4-dimensional vectors, without imposing any constraints on the norm, so the output of GMR yields a 4-dimensional vector which might not be of unit norm. Therefore, at each time step n of the reproduction, we normalize the entries of $\boldsymbol{\xi}_n^{\mathcal{O}}$ that correspond to the quaternion reference. In Section V we will show that, even though this post-processing step is an approximation, the accuracy of the reproductions with respect to the demonstrations is not compromised.

IV. Encoding and retrieving full poses with statistical dynamical systems

In [5] we have introduced a probabilistic formulation of dynamic movement primitives. The approach consists of modeling, after gravity compensation of the robot, the position of the robot's end-effector as a virtual unit mass driven by a weighted superposition of spring-damper systems, whose equilibrium points and weights are inferred from the demonstrations. This approach provides several advantages for task-space control, namely: robust handling of perturbations during task execution, and the possibility of selecting the tracking gains according to the desired level of compliance. Assuming constant stiffness and damping gain matrices, selected *a priori*, the learning problem becomes that of computing a virtual attractor for the end-effector position from every demonstration and encoding the distribution of demonstrated attractors in a statistical model, as described in Section III. GMR (Section III-B) is then used during reproductions to estimate the attractor as a normal distribution with full covariance matrix. Note that the tracking gains can be estimated from the covariance [7], but in the present work, as in [5], they were predefined. In this section we explain how the probabilistic formulation of dynamic movement primitives can be generalized to orientations using quaternion notation.

A. Statistical dynamical systems for position

The dynamics of a unit mass-spring-damper system for position are governed by a second order linear differential equation, given by:

$$\ddot{\boldsymbol{x}} = \boldsymbol{K}_p(\hat{\boldsymbol{x}} - \boldsymbol{x}) - \boldsymbol{K}_v\dot{\boldsymbol{x}}, \tag{5}$$

where $\boldsymbol{K}_p$, $\boldsymbol{K}_v$ are respectively the stiffness and damping gain matrices. The variable $\hat{\boldsymbol{x}}$ represents the virtual attractor that is computed for every demonstration based on the assumed dynamics given in (5), through the choice of $\boldsymbol{K}_p$, $\boldsymbol{K}_v$. If $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$, $\ddot{\boldsymbol{x}}$ are, respectively, the demonstrated position, velocity, and acceleration at any given instant, the corresponding virtual attractor is computed through:

$$\hat{\boldsymbol{x}} = \boldsymbol{K}_p^{-1}\ddot{\boldsymbol{x}} + \boldsymbol{K}_p^{-1}\boldsymbol{K}_v\dot{\boldsymbol{x}} + \boldsymbol{x}. \tag{6}$$

By formulating demonstrations in this way, it is possible to reproduce the demonstrated movement with any desired dynamics by setting $\boldsymbol{K}_p$ and $\boldsymbol{K}_v$ accordingly.

B. Statistical dynamical systems for orientation

In order to be able to generate movements with different levels of compliance in orientation, we formulate the learning problem as that of learning an attractor in unit quaternion space, similarly to position. We rely on the formulation proposed in [14], where the authors extend the DMP framework with a formulation that encodes orientation trajectories in $SO(3)$ using both direction-cosine matrices and unit quaternions. Since the DMP is described by a second order linear differential equation, the adaptation of the proposed rotational DMPs to the mass-spring-damper system formulation proposed in our work is straightforward:

$$\dot{\boldsymbol{\omega}} = \boldsymbol{K}_o\, 2\log(\hat{\boldsymbol{\epsilon}} * \bar{\boldsymbol{\epsilon}}) - \boldsymbol{K}_\omega\boldsymbol{\omega}, \tag{7}$$

where $\boldsymbol{K}_o$, $\boldsymbol{K}_\omega \in \mathbb{R}^{3\times3}$ are respectively the angular stiffness and damping matrices, while $\boldsymbol{\omega}$, $\dot{\boldsymbol{\omega}} \in \mathbb{R}^3$ represent the angular velocity and acceleration. The orientation at any given instant is denoted by $\boldsymbol{\epsilon}$ while $\hat{\boldsymbol{\epsilon}}$ represents the orientation attractor, analogous to $\hat{\boldsymbol{x}}$ in (5). The quaternion product $\hat{\boldsymbol{\epsilon}} * \bar{\boldsymbol{\epsilon}}$ gives the orientation error in unit quaternion space, similarly to $(\hat{\boldsymbol{x}} - \boldsymbol{x})$ in (5), that is, the amount by which the orientation represented by $\boldsymbol{\epsilon}$ needs to be rotated in order to reach $\hat{\boldsymbol{\epsilon}}$ in the unit time. The logarithmic map, log: $\mathbb{S}^3 \to \mathbb{R}^3$, converts the quaternion error into an axis-angle representation:

$$\log(\boldsymbol{\epsilon}) = \log\left(\begin{bmatrix} v \\ \boldsymbol{u} \end{bmatrix}\right) = \begin{cases} \arccos(v)\frac{\boldsymbol{u}}{||\boldsymbol{u}||}, & \boldsymbol{u} \neq 0 \\ [0\ \ 0\ \ 0]^\top, & \text{otherwise.} \end{cases}$$

An inverse map, given by the exponential map, exp: $\mathbb{R}^3 \to \mathbb{S}^3$, converts an orientation in axis-angle representation into a quaternion:

$$\exp(\boldsymbol{r}) = \begin{cases} \left[\cos(||\boldsymbol{r}||) \quad \sin(||\boldsymbol{r}||)\frac{\boldsymbol{r}^\top}{||\boldsymbol{r}||}\right]^\top, & \boldsymbol{r} \neq 0 \\ [1\ \ 0\ \ 0\ \ 0]^\top, & \text{otherwise,} \end{cases}$$

where $\boldsymbol{r} \in \mathbb{R}^3$ is a vector that represents an orientation in axis-angle notation. For $||\boldsymbol{r}|| < \pi$ the two mappings are bijective and inverse to each other, and $\hat{\boldsymbol{\epsilon}}$ can be easily computed from a given quaternion, angular velocity and acceleration by re-writing (7) as:

$$\hat{\boldsymbol{\epsilon}} = \exp\left(\frac{1}{2}\boldsymbol{K}_o^{-1}\dot{\boldsymbol{\omega}} + \frac{1}{2}\boldsymbol{K}_o^{-1}\boldsymbol{K}_\omega\boldsymbol{\omega}\right) * \boldsymbol{\epsilon}, \tag{8}$$

which is the rotational counterpart of (6). This formulation takes into account the topology of $SO(3)$ by computing the attractor using the exponential map between $\mathbb{R}^3$ and $\mathbb{S}^3$. Consequently, it is possible to retrieve a quaternion attractor, based on the assumed dynamics for the demonstrated orientation trajectory, through the choice of $\boldsymbol{K}_o$ and $\boldsymbol{K}_\omega$. With this formulation we generalize the learning problem to encode full end-effector poses, becoming that of learning an attractor in both position and orientation. We therefore use the attractors $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{\epsilon}}$, defined in this section, as the position and orientation references in the TP-GMM.

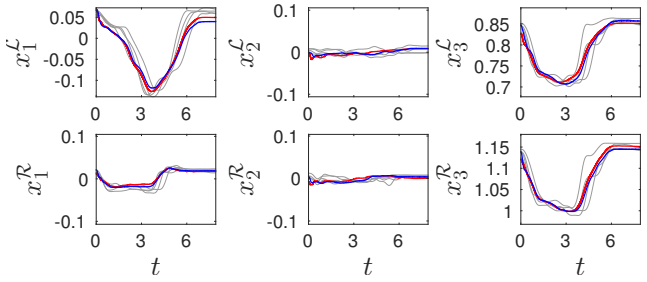

Fig. 2. Time evolution of the position dimensions of both end-effectors during one reproduction attempt, represented in the coordinate system of the area to be swept by the broom. The grey lines show the demonstrated trajectories, the red line shows the position attractor that is output by GMR and the blue line shows the resulting end-effector positions. Time is displayed in seconds and positions in meters.

V. Experimental Results

We apply the proposed framework to the learning of a bimanual sweeping task, a particular case where bimanual coordination patterns, that encompass both position and orientation constraints, arise. For this task, we employed two torque-controlled 7-DoF WAM robots (see Fig. 1). A broom is attached to the tool plate of the right arm using a Cardan joint, while the left arm uses a Barrett robotic hand to hold the broom. Since the broom is passively attached to the right arm, the sweeping movement consists of a rotation between the two end-effectors, with the hand grabbing the broom and describing a pendulum-like motion with respect to the end-effector of the right arm.

We collected 4 demonstrations with a duration of approximately 8 seconds each through kinesthetic teaching, while the position and orientation of each of the end-effectors were recorded. During the demonstrations, both robots were physically moved with a controller compensating for the effect of gravity. Since quaternions provide a double coverage of $SO(3)$ ($\boldsymbol{\epsilon}$ and $-\boldsymbol{\epsilon}$ represent the same orientation) we pre-process the data to ensure that all demonstrated quaternions are in the same hemisphere of $\mathbb{S}^3$. In order to learn the movement, we trained the task-parameterized model using $K = 10$ Gaussian components, chosen empirically (methods based on a Bayesian information criterion [16] could alternatively be used for model selection).

In all of the experiments described in this section we have used $\boldsymbol{K}_p = 500\boldsymbol{I}$ and $\boldsymbol{K}_v = 45\boldsymbol{I}$ to compute the position attractors of the left arm, $\boldsymbol{K}_p = 250\boldsymbol{I}$ and $\boldsymbol{K}_v = 35\boldsymbol{I}$ for the right arm and $\boldsymbol{K}_o = 250\boldsymbol{I}$, $\boldsymbol{K}_\omega = 35\boldsymbol{I}$ to compute the orientation attractors of both arms (where $\boldsymbol{I} \in \mathbb{R}^{3\times3}$). The values of $\boldsymbol{K}_v$ and $\boldsymbol{K}_\omega$ were chosen empirically by keeping the unit mass-spring-damper system overdamped ($\boldsymbol{K}_{v,\omega} > 2\sqrt{\boldsymbol{K}_{p,o}}$). A video accompanying this paper shows the results of the experiments, and is available at `http://programming-by-demonstration.org/iros2015/`. We also provide Matlab source codes presenting the overall approach described in sections III and IV with a simple example (codes compatible with GNU Octave).

A. Learning generalized bimanual coordination patterns

We encode the task using two frames ($P = 2$): a moving frame ($j = 1$) given for each arm by the coordinate system defined by the position and orientation of the opposite end-effector, and a fixed frame ($j = 2$), defined by the area to be swept by the broom. Instead of considering two separate TP-GMM (one per arm) we use one single model, for compactness, that includes the task-parameters associated with both end-effectors. Therefore, we define $\boldsymbol{\xi}_n$ and $\boldsymbol{b}_{n,1}, \boldsymbol{A}_{n,1}$, the task parameters of frame $j = 1$, as:

$$\boldsymbol{\xi}_n = \begin{bmatrix} t_n \\ \hat{\boldsymbol{x}}_n^{\mathcal{L}} \\ \hat{\boldsymbol{\epsilon}}_n^{\mathcal{L}} \\ \hat{\boldsymbol{x}}_n^{\mathcal{R}} \\ \hat{\boldsymbol{\epsilon}}_n^{\mathcal{R}} \end{bmatrix} \begin{matrix} \} \boldsymbol{\xi}_n^{\mathcal{I}} \\ \\ \\ \\ \} \boldsymbol{\xi}_n^{\mathcal{O}} \end{matrix}, \ \boldsymbol{b}_{n,1} = \begin{bmatrix} 0 \\ \boldsymbol{x}_n^{\mathcal{R}} \\ \boldsymbol{0} \\ \boldsymbol{x}_n^{\mathcal{L}} \\ \boldsymbol{0} \end{bmatrix}, \ \boldsymbol{A}_{n,1} = \begin{bmatrix} 1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_n^{\mathcal{R}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\mathcal{E}}_n^{\mathcal{R}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R}_n^{\mathcal{L}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\mathcal{E}}_n^{\mathcal{L}} \end{bmatrix},$$

where the superscripts $\mathcal{L}$ and $\mathcal{R}$ denote the left and right arm respectively, indicating to which arm corresponds each rotation matrix $\boldsymbol{R}_n$, quaternion matrix $\boldsymbol{\mathcal{E}}_n$, end-effector position $\boldsymbol{x}_n$ and attractors $\hat{\boldsymbol{x}}_n$ and $\hat{\boldsymbol{\epsilon}}_n$. In addition, t_n is a time step and $\boldsymbol{0}$ are zero matrices of appropriate dimension.

The frame $j = 2$ is given by the position of the point to be swept by the broom on the floor and an orientation corresponding to that of the bases of the robots (which are aligned in our case). We thus have $\boldsymbol{b}_{n,2} = \begin{bmatrix} 0 & \boldsymbol{x}_{sweep}^{\mathcal{L}\top} & \boldsymbol{0} & \boldsymbol{x}_{sweep}^{\mathcal{R}\top} & \boldsymbol{0} \end{bmatrix}^\top$ and $\boldsymbol{A}_{n,2} = \boldsymbol{I}$ (where $\boldsymbol{I} \in \mathbb{R}^{15\times15}$). Note that this definition of frames remains valid for a wide range of bimanual skills involving objects or landmarks. In the first experiment, the values of $\boldsymbol{x}_{sweep}^{\mathcal{L}}$ and $\boldsymbol{x}_{sweep}^{\mathcal{R}}$ were randomly selected from the demonstration set[1].

Figures 2 and 3 show the positions and orientations of both end-effectors over time during the reproduction of one sweep, projected on the frame of the area to be swept, together with local representations of the demonstrations. We observe that our framework successfully reproduces both the demonstrated position and orientation profiles.

The role of the end-effector frames in coordination: As aforementioned, the main advantage of encoding the task in the frame of the other end-effector is that the bimanual coordination patterns underlying the task, both in position and orientation, are encapsulated in the model. In order to provide a practical example of this property, in particular at the level of orientation, we train a new model using $P = 1$, that describes the movement only in the frame of the sweeping area. We then compare how both models react to perturbations, in particular how one arm compensates for perturbations applied to the other. Our hypothesis is that the model with $P = 2$ should provide a better compensation given that, in addition to the area to be swept, it also takes into consideration the position and orientation constraints between the two end-effectors. We therefore apply a perturbation to the right arm which consists of adding, via the controller, a force and a torque to its end-effector in task-space. The force is applied along the negative direction of the x_2-axis, while the torque is applied around the x_3-axis of the base of the robot.

[1] Alternatively, vision or optical tracking systems may be used to obtain the parameters of the area to be swept.

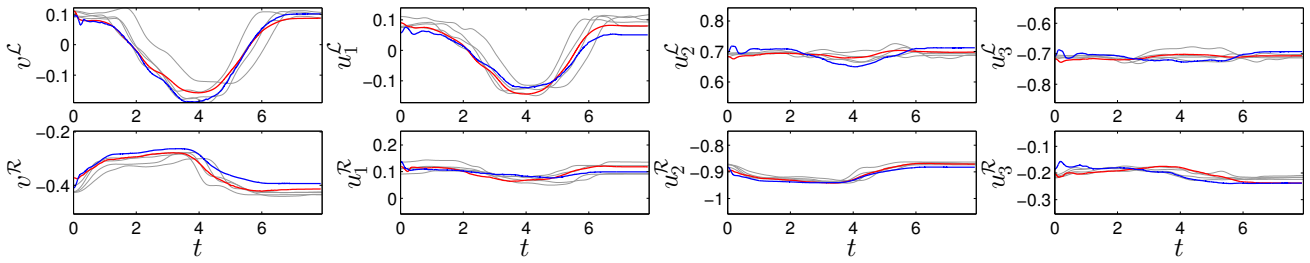

Fig. 3. Time evolution of the quaternions representing the orientation of each end-effector with respect to the coordinate system of the sweeping area, during one reproduction attempt. Grey lines correspond to the demonstrations, while red and blue lines depict the quaternion attractors (normalized output of GMR) and the retrieved quaternions, respectively.

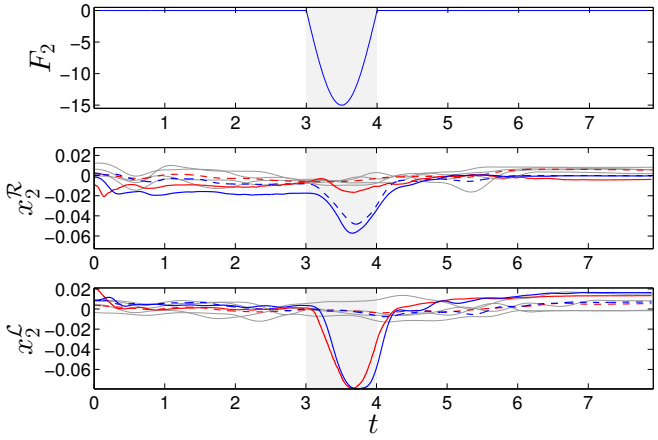

Fig. 4. Response of the position along x_2 of both arms when a force perturbation is applied to the right arm. The first plot depicts the applied force. The middle and bottom plots represent the attractor (red) and retrieved position value (blue) of the right and left arms, respectively. Dashed lines correspond to $P = 1$ while solid lines correspond to $P = 2$. Time is in seconds, forces are in Newton, and positions are in meters.

1) Position: Fig. 4 shows the applied force and how it affects the x_2 coordinate of both end-effectors. In particular, we observe that the perturbation generates a displacement along $x_2^{\mathcal{R}}$, for both $P=1$ (dashed blue line) and $P=2$ (solid blue line). The effect of this perturbation on the left arm, however, differs between $P = 1$ and $P = 2$. In the former case, the attractor $\hat{x}_2^{\mathcal{L}}$ keeps its trend during the perturbation, while in the latter, the model compensates the displacement that occurred in $x_2^{\mathcal{R}}$ by shifting the attractor $\hat{x}_2^{\mathcal{L}}$ accordingly.

2) Orientation: We begin by computing the sequence of quaternions that represent the orientation between the two end-effectors during a reproduction of the task, $\bar{\boldsymbol{\epsilon}}_n^{\mathcal{L}} * \boldsymbol{\epsilon}_n^{\mathcal{R}}$. These quaternions are then converted into Euler angles, for a more intuitive interpretation. Figure 5 shows the 3 Euler angles (α, β, γ for rotations around x_3, x_2, x_1, respectively) alongside with the applied torque. We observe that, during the perturbation, the drift in orientation, with respect to the demonstrations, is greater for the model that was trained using only the frame of reference of the area to be swept (dashed line). As for position, the coordination patterns at orientation level are encoded in the model when $P = 2$, resulting in a better reaction to perturbations since the relative orientation is considered by the model.

These results show that the proposed learning framework is able to successfully encode and synthesize position and orientation constraints in bimanual manipulation scenarios,

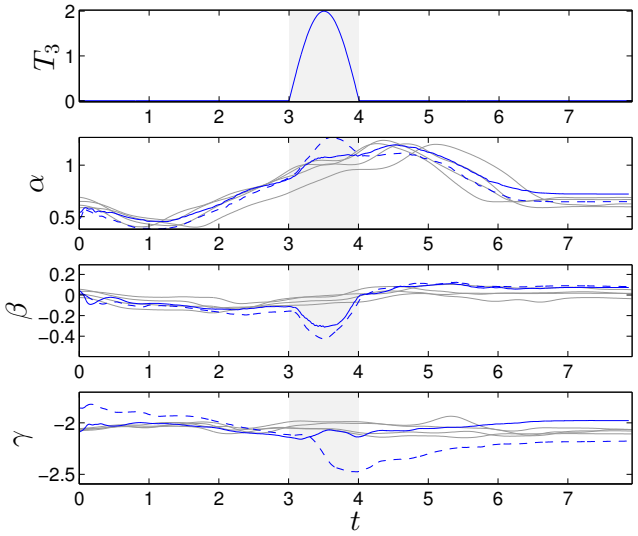

Fig. 5. The first plot depicts the task-space perturbation torque that was applied to the end-effector of the right arm. The remaining three plots represent the orientation between the two end-effectors as Euler angles (computed, for visualization purposes, from the quaternions $\bar{\boldsymbol{\epsilon}}_n^{\mathcal{L}} * \boldsymbol{\epsilon}_n^{\mathcal{R}}$ for every n). The dashed line corresponds to the model that was trained using $P = 1$, while the solid line corresponds to $P = 2$. Time is in seconds, torques are in Newton-meter, and angles are in radians.

by taking advantage of the new structure of the task parameters, described in Section III. In particular, they highlight how encoding the bimanual coordination patterns in the model is essential for a robust execution of the task and how the quaternion-based dynamical systems formulation yields a correct reproduction of the demonstrated orientation patterns, even when facing perturbations.

B. Extrapolation to new orientations

We now test the generalization capabilities of the approach. A new reproduction is performed with a new set of task parameters for the frame of the sweeping area, $j = 2$, in which its position was shifted by 0.28m along the negative direction of x_1, bringing it closer to the bases of the robots, and by 0.05m along the positive direction of x_2. In addition, the frame was rotated clockwise by 45°, with respect to x_3. Figures 6 and 7 show the Euler angles that represent the orientation of both end-effectors during one reproduction of the sweeping movement using the new frame. The orientations are represented in the coordinate system of the bases of the robots (Fig. 6) and in the coordinate system of the area to sweep (Fig. 7). Figure 6 shows that the values of $\alpha^{\mathcal{L}}$ and $\alpha^{\mathcal{R}}$ (rotations around x_3),

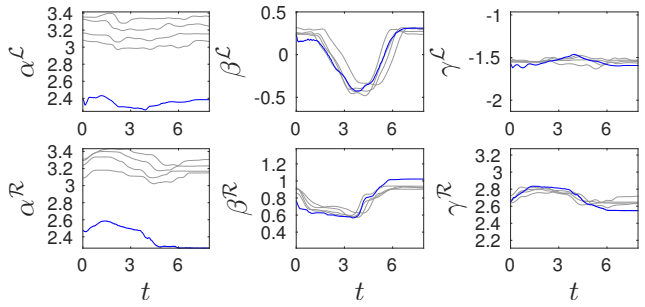

Fig. 6. The orientation of the end-effectors with respect to the frames of the robot bases (represented in Euler angles) during a reproduction attempt with new task parameters. Blue lines show the retrieved Euler angles, while grey lines show the demonstrations. Angles are in radians and time in seconds.

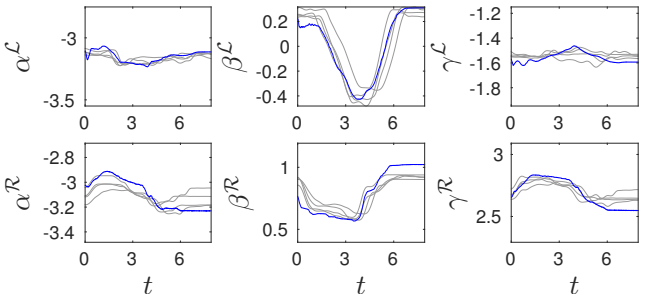

Fig. 7. End-effector orientations, represented in Euler angles, locally represented in the coordinate system of the sweeping area for the new values of the task parameters. The blue lines correspond to the retrieved values during the reproduction, while the demonstrations are depicted in grey. Angles are in radians and time is in seconds.

that were retrieved during the reproduction, have an offset of about $0.8rad$ ($\approx 45°$) with respect to the demonstrations. Hence, we can conclude that the movement was properly extrapolated to an orientation that was not demonstrated. In addition, as we can see in Fig. 7, the orientation of both end-effectors represented in the frame of the sweeping area remained consistent with the demonstrations in that frame, confirming that the movement was correctly reproduced.

VI. Conclusions and Future Work

We proposed a method that combines task-parameterized Gaussian mixture models and dynamical systems to learn full end-effector poses in a Programming by Demonstration scenario. The approach was showcased through a bimanual sweeping experiment, where the orientation between the two end-effectors is essential for the correct execution of the task. We showed that the proposed formulation of TP-GMM, simultaneously encoding the demonstrations in multiple frames, can be extended to task-adaptive orientation control to efficiently encode and retrieve coordination patterns between the two end-effectors. In addition, this formulation makes it possible to generalize the demonstrated orientation profiles to unseen situations, namely, to new orientations of the area to be swept. Finally, we showed that the quaternion-based dynamical systems formulation permits the computation of virtual attractors in $SO(3)$ that consider a desired impedance for the reproduction of the task.

In the proposed approach, unit quaternions are modeled probabilistically as 4-dimensional vectors, without taking into account the unit norm constraint inherent to this parameterization of $SO(3)$. Therefore, the output of GMR has to be normalized. As future work, this issue could be addressed by studying alternative ways of modeling quaternion distributions that consider the fact that quaternions are elements of $\mathbb{S}^3$. One possible avenue could be that of exploiting the Bingham distribution [17].

Another route for future work is to exploit the structure of the controllers in (5) and (7) to autonomously determine impedance gains. Hence, we plan to apply optimal control methods to learn optimal orientation gains, similarly to [7]. Ultimately, alternative representations could also be explored to describe end-effector poses. Representations such as those used in [11] and [18], that employ dual quaternions, could potentially be extended to a task-parameterized formulation.

References

[1] A. G. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Secaucus, NJ, USA: Springer, 2008, pp. 1371–1394.

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

[3] S. Vijayakumar, A. D'souza, and S. Schaal, "Incremental online learning in high dimensions," *Neural Computation*, vol. 17, no. 12, pp. 2602–2634, Dec. 2005.

[4] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, no. 25, pp. 328–373, 2013.

[5] S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis, and D. G. Caldwell, "Statistical dynamical systems for skills acquisition in humanoids," in *Proc. IEEE Humanoids*, Osaka, Japan, 2012, pp. 323–329.

[6] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, "Dual arm manipulation – a survey," *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1340 – 1353, 2012.

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

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

[9] J. Umlauft, D. Sieber, and S. Hirche, "Dynamic movement primitives for cooperative manipulation and synchronized motions," in *Proc. IEEE ICRA*, Hong Kong, China, May-June 2014, pp. 766–771.

[10] D. Sieber, F. Deroo, and S. Hirche, "Formation-based approach for multi-robot cooperative manipulation based on optimal control design," in *Proc. IEEE/RSJ IROS*, Tokyo, Japan, November 2013, pp. 5227–5233.

[11] M. Lang, O. Dunkley, and S. Hirche, "Gaussian process kernels for rotations and 6D rigid body motions," in *Proc. IEEE ICRA*, Hong Kong, China, May-June 2014, pp. 5165–5170.

[12] S. Calinon, T. Alizadeh, and D. G. Caldwell, "On improving the extrapolation capability of task-parameterized movement models," in *Proc. IEEE/RSJ IROS*, Tokyo, Japan, November 2013, pp. 610–616.

[13] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal, "Online movement adaptation based on previous sensor experiences," in *Proc. IEEE/RSJ IROS*, San Francisco, USA, September 2011, pp. 365–371.

[14] A. Ude, B. Nemec, T. Petrič, and J. Morimoto, "Orientation in cartesian space dynamic movement primitives," in *Proc. IEEE ICRA*, Hong Kong, China, May-June 2014, pp. 2997–3004.

[15] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, 2009.

[16] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[17] J. Glover and L. P. Kaelbling, "Tracking the spin on a ping pong ball with the quaternion Bingham filter," in *Proc. IEEE ICRA*, Hong Kong, China, May-June 2014, pp. 4133–4140.

[18] B. Adorno, P. Fraisse, and S. Druon, "Dual position control strategies using the cooperative dual task-space framework," in *Proc. IEEE/RSJ IROS*, Taipei, Taiwan, October 2010, pp. 3955–3960.