2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Learning robust manipulation tasks involving contact using trajectory parameterized probabilistic principal component analysis

Cristian Vergara Perico[1], Joris De Schutter[1] and Erwin Aertbeliën[1]

***Abstract*— In this paper, we aim to expedite the deployment of challenging manipulation tasks involving both motion and contact wrenches (forces and moments). To this end, we acquire motion and wrench signals from a small set of demonstrations using passive observation. To learn these tasks, we introduce *Trajectory parameterized Probabilistic Principal Component Analysis* (traPPCA) which compactly re-parameterizes the acquired signals using trajectory information and encodes the signal correlations using *Probabilistic Principal Component Analysis* (PPCA). Finally, the task is transferred to a robot setup by specifying the robot behavior using a constraint-based task specification and control approach. This framework results in increased robustness of the system against different sources of uncertainty: imprecise sensors, adaptation of the tool, and changes in the execution speed.**

I. Introduction

Robot manipulation tasks often correspond to challenging operations that involve both motion and contact wrenches. As an example, let's consider the manipulation of a bottle opener to remove a crown cork from a bottle. We find inspiration for our control scheme in the human dexterity to perform manipulation tasks. We hypothesize that humans have prior knowledge about the required motion to approach with the bottle opener to align its hook with the crown cork, as well as an estimate of the required movement to remove the crown cork. However, the latter fine motion is performed using the contact wrenches as feedback. Hence, they perform the corresponding movements while feeling resistance due to forces and moments. There is a large variety of tasks that involve a similar interaction between motion and contact wrenches, such as contour following, painting, sanding, among others.

We leverage passive observation [1] to facilitate the acquisition of the task information from human demonstrations. To this end, using a device equipped with a pose tracker and a force/torque (F/T) sensor, we perform a small set of demonstrations towards different positions of the workpiece while recording positions, orientations, forces, and moments. We use these demonstrations to capture the trajectory shape, but also to delimit the workspace in which the robot will execute the task.

Our system generalizes this information by extending the methodology presented in [2] which combines *Learning from Demonstration* LfD (only at the position level without learning the time evolution) with constraint-based task specification and control. On one hand, in this work, we introduce traPPCA to learn the motion model. This method re-parameterizes the acquired signals with respect to inherent trajectory information. First, all the signals are represented in a relevant *task frame* (e.g., the bottle opener hook). Then, instead of de-correlating poses and their time derivatives *doubling the number of the pose signals*, as in the state of the art, we encode the time evolution by re-parameterizing these signals with respect to the degree-of-advancement speed, which contains information of the time evolution of the position and orientation. This methodology increases the vector of task signals with only one signal, while increasing the robustness against signal noise and misalignment of the approach and contact phases. As in [2], the variability of the whole set of signals and demonstrations is encoded using the dimensionality reduction technique PPCA introduced in [3].

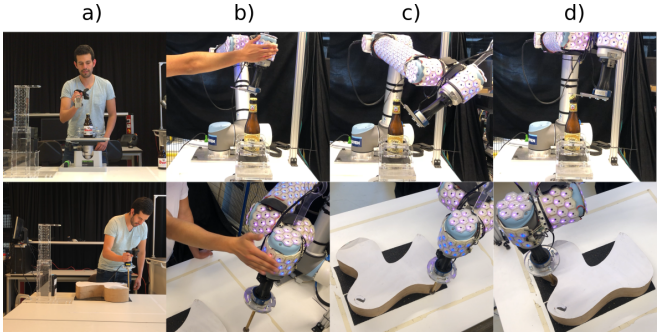

Fig. 1: Two robot applications are learned and deployed: bottle-opening (up) and non-clamped contour-following (down). a) Demonstrations are performed using passive observation. b) A robot moves to make contact on a stiff workpiece. Collisions are avoided during the approach motions based on proximity signals. c) The system aligns to reach the taught first contact point. d) pose, wrench and evolution constraints are used to robustly execute the learned tasks. See video attached to this paper.

On the other hand, we based our robot controller on task specification of motions involving contact wrenches in the *Task Frame Formalism* (TFF) introduced in [4]. To this end, we implement the TFF in the constraint-based framework *expression graph based Task Specification Language* (eTaSL) [5]. As a result, our system is able to generalize robot motions that involve contact wrenches towards non-demonstrated situations based on constraints imposed on the evolution, pose, and full wrench signals, while at the same time complying with environment and robot constraints (e.g., proximity-based collision avoidance [6] and joint limits).

The main contribution of this paper is to present a framework that combines the following features:

1) The use of passive observation to enable a user to perform demonstrations intuitively. In contrast to kinesthetic teaching, a dedicated device (see Fig. 2) enables

[1] Robotics Research Group, PMA Division, KU Leuven, Belgium, Core Lab Flanders Make, www.mech.kuleuven.be/robotics. All authors gratefully acknowledge the financial support of Flanders Make through projects YVES_SBO and PROUD. Corresponding author: cristian.vergara@kuleuven.be

the acquisition of pose and contact wrench signals without the need to compensate for the influence of friction and inertial properties of the robot.
2) The use of trajectory information, such as the task frame and its time evolution, to re-parameterize evolution, pose and wrench signals. This methodology facilitates the alignment of the contact and non-contact phases while decreasing the number of signals corresponding to the motion compared with the state of the art.
3) The implementation of the TFF using pose, wrench, and evolution soft-constraints in eTaSL. This methodology enables our system to generalize signals using the information encoded in the learned model while complying with robot and environment constraints. As a result, our system has increased robustness against different sources of geometric uncertainty such as transferring the task to a robot setup and imprecise sensors.
4) The modularity and composability offered by the constraint-based framework eTaSL. These properties enable a user to modify the degree-of-advancement speed and the tool dimensions (only adapting the already available kinematic model) while *maintaining* the learned model, control specification, and other hyperparameters.
5) The evaluation of our system by robustly generalizing, in real-world scenarios, evolution, pose and wrench signals from a small set of demonstrations. As a result, our system can execute approach-and-contact tasks in both *stiff* and *deformable* environments without using motion-capture aids such as markers or trackers.

This framework is evaluated by learning and deploying two robot applications (see Fig. 1): first, a robot is taught to approach and open bottles with varying sizes positioned and clamped vertically in non-demonstrated locations; and second, a robot is taught to approach to get into contact with a non-clamped workpiece positioned in non-demonstrated locations, and subsequently follow a section of its contour while maintaining contact.

II. Related Work and Assumptions

Several learning algorithms have been used to encode information from demonstrations in a set of basis functions. For instance, *Probabilistic Movement Primitives* ProMPs [7] and *Dynamic Movement Primitives* DMPs [8] encode the information in the weights of chosen radial basis functions, while *Gaussian Mixture Models* GMM, used in [9], [10], [11], [12] employ a mixture of Gaussians as basis functions. As these methods use general Gaussian representations, the number of independent parameters scales quadratically with the number of signals to learn. In contrast, as described in [13], the PPCA algorithm captures the most significant correlation while still ensuring that the number of independent parameters grows only linearly with the number of signals. This aspect is important to deal with the Machine Learning phenomenon: *curse of dimensionality* [10]. Moreover, the PPCA technique generalizes the learned information using a linear combination of a weighted set of basis functions. These weights correspond to the *Degrees Of Freedom* DOF of the learned model and are controlled in eTaSL by specifying them as feature variables while complying with robot and environment constraints.

Several approaches have extended the mentioned methods from only the generalization of joint/cartesian positions in free space towards generalization of: *(i) orientation during transfer motions*, as in [14] by learning unit quaternions using DMPs, [11] by learning TP-GMM on Riemannian Manifolds, or [12] by encoding angular velocity using GMM in Coupled Dynamical Systems (CDS) [9]; *(ii) orientations during contact tasks*, as in [15], [16] by learning unit quaternions using DMPs, *(iii) contact forces*, as in [17], [18] by learning this information using TP-GMM and extracting stiffness parameters from demonstrations in the former, [19] by encoding variable stiffness using a time varying linear Gaussian control law, or [12] by explicitly expressing forces as function of the distance to the CDS attractor; *(iv) contact moments in the operational space*, as in [15] and [16], by learning these variables using DMPs; and *(v) transition between contact and non-contact tasks*, as in [18] by using the variance of a Gaussian Process Regression (GPR) as confidence value of belonging to a segment, [20], [21], [22], [23], [24], [25] by using methods based on *Hidden Markov Models* (HMM) models to represent the different segments, or [12] by performing automatic segmentation using a criterion defined by high variability within each demonstration and systematic changes between different demonstrations. Although these works introduced some of the mentioned features, they did not present a simultaneous combination of all of them. Specifically, on the one hand, all the mentioned methods use Kinesthetic Teaching as the way to perform demonstrations. Although this method allows the system extracting parameters such as joint stiffness from demonstrations, its application is limited to robot platforms that offer active compliance. Furthermore, besides making the demonstration process more cumbersome for non-expert users, the platform's friction and inertial properties also influence the contact wrench measurements. On the other hand, to the authors' knowledge, none of these methods have presented the ability to modify the execution speed and the tool dimensions while maintaining learned model, control specification, and hyperparameters, as well as complying with robot and environment constraints.

To assess the potential to further generalize our approach to other applications, we make our assumptions explicit:

Assumption 1: We consider applications consisting of a non-contact phase (approach motion), followed by a contact phase in which both motion and wrench signals are relevant. The transition between both phases is well-defined by the initial contact of the tool and the workpiece.

Assumption 2: The position of the object, the initial position of the tool, and the shape of the position profile of the approach can vary substantially during demonstrations and executions; the other signals, however, vary only in a limited way.

Assumption 3: All properties of the demonstration and execution set-up are known or can be easily estimated. These

properties correspond to: *(i)* the used robot setup including robot, tool, F/T-sensor, tracking sensor, and camera system; *(ii)* the kinematic model of the robot setup, which can be easily specified using the Universal Robot Description Format (URDF); *(iii)* tool weight and center of gravity, while inertial forces can be neglected due to the low accelerations during demonstration and execution; and *(iv)* a rough estimate of the mechanical contact stiffness between the tool and the workpiece, which is assumed to remain consistent;

Assumption 4: An available industrial vision solution, such as Pick-it, is calibrated to obtain the initial contact position.

III. Information generalization

In this section, we extend the LfD methodology presented in [2] towards traPPCA, which, besides encoding position signals $\boldsymbol{p}_f \in \mathbb{R}^3$, also deals with orientation $\boldsymbol{q}_f \in \mathbb{R}^4$, wrenches $\boldsymbol{u}_f \in \mathbb{R}^6$, time evolution $\dot{\xi} \in \mathbb{R}$, and segmentation between contact and non-contact motions. To this end, a user performs N demonstrations via passive observation (see Fig. 1.a). These demonstrations contain $D = 13$ signals with T samples of pose and wrench measurements. Considering assumptions 1 to 4, these signals are parameterized using trajectory information as follows:

1) Measured poses (position and orientation) of each demonstration are expressed in the task-frame corresponding to the tool and referenced with respect to its initial pose.
2) Based upon two thresholds on the Euclidean norm of the force and moment signals, each demonstration is subdivided into $A = 2$ segments corresponding to motions without and with contact between the tool and the workpiece.
3) The D signals of each a-th segment are normalized separately such that they become independent of time. These signals are re-parameterized using the degree-of-advancement $\xi(t)$, defined in [26] as a linear combination of the normalized position path coordinate and an equivalent value for the orientation. This value is calculated as follows:

$$\xi_a(t) = \mathrm{w}\frac{\int_{t_{a,0}}^{t} ||\boldsymbol{v}_p||\,\mathrm{d}t}{\int_{t_{a,0}}^{t_{a,T}} ||\boldsymbol{v}_p||\,\mathrm{d}t} + (1-\mathrm{w})\frac{\int_{t_{a,0}}^{t} ||\boldsymbol{\omega}||\,\mathrm{d}t}{\int_{t_{a,0}}^{t_{a,T}} ||\boldsymbol{\omega}||\,\mathrm{d}t} \quad (1)$$

$$\xi(t) = \xi_a(t) + (a-1), \quad (2)$$

where $\boldsymbol{v}_p$ and $\boldsymbol{\omega}$ correspond to the translational and angular velocity computed using Euler approximation, respectively; the weighting factor $\mathrm{w} = 0.5$ gives equal importance to the translation and rotation information; and $t_{a,0}$ and $t_{a,T}$, are the initial and final time for each a-th segment, respectively. As a result, for the type of applications aimed in this paper, approach and contact phases correspond to $0 \leq \xi \leq 1$ and $1 \leq \xi \leq 2$, respectively, while $\xi = 1$ corresponds to the expected first contact.

4) The time evolution is encoded in the learned model by first computing the degree-of-advancement speed $\dot{\xi}(t)$ applying the Euler approximation over $\xi(t)$ and then, adding this as a new signal resulting in $D = 14$.
5) Subsequently, each segment a is re-sampled with the same number $T = 50$ of ξ-equivalent samples using spline interpolation.
6) Finally, let us express $f_{d,n}(\xi_j)$ as the j-th sample that corresponds to the d-th signal of the n-th demonstration, with $j \in 1 \dots TA$, $d \in 1 \dots D$, and $n \in 1 \dots N$. For each demonstration n, the discrete samples $f_{d,n}(\xi_j)$ are assembled in the rows of a matrix $\bar{\boldsymbol{F}} \in \mathbb{R}^{N \times TAD}$ in the following form:

$$\bar{\boldsymbol{F}}_{n,TA(d-1)+j}(\xi_j) = f_{d,n}(\xi_j), \quad (3)$$

where the indices of $\bar{\boldsymbol{F}}$ correspond to row and column, respectively.

As in [2], a Gaussian distribution is assumed for the elements of each row in (3). This distribution can be described by the following *latent variable model* as a function of the degree-of-advancement ξ and a vector $\boldsymbol{\chi}_{\boldsymbol{f}_{\mathrm{lv}}} \in \mathbb{R}^M$ containing M latent variables $\boldsymbol{\chi}_{\boldsymbol{f}_{\mathrm{lv}}}$, referred to as PPCA in [3]:

$$\boldsymbol{f}(\xi_j, \boldsymbol{\chi}_{\boldsymbol{f}_{\mathrm{lv}}}) = \boldsymbol{W}(\xi_i)\boldsymbol{\chi}_{\boldsymbol{f}_{\mathrm{lv}}} + \boldsymbol{b}(\xi_i) + \sigma^2(\xi_j), \quad (4)$$

where $\boldsymbol{f}(\xi_j, \boldsymbol{\chi}_{\boldsymbol{f}_{\mathrm{lv}}}) \in \mathbb{R}^D$ is the generated signal, $\boldsymbol{W}(\xi_j) \in \mathbb{R}^{D \times M}$ is a matrix containing $M \leq N-1$ basis functions encoding the signal variability, $\boldsymbol{b}(\xi_j) \in \mathbb{R}^D$ is the signal average over all demonstrations, and $\sigma^2(\xi_j)$ corresponds to a Gaussian noise. As explained in [2], these parameters are calculated by first estimating the maximum likelihood from $\bar{\boldsymbol{F}}$ as follows:

$$\boldsymbol{b}(\xi_j) = \frac{1}{d}\sum_{n=1}^{N} \bar{\boldsymbol{F}}_n(\xi_j)^T, \quad (5)$$

$$\boldsymbol{D}(\xi_j) = \left[\bar{\boldsymbol{F}}_1(\xi_j)^T - \boldsymbol{b}(\xi_j) \dots \bar{\boldsymbol{F}}_N(\xi_j)^T - \boldsymbol{b}(\xi_j)\right], \quad (6)$$

$$\sigma^2(\xi_j) = \frac{1}{DTA - M}\sum_{j=M+1}^{N} \lambda_j, \quad (7)$$

$$\boldsymbol{W}(\xi_j) = \boldsymbol{U}_M(\boldsymbol{\Lambda}_M - \sigma^2\,\boldsymbol{I})^{1/2}\,\mathbf{R}, \quad (8)$$

where $\boldsymbol{U}_M$ and $\{\lambda_1, \dots, \lambda_M\}$ are the eigenvectors and eigenvalues of the covariance matrix $\boldsymbol{D}\boldsymbol{D}^T$, $\boldsymbol{\Lambda}_M$ is a diagonal matrix containing the eigenvalues $\boldsymbol{\Lambda}_M = diag(\lambda_1, \dots, \lambda_M)$; and $\mathbf{R}$ corresponds to an arbitrary transformation matrix that, for convenience, is chosen as the identity. Then, the terms in (4) are made continuous by performing a cubic spline interpolation over the TA samples for each signal as a function of ξ. Therefore, subtracting the signal noise from (4), it yields the following continuous function:

$$\boldsymbol{f}(\xi, \boldsymbol{\chi}_{\boldsymbol{f}_{\mathrm{lv}}}) = \boldsymbol{W}(\xi)\boldsymbol{\chi}_{\boldsymbol{f}_{\mathrm{lv}}} + \boldsymbol{b}(\xi), \quad (9)$$

Finally, the generated signals in (9) are mapped to the following representations: *(i)* the position signals are expressed as a position vector $\boldsymbol{p}_f(\xi, \boldsymbol{\chi}_{\boldsymbol{f}_{\mathrm{lv}}}) \in \mathbb{R}^3$, *(ii)* the quaternion corresponding to orientation signals is normalized and expressed as a rotation matrix $\boldsymbol{R}_f(\xi, \boldsymbol{\chi}_{\boldsymbol{f}_{\mathrm{lv}}}) \in \mathbb{R}^{3\times3}$, *(iii)* the force and moment signals are assembled in a wrench

vector $\boldsymbol{u}_f(\xi, \boldsymbol{\chi}_{\boldsymbol{f}_{\text{lv}}}) \in \mathbb{R}^6$, and *(iv)* the time derivative of the degree-of-advancement is expressed as $\dot{\xi}_f(\xi, \boldsymbol{\chi}_{\boldsymbol{f}_{\text{lv}}}) \in \mathbb{R}$. Note that the generated orientation corresponds to a quaternion linear interpolation. These components are then normalized, resulting in a methodology similar to the well-known *normalized quaternion linear interpolation* (LERP) [27].

IV. Control Design

In this section, we present the most relevant aspects from [2] to specify robot behaviors that consider sensor input as well as robot and environment constraints. Furthermore, we introduce an extension by implementing the TFF to deal with complex tasks in which the robot tool is in contact with a workpiece. To this end, the robot control is expressed leveraging the definition of eTaSL soft-constraints, which are defined by *joint position variables* $\boldsymbol{q}$, *feature variables* $\boldsymbol{\chi}_{\boldsymbol{f}}$, and the *time variable* t. The introduction of feature variables in eTaSL allows the specification of DOF associated with the task such as the degree-of-advancement ξ or the latent variables $\boldsymbol{\chi}_{\boldsymbol{f}_{\text{lv}}}$ in the learned model. The constrained optimization problem in eTaSL takes the following form:

$$\underset{\boldsymbol{x}}{\text{minimize}} \qquad \boldsymbol{x}^T \boldsymbol{H} \boldsymbol{x} \tag{10a}$$

$$\text{subject to} \qquad \boldsymbol{L}_{\boldsymbol{A}} \leq \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{U}_{\boldsymbol{A}} \tag{10b}$$

where the argument $\boldsymbol{x}$ is a vector $\begin{bmatrix}\dot{\boldsymbol{q}}^T & \dot{\boldsymbol{\chi}}_f^T & \boldsymbol{\varepsilon}^T\end{bmatrix}^T$ that contains: the time derivative of the robot joint variables $\dot{\boldsymbol{q}}$, the time derivative of the feature variables $\dot{\boldsymbol{\chi}}_f$, and a slack variable $\boldsymbol{\varepsilon}$ for each soft-constraint. The *weights* w in the diagonal of $\boldsymbol{H}$ enable the system to deal with conflicting constraints. $\boldsymbol{L}_{\boldsymbol{A}}$ and $\boldsymbol{U}_{\boldsymbol{A}}$ are lower and upper bounds of the constraints described by matrix $\boldsymbol{A}$, respectively.

In [2] it is described how eTaSL enables the specification of constraints (10b) using three kinds of velocity-resolved controllers: first, commanding a task expression $\boldsymbol{e}(\boldsymbol{q}, \boldsymbol{\chi}_f, t)$ to evolve towards zero (11) following a first-order system with time constant k^{-1}; second, commanding a task expression $\boldsymbol{g}(\boldsymbol{q}, \boldsymbol{\chi}_f, t)$ to follow a desired velocity $\boldsymbol{v}$ (12); third, commanding a (sensor) measurement $\boldsymbol{h}$ to evolve towards a target $\boldsymbol{h}^*$ following a first-order system with time constant k^{-1}, by describing this evolution as a model $\boldsymbol{m}(\boldsymbol{q}, \boldsymbol{\chi}_f, t)$. These constraints are formulated as follows:

$$\boldsymbol{J}\begin{bmatrix}\dot{\boldsymbol{q}} \\ \dot{\boldsymbol{\chi}}_f\end{bmatrix} = -k\boldsymbol{e} - \frac{\partial \boldsymbol{e}}{\partial t} + \boldsymbol{\varepsilon}, \tag{11}$$

$$\boldsymbol{J}\begin{bmatrix}\dot{\boldsymbol{q}} \\ \dot{\boldsymbol{\chi}}_f\end{bmatrix} = \boldsymbol{v} - \frac{\partial \boldsymbol{g}}{\partial t} + \boldsymbol{\varepsilon}, \tag{12}$$

$$\boldsymbol{J}_{\boldsymbol{m}}\begin{bmatrix}\dot{\boldsymbol{q}} \\ \dot{\boldsymbol{\chi}}_f\end{bmatrix} = -k(\boldsymbol{h} - \boldsymbol{h}^*) - \frac{\partial \boldsymbol{m}}{\partial t} + \boldsymbol{\varepsilon}. \tag{13}$$

The task function Jacobians $\boldsymbol{J}$ and $\boldsymbol{J}_{\boldsymbol{m}}$ map the time derivatives of states $\boldsymbol{q}$ and $\boldsymbol{\chi}_f$ into the time derivatives of the task variables $\boldsymbol{e}$.

We define the following constraints to describe the robot behavior:

Constraint 1. *Motion and wrench signal generation:* The generation of new signals is done by first defining the latent variables $\boldsymbol{\chi}_{\boldsymbol{f}_{\text{lv}}}$ in (9) as eTaSL feature variables. Then, a constraint (11) is imposed such that at the instance in which the first contact is expected, $\boldsymbol{p}_f(\xi, \boldsymbol{\chi}_{\boldsymbol{f}_{\text{lv}}})|_{\xi=1}$ coincides with the target contact position $\boldsymbol{p}^* \in \mathbb{R}^3$ obtained from a 3D vision system. This behavior is formulated as:

$$\boldsymbol{e}_1 = \boldsymbol{p}_f(\xi, \boldsymbol{\chi}_{\boldsymbol{f}_{\text{lv}}})|_{\xi=1} - \boldsymbol{p}^*. \tag{14}$$

Constraint 2. *Reactive time evolution:* The evolution along the signals is driven by, first, defining the degree-of-advancement ξ as a feature variable; and then, commanding its time derivative to follow the generated profile $\dot{\xi}_f(\xi, \boldsymbol{\chi}_{\boldsymbol{f}_{\text{lv}}})$ by specifying constraint (12) with:

$$g_2 := \xi \quad \text{and} \quad v := \dot{\xi}_f(\xi, \boldsymbol{\chi}_{\boldsymbol{f}_{\text{lv}}}) \tag{15}$$

This results in a behavior in which the degree-of-advancement speed $\dot{\xi}$ can deviate from the generated profile due to the reactive behavior during the task execution.

Constraint 3. *Motion control:* The pose of a task frame in the robot $\boldsymbol{\Gamma}(\boldsymbol{q}) := \{\boldsymbol{p}(\boldsymbol{q}), \boldsymbol{R}(\boldsymbol{q})\}$ is commanded to follow a generated pose $\boldsymbol{\Gamma}_{\boldsymbol{f}}(\xi, \boldsymbol{\chi}_{f_{\text{lv}}}) := \{\boldsymbol{p}_f(\xi, \boldsymbol{\chi}_{\boldsymbol{f}_{\text{lv}}}), \boldsymbol{R}_f(\xi, \boldsymbol{\chi}_{\boldsymbol{f}_{\text{lv}}})\}$ by imposing the position evolution and orientation evolution using constraint (11) with:

$$\boldsymbol{e}_{3,p} := \boldsymbol{p}(\boldsymbol{q}) - \boldsymbol{p}_f(\xi, \boldsymbol{\chi}_{\boldsymbol{f}_{\text{lv}}}), \tag{16}$$

$$\boldsymbol{e}_{3,R} := \boldsymbol{\delta}(\boldsymbol{q}, \xi, \boldsymbol{\chi}_{f_{\text{lv}}}) \tag{17}$$

where $\boldsymbol{\delta}(\boldsymbol{q}, \xi, \boldsymbol{\chi}_{f_{\text{lv}}})$ corresponds to the *equivalent axis representation* [28] of $\boldsymbol{R}(\boldsymbol{q})\boldsymbol{R}_f(\xi, \boldsymbol{\chi}_{f_{\text{lv}}})^{-1}$.

Constraint 4. *Wrench control:* To influence the robot motion given the generated wrench profile, first, we define instantaneous changes in the position $\boldsymbol{p}(\boldsymbol{q})^\dagger \in \mathbb{R}^3$ and the *equivalent axis representation* of the orientation $\boldsymbol{\phi}(\boldsymbol{q})^\dagger \in \mathbb{R}^3$ measured with respect to an instantaneous coinciding task frame. Then, we constrain their time derivative to be proportional to the difference between a target wrench $\boldsymbol{u}_f \in \mathbb{R}^6$ and a measured wrench $\boldsymbol{u} \in \mathbb{R}^6$. We introduce this behavior using constraint (13).

$$\boldsymbol{m}_4 := \begin{bmatrix}\boldsymbol{p}(\boldsymbol{q})^\dagger \\ \boldsymbol{\phi}(\boldsymbol{q})^\dagger\end{bmatrix}, \tag{18}$$

$$\boldsymbol{h}_4 := -\boldsymbol{K}_x^{-1}\boldsymbol{u}, \quad \text{and} \quad \boldsymbol{h}_4^* := -\boldsymbol{K}_x^{-1}\boldsymbol{u}_f, \tag{19}$$

where the stiffness between the tool and the environment is represented by a matrix with stiffness coefficients for translation $\boldsymbol{k}_p \in \mathbb{R}^{3\times 1}$ and orientation $\boldsymbol{k}_\delta \in \mathbb{R}^{3\times 1}$ in its diagonal $\boldsymbol{K}_x := \text{diag}(k_{p_x}, k_{p_y}, k_{p_z}, k_{\delta\text{x}}, k_{\delta\text{y}}, k_{\delta\text{z}})$.

Constraint 5. *Additional Constraints:* To increase safety in the system and enable human-robot interaction additional constraints were composed, offering proximity-based collision avoidance [6], workspace limits, and joint position and velocity limits [5] and [2].

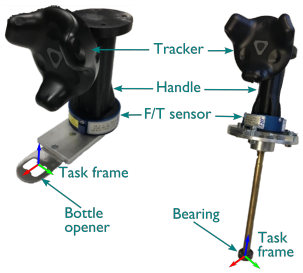

Fig. 2: A teaching tool equipped with a bottle opener *left* and a bearing to follow a contour *right*. The tool is instrumented with a tracker, a handle and an F/T sensor. The corresponding task frame is depicted for each tool.

TABLE I: Constraint parameters.

Constraint	Parameter	Value	Unit
1	k_1	15	s^{-1}
	w_1	1.25	
2	w_2	2	
3	k_3	4	s^{-1}
	w_3	1	
4	$k_{p_x}, k_{p_y}, k_{p_z}$	1800	N/m
	$k_{\delta x}, k_{\delta y}, k_{\delta z}$	180	Nm/rad
	k_4	4	s^{-1}
	w_4	$[0, 1]$	

V. Experimental Results

This section discusses two experimental use cases to validate our approach. Both experiments involve robot motions with and without contact while satisfying assumptions 1 to 4. However, both applications differ radically.

In both experiments, we leverage passive observation to record time-series of poses and wrenches from *five* demonstrations. To this end, we use a device equipped with an HTC-Vive tracker (accuracy up to $8\,\mathrm{mm}$) and a 6-axis JR3-67M F/T sensor (accuracy of $1\,\%$ of the measuring range, i.e., ± 200 N for forces and ± 12 Nm for moments). Both sensors are mounted as shown in Fig. 2.a.

Then, we subdivide the demonstrations in non-contact and contact segments using thresholds for the force and moment norms of 0.9 N and 0.01 Nm, respectively. To calculate these thresholds, we lower their value until obtaining false-positive contacts. To improve the F/T sensor's accuracy, we balance the sensor removing the wrench drift before every demonstration and execution. *Four* basis functions ($M = 4$) are chosen to encode the demonstrated information based on the variance encoded in each of them (see Fig. 3). The selection of this number is very intuitive: three is the least number to generalize at the position level and four is the maximum number given the small number of demonstrations. During the execution, the position of the contact point $\boldsymbol{p}^*$ is sensed using the commercial Pick-it 3D vision system with a reported accuracy of ± 5 mm.

The hyperparameters of the controller presented in section IV are reported in table I. These values were intuitively tuned since all of these parameters have a clear physical meaning, such as time constant k_i^{-1} for each constraint i of

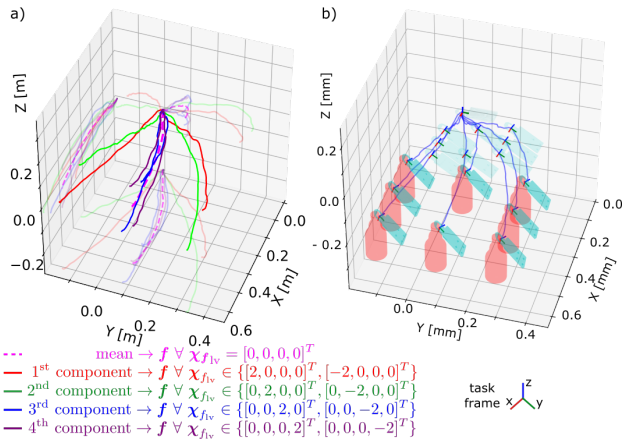

Fig. 3: Graphical representation of the position information captured in the learned model and its corresponding generated signals for the approach motion $0 \leq \xi \leq 1$ in the bottle-opening case. a) Variance in the position signals captured by the decoupled basis functions in the learned model (9). Their shaded projections in the planes are displayed. b) Position signals generated by a linear combination of the basis functions in a) for eight bottle positions.

type (11) or (13), or stiffness. When combining constraints, the robot behavior is influenced by the relation between the weights w_i for each constraint i. Therefore, they are specified according to the following intuitive guidelines:

1) k_1 has a high value to impose a fast convergence of constraint 1. w_1 has a slightly larger value than w_3 and w_4, thereby ensuring to reach the sensed point $\boldsymbol{p}^*$ while enabling deviations to deal with uncertainties of the vision system.
2) Similarly, w_2 is defined to command $\dot{\xi}$ to follow the generated degree-of-advancement speed signal $\dot{\xi}_f(\xi, \boldsymbol{\chi}_{f_{1v}})$ while allowing deviations to comply with the rest of the constraints.
3) w_3 is defined as a reference to tune w_1, w_2, and w_4.
4) w_4 is set to *zero* during the approach phase while increasing its value at a constant rate $1\ \mathrm{s}^{-1}$ until it reaches *one* at $\xi = 0.95$, just before the initial contact is expected. This behavior enables enough flexibility to deal with cases in which the initial contact occurs before expected. The maximum value of w_4 is designed to give constraint *4* the same priority as constraint *3* during the contact phase.
5) k_3 and k_4 are set considering that in velocity-resolved controllers affecting directly the motion of industrial robot gains are typically limited to 4 to $5\ \mathrm{s}^{-1}$.
6) Stiffness values $\boldsymbol{k}_p$ and $\boldsymbol{k}_\delta$ are estimated experimentally.

Constraints *1* and *2* correspond to task constraints. They do not affect the robot motion directly, but they do so by interacting with constraints *3* to *5*. Furthermore, the estimated stiffness is primarily a property of the contact interaction between the tool and the workpiece. However, compliance of the robot may reduce the overall stiffness. These insights enable tuning the parameters for a particular setup and then transferring them to other setups.

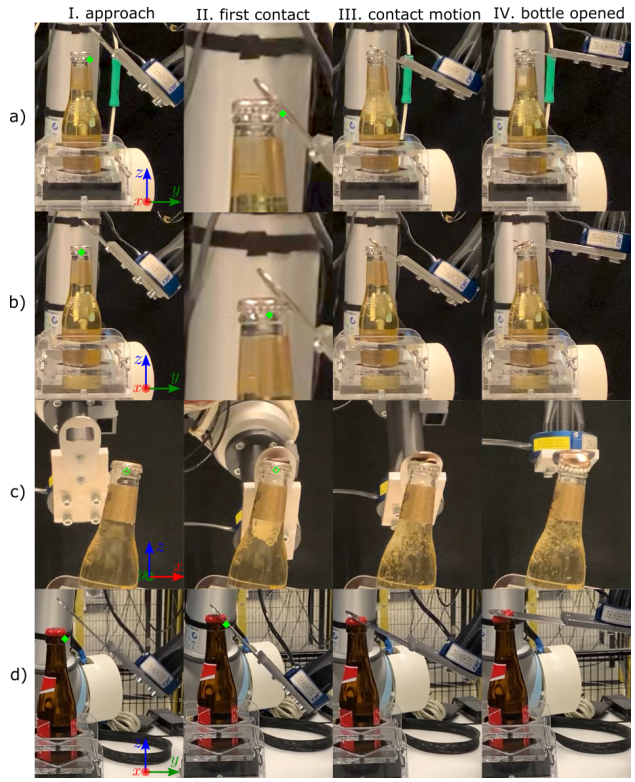

Fig. 4: Snapshots of four experiments to test the robustness of the system. The green rhombuses correspond to the expected contact point $\boldsymbol{p}^*$ provided to the system. a) Baseline experiment, the robot makes contact at the ground truth contact point. b) The expected contact point is intentionally shifted -5 mm in the y-axis with respect to the ground truth. c) The bottle is tilted 10° around the world y-axis. d) The degree-of-advancement speed and the tool length are increased 3.5 and 1.5 times, respectively.

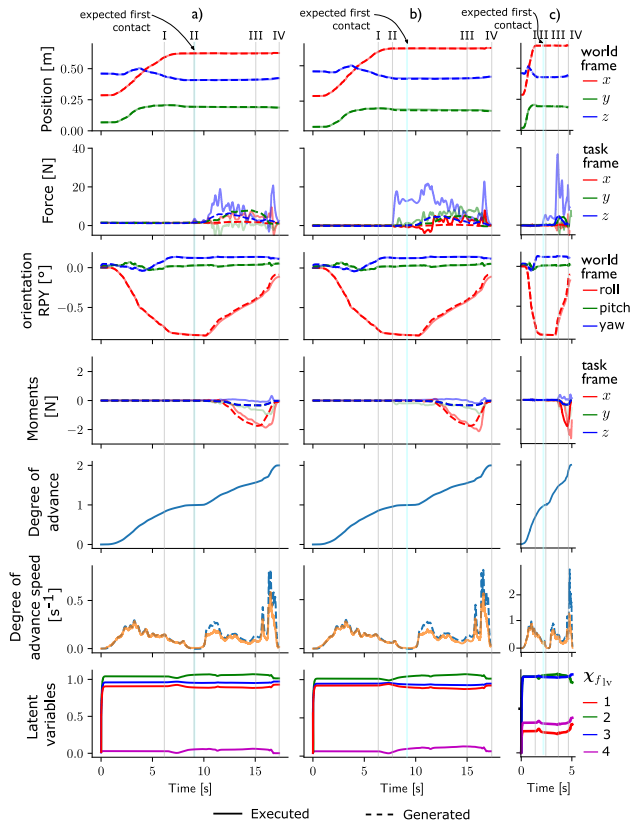

Fig. 5: Signals reported from successful experiments in the bottle-opening application by commanding robot motions: a) towards a ground truth point (see Fig. 4.a); b) towards a point shifted from the ground truth 5 mm in the y axis (see Fig. 4.b); and c) by increasing the degree-of-advancement speed and the tool length 3.5 and 1.5 times, respectively (see Fig. 4.d). The instances in roman numbers correspond to the snapshots in Fig. 4.

A. *First experiment: bottle-opening*

In the first experiment, a user demonstrates how to manipulate a bottle opener to remove a crown cork (see Fig. 1.a). To show the generalization of our system, we clamped bottles with different sizes oriented vertically in different positions. Slight variations in their orientations ($\pm 10^\circ$ measured with respect to the normal to the table) were intentionally introduced.

Fig. 3 shows how our approach linearly combines the learned basis functions (Fig. 3.a) to generate new trajectories towards non-demonstrated bottle positions (see Fig. 3.b). As a result, for a new bottle location, the opener is aligned to hook the cap while avoiding collisions with the bottle during the approach motion.

We tested the robustness of our system by establishing a baseline in which we commanded the robot to follow motion-wrench signals generated towards a ground truth contact point (see Fig. 4.a). Subsequently, we did experiments commanding the robot to follow motion-wrench signals generated towards positions shifted $\pm$ 3, 5 and 8 mm in the world x, y and z axis from the ground truth. For instance, Fig. 4.a depicts an experiment in which the expected contact point is shifted -5 mm in the y axis with respect to the ground truth. During execution, the robot got stuck at one side of the cap. The generated wrench controller enabled the system to recover and successfully complete the task.

We performed additional experiments to prove robustness against changes in the orientation of the bottle. To this end, the bottle was tilted $\pm$ 5°, 10° and 15° around the world x, and y axis (see Fig.4.c). The results of the experiments are reported in table II. Finally, we performed experiments in which we extended the length of the bottle opener used to record demonstrations 1.5 times (see Fig.4). We modified the kinematic specification for the task frame and the inertial properties of the tool accordingly. Subsequently, we sped up the task execution by multiplying the generated degree-of-advance speed $\dot{\xi}_f$ by a factor of 3.5 while *maintaining* the learned model, control specification and hyperparameters reported in table I.

Fig. 5 depicts a comparison between the mentioned experiments. Position, orientation, and moment signals were similar in the three cases. However, reported force deviations are high in b) and c) due to the misalignment of the expected contact point and the higher speed, respectively. In a) and c) the initial contact happens at the expected instance $\xi = 1$, while in experiment b) it occurs before the expected instance $\xi < 1$ (see first contact in Fig.4). The latent variables in Fig 5 remain constant around the same value when all constraints are satisfied, i.e., the executed motion corresponds to the

TABLE II: Successful (✓) and failed (×) experiments of the bottle-opening case when shifting positions and orientations from a baseline

mm	x-axis	y-axis	z-axis	°	x-axis	y-axis
$+3$	✓	✓	✓	$+5$	✓	✓
$+5$	✓	✓	✓	$+10$	×	✓
$+8$	×	×	×	$+15$	×	×
-3	✓	✓	✓	-5	✓	✓
-5	✓	✓	✓	-10	✓	✓
-8	×	×	✓	-15	×	×

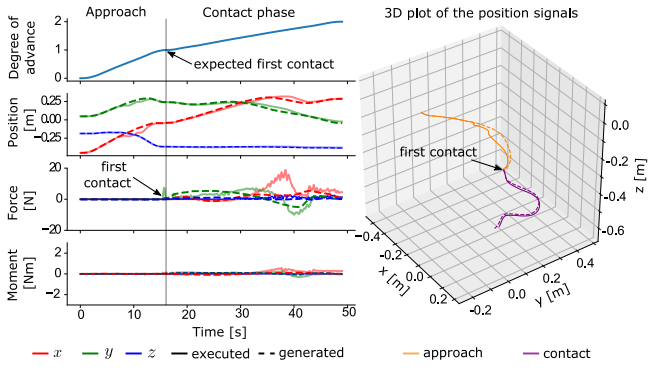

Fig. 6: Relevant signals reported for a contour tracking experiment. During the approach motion, $\boldsymbol{p}(\boldsymbol{q})$ deviates from the generated signals to avoid obstacles sensed by the proximity sensors. During the contact phase, the system is able to correct the deviations between the position of the real contour and the estimation of the learned model.

generated one. However, when there are deviations, their values change, adapting the generated signals in (9), and hence adding flexibility to the system. Additionally, deviations from the generated degree-of-advancement speed add flexibility to the system. As result, rapid motions encoded from demonstration, such as the one performed by the user immediately after removing the cap are supressed during execution, thereby complying with robot constraints (see after second 15 in Figs. 5.a and 5.b, and second 5 in Fig. 5.c),

B. Second experiment: contour-following

In a second experiment, a user demonstrates how to manipulate a tool with a bearing to make contact at a specific point of a workpiece and subsequently follow its contour (see Fig. 1). The workpiece is placed in different positions (with low variability in the orientation) lying on a rubber mat to increase friction with the table while still enabling displacement by exerting high forces. The desired first point of contact is perceived by the same vision system as in the first experiment. We executed this task by adapting the corresponding kinematic chain while *maintaining* the same control specification and hyperparameters as in table I.

Fig. 6 shows that the robot is able to avoid collisions with the hand of a person perceived by proximity sensors, while successfully completing the task (see Fig. 1.b). During the contact phase, the pose-wrench constraints enable the system to correct the motion of $\boldsymbol{p}(\boldsymbol{q})$ to follow the real contour, while maintaining contact.

Larger deviations between the generated and executed position position paths can be observed in Fig. 6 (30 - 40s). These deviations cause increments in the reported forces, however, without dragging the contour.

Finally, additional experiments were performed, in the bottle-opening and contour-following case, to compare the task execution resulting from the proposed motion-wrench constraints against a task execution only defined by the motion constraints, and another one only defined by the force constraints. Ten trials were performed for each of the three sets of constraints. Success in the first experiment corresponds to opening the bottle, while in the second it corresponds to follow at least 95 % of the contour. For these 10 trials the proposed methodology based on motion-wrench constraints showed 100 % success rate while the other constraint definitions only succeeded in 10 % of the trials. Results showed that by using only motion constraints in the contour-following, the robot exerted more than 20 N, thereby dragging the contour. In the bottle-opening case, the use of only motion or wrench constraints resulted in either missing the cap or activating the robot emergency stops due to the high forces. These insights confirm that the definition of only wrench or pose constraints is insufficient to achieve the proposed tasks robustly.

VI. Discussion

This paper combines LfD and constraint-based task specification to learn manipulation tasks that involve contact. Our assumptions fully describe a category of tasks which, besides covering the presented use cases, also encompasses other applications such as painting, polishing, among others.

The acquisition of motion and wrench information via passive observation facilitates a non-expert user to perform demonstrations intuitively without affecting the measurements compared to kinesthetic teaching. However, as future work, we propose to improve the human interface by decreasing the size of the pose and F/T sensors. In this way, we will be able to further improve the ergonomy of the teaching device.

Our method alleviates the curse of dimensionality affecting machine learning. This characteristic is attributed to

(i) the representation of the time evolution by reparameterizing signals and adding its time-derivative as one signal to the model instead of six as used in the state of art; and *(ii)* the use of the dimensionality reduction technique PPCA to capture the most meaningful information. Linearity in the learned space increases the predictability of our system, thereby improving robustness and safety in a human-robot collaborative scenario. Generalization of behaviors that cannot be captured due to the linearity assumption, e.g., circumventing obstacles by multiple sides, can be included by either specifying them in eTaSL or encoding them using a mixture of PPCA models [13] at the expense of increasing the number of demonstrations.

The use of passive observation and the implementation of the TFF using pose, wrench, and evolution constraints enabled the decoupling of the task and the robot platform. We proved that our system was able to execute the task by modifying the kinematic chain while maintaining the learned model, control specification, and hyperparameters. As future work, we will transfer the same task to other platforms.

Our system can generalize motion and wrench signals towards non-demonstrated positions while being robust against uncertainties associated with the vision system, small variations in workpiece orientation, and large variations in execution speed. As future work, we propose to encode information of the angular velocity as in [26], [15] to improve the generalization in the orientation signals.

Future work will focus on relaxing several of the assumptions and extending the robustness to other types of variations, allowing a rapid generalization for a wider range of applications.

VII. Conclusions

A learning approach was proposed for a specific category of tasks involving contact and non-contact motions. This category of tasks involves positions, orientations, forces, moments, a sensor-skin and a camera. Assumptions related to this category were made explicit. Learning from demonstration was applied to the position, orientation, force and moment signals and integrated with the rest of the task specification using the task specification language eTaSL. We succeeded in learning two radically different applications, a bottle-opening and a contour-following application, from a limited number of demonstrations. Although not explicitly demonstrated in this letter, we are independent of the specific robot used, as long as the kinematics and robot-related constraints are correctly modeled and in range for the given task.

As a result, our system has the potential to decrease the costs associated with the required hardware to deploy robot applications in industry. This aspect is essential to bridge the gap between LfD and real industrial applications.

References

[1] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent Advances in Robot Learning from Demonstration," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 13.1–13.34, 2020.

[2] C. A. Vergara, J. De Schutter, and E. Aertbeliën, "Combining imitation learning with constraint-based task specification and control," *IEEE Robot. and Autom. Lett.*, vol. 4, no. 2, pp. 1892–1899, April 2019.

[3] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[4] H. Bruyninckx and J. De Schutter, "Specification of force-controlled actions in the "Task Frame Formalism": A synthesis," *IEEE Trans. Robot.*, vol. 12, no. 4, pp. 581–589, 1996.

[5] E. Aertbeliën and J. De Schutter, "ETaSL/eTC: A constraint-based task specification language and robot controller using expression graphs," in *Proc. IEEE/RSJ Int. Conf. Int. Robots and Systems*, 2014, pp. 1540–1546.

[6] C. A. Vergara, G. Borghesan, E. Aertbeliën, and J. De Schutter, "Incorporating artificial skin signals in the constraint-based reactive control of human-robot collaborative manipulation tasks," *Industrial Robot*, vol. 46, no. 3, pp. 360–368, 2019.

[7] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in neural information processing systems*, 2013, pp. 2616–2624.

[8] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.

[9] A. Shukla and A. Billard, "Coupled dynamical system based arm–hand grasping model for learning fast adaptation strategies," *Rob. Auton. Systems*, vol. 60, no. 3, pp. 424–440, mar 2012.

[10] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent Service Robotics*, vol. 9, no. 1, pp. 1–29, 2016.

[11] M. J. A. Zeestraten, I. Havoutis, J. Silverio, S. Calinon, and D. G. Caldwell, "An Approach for Imitation Learning on Riemannian Manifolds," *IEEE Robot. and Autom. Lett.*, vol. 2, no. 3, pp. 1240–1247, 2017.

[12] L. P. Ureche and A. Billard, "Constraints extraction from asymmetrical bimanual tasks and their use in coordinated behavior," *Rob. Auton. Systems*, vol. 103, pp. 222–235, may 2018.

[13] C. M. Bishop, *Pattern Recognition and Machine Learning*, 2006.

[14] A. Kramberger, A. Gams, B. Nemec, and A. Ude, "Generalization of orientational motion in unit quaternion space," in *IEEE Int. Conf. Hum. Rob.*, 2016, pp. 808–813.

[15] A. Kramberger, A. Gams, B. Nemec, D. Chrysostomou, O. Madsen, and A. Ude, "Generalization of orientation trajectories and force-torque profiles for robotic assembly," *Rob. Auton. Systems*, vol. 98, pp. 333–346, dec 2017.

[16] B. Nemec, L. Žlajpah, S. Šlajpa, J. Piškur, and A. Ude, "An efficient pbd framework for fast deployment of bi-manual assembly tasks," in *IEEE Int. Conf. Hum. Rob.*, nov 2018, pp. 166–173.

[17] L. Rozo, D. Bruno, S. Calinon, and D. G. Caldwell, "Learning optimal controllers in human-robot cooperative transportation tasks with position and force constraints," in *Proc. IEEE/RSJ Int. Conf. Int. Robots and Systems*, vol. 2015-December, 2015, pp. 1024–1030.

[18] J. Silverio, Y. Huang, L. Rozo, S. Calinon, and D. G. Caldwell, "Probabilistic learning of torque controllers from kinematic and force constraints," in *Proc. IEEE/RSJ Int. Conf. Int. Robots and Systems*, 2018, pp. 6552–6559.

[19] A. X. Lee, H. Lu, A. Gupta, S. Levine, and P. Abbeel, "Learning force-based manipulation of deformable objects from multiple demonstrations," in *Proc. IEEE Int. Conf. Robotics and Automation*, no. June, 2015, pp. 177–184.

[20] A. K. Tanwani, J. Lee, B. Thananjeyan, M. Laskey, S. Krishnan, R. Fox, K. Goldberg, and S. Calinon, "Generalizing Robot Imitation Learning with Invariant Hidden Semi-Markov Models," in *International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, nov 2018.

[21] N. Figueroa and A. Billard, "Transform-Invariant Non-Parametric Clustering of Covariance Matrices and its Application to Unsupervised Joint Segmentation and Action Discovery," *arXiv preprint arXiv:1710.10060*, oct 2017.

[22] N. Figueroa, A. L. P. Ureche, and A. Billard, "Learning complex sequential tasks from demonstration: A pizza dough rolling case study," in *ACM/IEEE International Conference on Human-Robot Interaction*, vol. 2016-April, 2016, pp. 611–612.

[23] T. M. Hagos, M. Suomalainen, and V. Kyrki, "Segmenting and sequencing of compliant motions," in *IEEE Int. Conf. Intell. Robot. Syst.*, oct 2018, pp. 6057–6064.

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

[25] M. Racca, J. Pajarinen, A. Montebelli, and V. Kyrki, "Learning in-contact control strategies from demonstration," in *Proc. IEEE/RSJ Int. Conf. Int. Robots and Systems*, 2016, pp. 688–695.

[26] J. De Schutter, "Invariant Description of Rigid Body Motion Trajectories," *Journal of Mechanisms and Robotics*, vol. 2, no. 1, pp. 011 004/1–9, feb 2010.

[27] E. B. Dam, M. Koch, and M. Lillholm, "Quaternions , interpolation and animation," pp. Technical Report DIKU–TR–98/5,40–50, 1998.

[28] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. Boca Raton, FL: CRC Press, 1994.