Task Parameterization Using Continuous Constraints Extracted From Human Demonstrations

Ana Lucia Pais Ureche, Keisuke Umezawa, Yoshihiko Nakamura, *Fellow, IEEE*, and Aude Billard

***Abstract*—In this paper, we propose an approach for learning task specifications automatically, by observing human demonstrations. Using this approach allows a robot to combine representations of individual actions to achieve a high-level goal. We hypothesize that task specifications consist of variables that present a pattern of change that is invariant across demonstrations. We identify these specifications at different stages of task completion. Changes in task constraints allow us to identify transitions in the task description and to segment them into subtasks. We extract the following task-space constraints: 1) the reference frame in which to express the task variables; 2) the variable of interest at each time step, position, or force at the end effector; and 3) a factor that can modulate the contribution of force and position in a hybrid impedance controller. The approach was validated on a seven-degree-of-freedom Kuka arm, performing two different tasks: grating vegetables and extracting a battery from a charging stand.**

***Index Terms*—Constraint extraction, learning and adaptive systems, motion control, programming by demonstration (PbD).**

I. Introduction

DAILY activities such as dish washing or preparing a meal often require completing multiple actions while interacting with different objects. When performing such tasks, humans are able to focus on the key aspects necessary for achieving the goal. For example, when grating a vegetable, they naturally push against the grater and focus on maintaining a certain speed and contact force with the grating surface. Moreover, humans naturally introduce variability by repositioning objects or by using different paths between two objects.

Consequently, obtaining a feature-based representation for such high-level tasks requires:

1) relating these features to the objects in the task (extracting the local frame of reference);
2) accounting for the large variability between demonstrations and deciding what feature should be reproduced (extracting task constraints with respect to trajectories, force profiles, and necessary stiffness modulation).

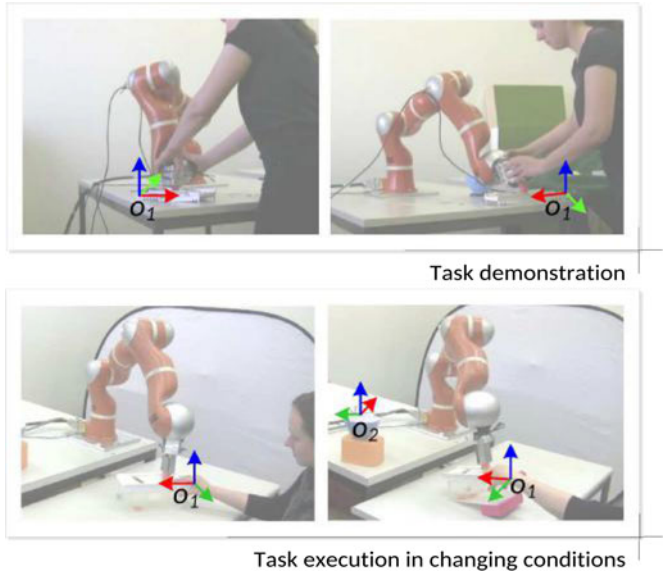

Fig. 1. From recording human demonstrations (top row), we detect the relevant frame of reference and the direction in which to apply a hybrid force and position controller. In this figure, the robot has correctly extracted that the frame of reference is attached to the grater and that force has to be applied along the vertical axis, whereas position control is needed along the horizontal plane of the grater. This allows the robot to reproduce the task even when the grater is moved to a different position and orientation (bottom row).

In this paper, we propose an approach for automatically extracting continuous task constraints required for successfully completing a task. We consider the task presented in Fig. 1, consisting of grating a vegetable and disposing the remains. We use programming by demonstration (PbD) to record a set of kinesthetic demonstrations while varying the initial positions of the robot and the spatial configuration of the objects used.

We use the demonstration data to extract the object to be used in each part of the task (either o_1 or o_2; see Fig. 1) and the way the task should be performed (i.e., alternating force and position control). More specifically, we consider a hybrid impedance controller

$$\tau = J^T\left(K(x - x_r) + F\right) \tag{1}$$

where $\tau \in \mathbb{R}^n$ is the joint control input for an n-degree-of-freedom (DOF) manipulator, and J is the Jacobian. The following variables $x_r \in \mathbb{R}^D$, the reference Cartesian position, $F \in \mathbb{R}^D$, the desired force, and $K \in \mathbb{R}^{D\times D}$, the stiffness matrix

Manuscript received March 18, 2015; revised August 31, 2015; accepted October 20, 2015. Date of publication December 2, 2015; date of current version December 2, 2015. This paper was recommended for publication by Associate Editor A. Saxena and Editor C. Torras upon evaluation of the reviewers' comments. This work was supported by the European Union Seventh Framework Programme FP7/2007-2013 under Grant 288533 ROBOHOW.COG.

A. L. P. Ureche and A. Billard are with the Laboratory of Learning Algorithms and Systems, École Polytechnique Fédérale de Lausanne, Lausanne 1015, Switzerland (e-mail: lucia.pais@epfl.ch; aude.billard@epfl.ch).

K. Umezawa was with the Graduate School of Information Science and Technology, University of Tokyo, Tokyo 113-8685, Japan. He is now with Bank of Tokyo Mitsubishi UFJ, Tokyo 100-8388, Japan (e-mail: keisuke.umezawa@gmail.com).

Y. Nakamura is with the Graduate School of Information Science and Technology, University of Tokyo, Tokyo 113-8685, Japan (e-mail: nakamura@ynl.t.u-tokyo.ac.jp).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2015.2495003

are extracted from the user demonstrations. In this case, the number of dimensions is $D = 3$.

We aim to learn a parameterization of this control law applicable for the whole task duration. Our approach exploits the variability between demonstrations to learn a criterion for determining a notion of coherence in the demonstration.

First, for each time step, we extract a reference frame R in which the variables are most consistent. In some cases, this may represent a quasi-orthogonal decomposition of position and force control along the axes of the object, although we continuously use a hybrid controller.

Second, we extract the variables of interest in the selected reference frame. Specifically, a task variable (such as the force perceived at the end effector) might have a large variability within a demonstration, thus indicating that it becomes important only in a given region of the task. Regions in which a variable changes very little throughout a set of sequential demonstrations prove coherency in that part of the task. Therefore, we focus on extracting such behaviors as the task constraints that should be reproduced.

Third, we extract the stiffness parameter K, which allows us to modulate the contribution of position and force when there is no decomposition as hybrid control as well as to ensure safe interaction and proper task completion.

These task specifications change when switching from one action to another. Typically, we record demonstrations of a full task, consisting of several such actions. Applying our method automatically segments the demonstration data. The grating task, for example, consists of three distinct phases: reaching for the grater, grating, and reaching for the trash can. Two reference frames are used (see Fig. 1): object o_1 (the grater) for the first two segments and object o_2 (the bowl) for the third part of the task.

Automatically obtaining this decomposition guides the learning phase of the PbD framework. A different model can be learnt for each atomic action (using various machine learning techniques for motion encoding), in the local frame of reference, using the data between two changes of constraints. In our approach, we learn from the demonstrations a time-invariant path profile for the directions along which position is the variable of interest (x_r). For the directions along which force is important, we learn a dependency between the desired force profile and the desired trajectory. For the particular task of grating vegetables, we obtain a decomposition of force and position control. This applies to the motion along the grater's surface where force control is performed (thus, F becomes a function of other variables, such as in (2), $F = f(x_1)$), while position is controlled on the other two axes, leading to

$$\tau = J^T R \begin{bmatrix} K_1(x_1 - x_{r_1}) \\ K_2(x_2 - x_{r_2}) \\ F \end{bmatrix}. \tag{2}$$

Our approach is aimed at bootstrapping information for learning and has the following contributions.

1) It automatizes learning, by bootstrapping information about the task, and parameterizing the learned models. It performs automatic task segmentation and reduces the number of the variables encoded for each segment by extracting the important ones. This simplifies the learned model by focusing only on the variables of interest resulted from this decomposition (e.g., instead of learning a full encoding of 3-D force versus 3-D position model, one simplifies the model by encoding just the force profile corresponding to the axis where this is applied).
2) It identifies task constraints directly from variables that can be used for control (end effector position, force, and stiffness) and offers a clear decomposition of these. This enables a consistent encoding of all the subtasks for using a single controller and ensures a smooth execution by directly embedding the constraints. This is applied through a Cartesian impedance controller, by modulating the stiffness (e.g., having zero stiffness on one axis is equivalent to performing pure force control on that axis). Therefore, we learn a stiffness modulation profile to be applied online during the execution.
3) The learned skill is generalizable to different locations or similar objects. This is achieved by learning the desired control with respect to the determined object frame (i.e., relating the action to the object on which this is performed). The system is robust to perturbations due to the time-invariant encoding.
4) It extracts task constraints without requiring any prior information about the goal of the task, actions in the task, or models of the objects.

Next, we review related work in Section II and describe the different stages of our task constraints extraction in Section III. We contrast the extraction of constraints for two tasks differing in the duration and number of important variables in Section IV. We discuss the advantages and limitations of our approach in Section V.

II. Related Work

In this paper, we focus on extracting artificial task constraints as described in [1] and [2], based on the variability observed in the demonstration data. The idea that invariants in motion determine important task features was first used by Bobick and Wilson [3] for recognizing gestures from continuous data and representing them as an enchainment of states. In our work, we use the variance not only to segment data but also to determine the relative importance between various variables and the frame in which these are most consistent. We reconstruct the task from a sequence of states, parameterized with the extracted constraints. Therefore, we review related work with respect to automatic extraction of constraints, task segmentation, and constraint-based motion planning.

A. Automatic Extraction of Reference Frames

In our previous work [4], we proposed extracting the reference frame in a manipulation task with respect to a proposed metric of imitation. Data recorded from demonstrations (arm joint angles, hand Cartesian position relative to the objects, and gripper status) are projected into a lower dimensionality latent space and further encoded in a time-dependent manner using a Gaussian mixture model (GMM). Gaussian mixture regression

(GMR) is used to reproduce the motion. In an early attempt, temporal variations are encoded in an hidden Markov model (HMM), and implicit segmentation is performed through HMM states [5]. These implementations have the limitations of encoding the motion in a time-dependent manner. Additionally, in our approach, we focus only on the end effector state (actual position and force, observed in the demonstration), thus making the skill easily transferable to other robotic platforms. Moreover, we increase the task complexity and the number of encoded constraints.

A different method of selecting a task space is based on three criteria [6]: a variance-based analysis of object trajectories, attention focus on objects in the task, and an evaluation of the teacher's discomfort during demonstration. While this method takes into account many factors, it is applied solely to vision-tracked human demonstrations. In our case, the demonstrations are performed kinesthetically in order to allow the robot to experience forces that should be applied on objects. Moreover, analyzing if the human maintains an uncomfortable posture during demonstration might reveal that the particular action was important for the task [6]. In our case, a direct evaluation is done on robot's proprioceptive data, while the user chooses an arbitrary position for demonstration.

The approaches mentioned above lack information about how the manipulation is performed that in some tasks may be key to successful execution. Therefore, we build on these existing approaches by extracting constraints with respect to force profiles and robot stiffness in different regions of the task and assess the effects this has on task completion.

Expressing the control variables in the local reference frame attached to the object on which manipulation is performed at a given time allows the robot to properly execute the task when the positions of the objects change in the scene. Moreover, this allows us to consider constraints not only as factors that limit the robot's motion [7], but that also add meaning to the motion (i.e., a grating motion, characterized by a given force and motion profile, is only meaningful when performed on a grater and in the context of a grating scenario).

In some cases, there might be multiple actions performed on the same object. The methods presented above extract one reference frame, but cannot disambiguate between the different positioning needed for each action. In our work, we address this issue by also extracting an attractor frame (relative to the reference frame extracted above).

B. Automatic Extraction of Force Information

The ability to successfully perform complex tasks resides in making use of additional sensing. For example, assessing joint torque values can be an indicator of whether the motion of the end effector is constrained [8]. Therefore, the second aspect that we address is detecting axes in task space where force control applies and encoding these force profiles. Typically, the decision of choosing an axis in task space on which to perform force control or position control is engineered in advance. In the proposed approach, we were able to automatically determine an arbitrary reference frame with respect to the object of interest in which a decomposition of force and position control can be obtained, and we selected the suitable type of control that applies to each axis. However, adding the force information, while of high importance for the task, can be challenging depending on the platform. Kinesthetic teaching for demonstrating the motion might need to be used in conjunction with a haptic device for demonstrating the required force profile [9].

Additionally, the stiffness is an important parameter when executing a task, as varying the robot's stiffness according to the task ensures safer interaction [10]. In our approach, we determine the required stiffness modulation as a relative measure between the contribution of force and position on each axis of the object. This leads to learning hybrid control in an automatically determined frame.

C. Task Segmentation

The constraint extraction topic is complementary to performing task segmentation which on the long term offers the possibility to easily recognize, classify, and reuse motions [11]–[13].

Typically, in robot learning from demonstration of a task that consists of several actions, each gesture is shown to the robot separately. The main reason is that task specifications change from action to action. In the proposed approach, we are able to automatically determine when these task specifications need to change and the next set of specifications.

In our work, we do not explicitly seek to segment the data; however, segmentation occurs naturally when the task constraints change, resulting in meaningful segments that encode atomic actions. This allows a flexible representation of the task, exploiting the local behavior in each subtask. A vast majority of recent works in segmentation focus solely on motion data represented by sets of joint positions or hand positions and orientation retrieved by motion capture systems in the case of human motion and by robots proprioception in the case of robotic motions. However, very few works focus on segmenting task data that includes force information.

The existing approaches for motion segmentation [11] rely on either 1) classification based on existing motion primitives used for prior training [14]–[16]; 2) looking for changes in a variable, like zero crossings [17]; or 3) clustering similar motions by means of unsupervised learning [18], [19]. The downside of these approaches is the need of prior task knowledge, which may be poor and incomplete in real-life situations. Moreover, they are sensitive to the variables encoded and have difficulties when applied to data such as force information where a large number of zero crossings may appear, making the encoding of motion primitives difficult.

The first approach for segmentation can ease robot control because of the existence of motion primitives. However, while it is safe to assume that human motions are likely to follow a specific pattern in a known context, rather than being random (as shown in [20]), a major drawback is the need to include prior knowledge. It also restricts the scope of segmentation by knowing what the task is about, such as segmenting motions used in robot-assisted surgery [15].

The second segmentation involves searching for zero velocity crossings [17] or other changes in a variable compared with a known state [8]. This approach is sensitive to the variables

encoded while one needs to find a way that would ensure optimal segmentation across all task dimensions. Regions of low variance have been alternatively used to determine segmentation points [21]. Furthermore, most of them rely on other techniques for human motion analysis, which include [11]: dynamic time warping (DTW) used in the temporal alignment of recorded data; or HMM for analyzing data that varies in time (such as hand movements sign language [22]). Additionally, when humans demonstrate a task to a robot, they may stop during the demonstration to rearrange an object or teach in a different manner. In these cases, the aforementioned approaches oversegment the data.

The third approach encompasses a more complex view of human motion, such as learning and clustering motion primitives in an incremental manner, from observing humans [18]. The method in [18] performs unsupervised segmentation based on motion encoded through an HMM. The obtained segments are clustered according to a measure of relative distance and organized in a tree structure. It encodes generic motions at the root, which gradually become more specialized close to the leaves. The algorithm allows us to change the model according to known primitives [23], and to use the same learned model not only for recognizing, but also for generating motions [16]. While being one of the most robust implementations to date, the approach lacks time independence in motion encoding.

These approaches, while efficient, have the shortcoming of being task specific and requiring a considerable amount of prior knowledge, which may be poor and incomplete in real-life situations. Thus, they achieve little generalization across a wide range of tasks. They also fail to model specific features of the motion, focusing mainly on changes in position. Moreover, these algorithms focus on extracting motion primitives, as opposed to learning a parameterization of a control system that remains the same all along the task, as in our approach. This allows learning and reproducing a task in a seamless manner.

Our approach departs from the aforementioned implementations by: 1) taking a broader view on the task and analyzing the motion also with respect to constraints that apply to forces and stiffness; 2) extracting task constraints from a low number of demonstrations, while removing the oversegmentation; 3) finding the relevant atomic actions in a task, without embedding any prior information about the goal of the task, nor models of the objects.

This makes the approach suitable for tasks that encompass switching between multiple atomic actions. Moreover, we consider continuous constraints that may apply throughout or only on a subpart of the task. Finally, we use a single controller throughout the task execution, while the constraints identify values taken by the variables of the impedance controller as the task unfolds.

D. Constraint-Based Motion Planning

Knowing the constraints that apply to each action that is to be performed can lead to a better task planning [24], [25]. A constraint-based representation of a complex task can be used by a high-level planner [26] for executing plans or for inferring motion grammars [27] for a high-level representation.

Common ways of encoding the task sequence use finite state machines (FSMs) [28], [29], Petri nets, Markov Models [21], [30], and graph and tree representations [31], [32].

In our work, we consider the sequence of atomic actions implicit in the demonstration. We, therefore, determine an FSM to execute the task. The states are not known *a priori* but extracted. They correspond to the atomic actions identified previously and encode their corresponding constraints. Our implementation takes a low-level approach by encoding constraints, directly in the control variables. This guarantees the task success without knowing the conceptual goal and allows isolating atomic actions for individual reuse. The task is executed using a single controller and embedding the constraints online, during the execution.

III. Method

We consider a set of N demonstrations of a task performed under changing conditions, using a number N_o of objects. The dataset is a vector of $L = 2$ components $\xi_d^i = \{F_d^i,\ x_d^i\}$ consisting of end effector measurements of force and position. The upper indices correspond to representing the data in the reference frame of each object o_i, $i \in 1 \dots N_o$, while the lower indices correspond to the dimensions considered $d = 1 \dots D$. The ξ^0 corresponds to the original recorded data (in R_0), the fixed referential in the base of the robot. The data were temporally aligned using DTW, resulting in a set of length T. Each demonstration is composed of a series of $T \cdot D \cdot L$ measurements, with $t = 1 \dots T$ number of time steps, $d = 1 \dots D$, dimension of each of the $l = 1 \dots L$ components.

We postulate that if a variable 1) changes value *significantly* within a single demonstration and 2) changes this value in a *systematic* way across demonstrations, then this variable is *significant* for the task. It, hence, becomes a task constraint that should be reproduced. We, thus, propose a criterion, computed for all variables $D \cdot L$ and all objects N_o, given by the difference between the variance over the time window and that over trials. This allows comparing the task variables in a relative manner, without setting any hard thresholds. At each time step, the criterion is computed on each dimension as

$$C\left(\xi_{d,l}^i\right) = \mathrm{Var}_{\mathrm{win}}\left(\xi_{d,l}^i\right) - \mathrm{Var}_{\mathrm{trial}}\left(\xi_{d,l}^i\right) \tag{3}$$

thus comparing the force and position measurements on each axis. The obtained value is normalized $C(\xi_{d,l}^i) \in [-1,\ 1]$. The variances $\mathrm{Var}_{\mathrm{trial}}$ and $\mathrm{Var}_{\mathrm{win}}$ are defined as

$$\mathrm{Var}_{\mathrm{trial}}\left(\xi_{d,l}^i\right) = \frac{1}{N}\sum_{i=1}^{N}\left(\mathrm{Var}(\xi_{d,l}^i)\right) \tag{4}$$

$$\mathrm{Var}_{\mathrm{win}}\left(\xi_{d,l}^i\big(t : t+\omega\big)\right) = \frac{1}{N}\sum_{i=1}^{N}\left(\mathrm{Var}(\xi_{d,l}^i(t : t+\omega))\right). \tag{5}$$

The values of the two variances are normalized such that $\mathrm{Var}_{\mathrm{trial}}, \mathrm{Var}_{\mathrm{win}} \in [-1, 1]$.

In a typical robotic task, a minimum of $D \cdot L$ variables have to be compared if using a 3-D measurement of position and force (three groups of two variables). The total number of criteria to

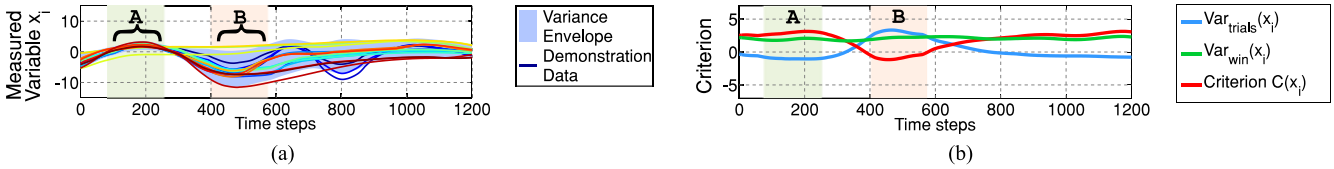

Fig. 2. Example of recorded data and computed variance over trials ($\mathrm{Var}_{\mathrm{trial}}$) and over a time window ($\mathrm{Var}_{\mathrm{win}}$) for a measured variable x_i. Region **A** shows data with little variance across trials (i.e., a feature of that should be reproduced). Region **B** shows data with large variance over trials, and low variance over a time window (almost constant). (a) Demonstration data for a measured variable. (b) Variance and computed Criterion.

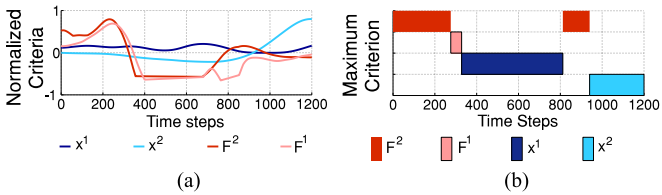

Fig. 3. Comparison between the criteria computed for unidimensional measurements of force (F) and position (x) in two reference frames (RF_1 and RF_2). (a) Criteria computed for force and position w.r.t two objects. (b) Maximum criterion at each time step.

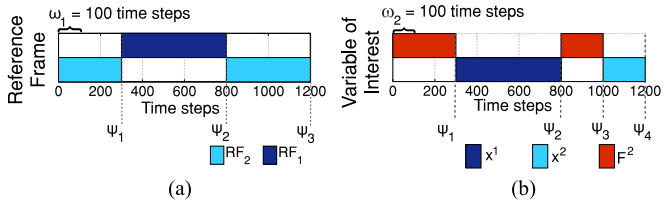

Fig. 4. Reference frame and variables of interest are given by the maximum criterion in a time window ω_i. (a) Extracted reference frame at each time step. (b) Extracted variable of interest at each time step.

be computed for a task is given by $N_C = N_o \cdot L \cdot D$. The size of the time window is an open parameter. In this case, it is chosen arbitrarily as being the shortest time period in which we see noticeable changes in the task flow.

The proposed method for extracting the task constraints is illustrated below, on an unidimensional measurement ($D = 1$) of two variables: force $F \in \mathbb{R}$ and Cartesian position $x \in \mathbb{R}$ of the robot's end effector. For the purpose of this example, we drop the lower index d. We also consider two objects o_1, o_2. The dataset is composed of the pair of elements: $\xi^i = \{F^i,\ x^i\}$ considered to be recorded over a number of N demonstrations of a task [see Fig. 2(a)]. This determines $N_C = 4$ computed criteria, as shown in Fig. 3(a).

A. Determining the Task Constraints

Using the defined criterion, we extract the following task constraints: the frame of reference (as explained in Section III-A1), the relative importance of position and force on each axis of the object (see Section III-A2), and a weighting factor between the two, used to modulate the controller's stiffness throughout the task (see Section III-A3). The procedure is summarized in Algorithm 1.

1) Extraction of the Reference Frame: For choosing a frame of reference, we compare the computed criteria and choose at each time index t, $t = 1 \ldots T$, the value of the highest criterion for all the variables considered $\max(C(\xi^i_{d,l}))$; see Fig. 3(b). Thus, the vector of obtained maximum values $\max(C(\xi^i_{d,l}))$ is analyzed separately for each dimension d, using a time window of arbitrary size (in this case $w_1 = 100$ time steps). We consider that in each time window, the reference frame is given by the object o with the highest number of occurrences of its corresponding criterion $\max(C(\xi^o_{d,l}))$. In this example, there are two changes of reference frame, as shown in Fig. 4(a): For the first 100 time steps, the reference frame R is given by object o_2, for the next 200 time steps, there is a change to o_1, and for the rest of the motion, the RF is changed to o_2.

Algorithm 1 Task Constraints Extraction

Bootstrapping(*Set of N demonstrations:* $\underset{1\to N}{\xi^i_{d,l}} = \{F^i_d, x^i_d\}$)
- Do DTW, dataset length T
- Criteria: $C(\xi^i_{d,l}) = Var_{win}(\xi^i_{d,l}) - Var_{trial}(\xi^i_{d,l})$
- $s = 0$ *% number of segmentation points*
- *% Determine the reference frame:*
- **for** $t = 1 : \omega_1 : T$ **do**
  - $R(t) = R_i$ for which $C_{max} = \underset{t:t+\omega_1}{max}\,(C(\xi^i_l))$
  - **if** $RF(t) \neq RF(t-1)$ **then**
    - $s = s + 1$; *% Create a new segmentation point*
    - $\psi_s = [t_s, R_i]$ *% add the current constraints*
  - **end if**
- **end for**
- *% Determine the variable of interest:*
- **for** each dimension $d = 1 : D$ **do**
  - **for** $t = 1 : \omega_2 : T$ **do**
    - add $\xi^i_{d,k}$ to the current constraints vector
    - $\psi_s = [t_s, R_i, \xi^i_{d,l}]$ for which $C_{max} = \underset{t:t+\omega_2}{max}\,(C(\xi^i_{d,l}))$
    - **if** $\xi^i_{d,l}(t) \neq \xi^i_{d,l}(t-1)$ **then**
      - *Insert a new segmentation point*
    - **end if**
  - **end for**
  - *% Determine the stiffness modulation factor:*
  - **for** each segment s **do**
    - add $\lambda_{d,s}(t) = C(\xi_{d,1}(t) - \xi_{d,2}(t))$ to the constraints vector
    - $\psi_s = [t_s, R_i, \xi^i_k, \lambda_{d,s}]$
  - **end for**
- **end for**
- **return** $\psi_{1:s}$

end

The changes in the reference frame determine a set of *segmentation points* ψ_s, $s = 1 \ldots S$, which delimit the actions performed on each object. In this example, there are three actions (one performed on object 1 and two performed on object 2) determined by the change of RF. Each segmentation point corresponds to a state that contains the time index t_s when the change occurred and the id of the reference frame used up to that point $\psi_s = [t_s, R_s]$.

2) Extraction of the Relevant Task Variables: The criterion defined in (3) allows us to compare in a relative manner the influence of variables of different types (like force versus position), and that vary across different scales [see Fig. 3(a)]. The aim is to be able to quantify their relevance with respect to the task so as to give more importance to the variable of interest in the controller and to adjust it when a change occurs.

For determining the relevant task variables, we analyze the criterion on each dimension d using a time window of arbitrary size (in this case, $w_2 = 100$ time steps). Similarly to extracting the reference frame, we consider the relevant variable in each time window to be the one that has the highest occurrence of its corresponding maximum criterion in that interval. In the given example, there are several changes between position and force as variables of interest [see Fig. 4(b)].

The changes in the variable of interest determine additional *segmentation points* which together with the initial points determined by the change of reference frame delimit individual atomic actions such as reaching movements. In the example described above, there are three segmentation points corresponding to the change of the variable of interest [see Fig. 4(b)]. The first two points are identical to the segmentation points ψ_1 and ψ_2 found by the change in the reference frame. The next point ψ_3 marks a change from a force-based part of the task to a position based part. The final point ψ_4 concludes the motion. The points are sorted according to the time index when the segmentation occurred. The information about the variable of interest is added to the vector $\psi_s = [t_s, R_s, \xi^s_{d,l}]$. The current $\xi^s_{d,l}$ now contains only the data between the previous and current segmentation points.

3) Extraction of the Stiffness Modulation Factor: Determining the axis-specific relative importance between the two variables can be done by computing a weighting factor λ that balances the contribution of the force and position according to the relevance determined above. Thus, for each dimension d, the value of $\lambda_d \in \mathbb{R}^D$ is given by the normalized difference between the criterion computed for position and the one computed for force

$$\lambda_d = C(x_d) - C(F_d). \tag{6}$$

Thus, the value of $\lambda \in [0, 1]$ becomes a weighting factor for the controller's stiffness K. Therefore, we can use an impedance controller for reproducing the motion with the factors described above representing continuous constraints, which can be directly embedded in the robot's control

$$\tau = J^T \cdot R \cdot (\lambda K(x - x_r) + F). \tag{7}$$

The corresponding λ profile for each segment of the motion is added to the constraints vector $\psi_s = [t_s, RF_s, \xi^s_{d,k}, \lambda_s]$

4) Choice of Time-Window Size: In the example presented above, the size of the time window was chosen arbitrarily. When performing manual tuning, our aim was to determine a time window that would result in avoiding very sudden changes from an important variable to another. For example, switching from force control to position control for less than 10 ms will not have an effect on the task.

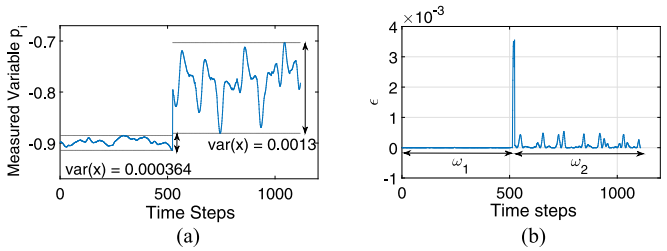

Fig. 5. Automatically adjusting the window size based on monitoring changes in the observed variance.

TABLE I
FINAL TASK PARAMETERIZATION FOR THE GIVEN EXAMPLE, CONSISTING OF STATES ψ_s, THE EXTRACTED CONSTRAINTS, AND THE CORRESPONDING STATISTICAL ENCODING TO BE USED BY THE CONTROLLER IN EACH SEGMENT, C_{ψ_s}

State	Constraints	Motion Encoding
ψ_1	$[t_{s_1}, R_1, F, \lambda_1]$	$C_{\psi_1} = [C_x, \theta^k_F, \theta^k_\lambda]$
ψ_2	$[t_{s_2}, R_2, x, \lambda_2]$	$C_{\psi_2} = [C_x, \theta^k_\lambda]$
ψ_3	$[t_{s_3}, R_2, F, \lambda_3]$	$C_{\psi_3} = [C_x, \theta^k_F, \theta^k_\lambda]$
ψ_4	$[t_{s_4}, R_1, x, \lambda_4]$	$C_{\psi_4} = [C_x, \theta^k_\lambda]$

However, a variable time window is desirable. We propose a way of determining a suitable time window by comparing the average variance with the instantaneous variance, therefore monitoring the rate of change in the signal. For example, in the signal presented in Fig. 5(a), choosing a large time window (e.g., 1200 time steps) leads to loosing information because the average variance in the first part of the signal is different and not representative for the second part of the signal. Therefore, the average variance in a local time window (var_{tw}) should be similar to the instantaneous variance computed at each time step (var_{ts}). A change in the average variance determines a step change in the instantaneous value. Therefore, we can compute a suitable time window using a variable ϵ defined as: $\epsilon = |\text{var}_{\text{tw}} - \text{var}_{\text{ts}}|$. A significant change in this measure determines starting a new window. According to this variable [see Fig. 5(b)], we were able to determine two windows ($\omega_1 = 513$ samples and $\omega_2 = 595$ samples) for the given example, based on an abrupt change in ϵ.

B. Constraint-Based Motion Learning

In our work, segmentation of the demonstrated data occurs implicitly whenever there is a change in the extracted task constraints. This is a natural manner of segmenting as the points in which either the reference frame or the variables of interest change, delimit atomic actions (e.g., the force sensed at the end effector might be relevant in the first part of the task, while after the segmentation point, end effector's position could become more relevant). Segmenting and interpreting the data in a stochastic manner allows regenerating the motion according to the measures determined to be important as well as finding optimal control strategies with respect to the variables of interest (see Table I], Columns 1 and 2).

When encoding the motion profile, we aim to preserve the exact behavior seen during demonstration. We, therefore, choose to encode variables that show a temporal coupling [like position and orientation, that change synchronously toward a target posture (*the attractor*)] using our coupled dynamical systems (CDS) approach [33]. This encompasses the following advantages: 1) the motion is encoded in a time-invariant manner and ensures asymptotical stability at the target of both dynamical systems; 2) the motion follows the demonstrated dynamics even if the execution starts from unknown regions of the space, far from the demonstrated motion, without the need to replan or rescale the trajectory; 3) the temporal-correlated behavior of the two variables is preserved, and thus, a perturbation in one of the systems does not cause an unsynchronized behavior, the robot being able to adapt online to changes in the environment.

With respect to a given reference frame R extracted previously, the CDS approach determines an attractor (a relative positioning and learns the motion profile with respect to this frame). In the given example, there are two attractors with respect to the grater object and one for the bowl.

1) Learning the Motion Profile: We choose to encode the motion using a CDS approach, as described in [33], which allows us to preserve the coupled evolution of position and orientation toward the target posture, that was observed in the demonstrations. The force profile is encoded separately, as a function of the position. This allows the robot to execute the task in changing conditions and to generalize to situations not seen during training (see Fig. 1).

Each individual variable is encoded as a nonlinear dynamical system of the form $\dot{x} = f(x)$, which encodes the mapping between a variable and its first derivative thus removing the explicit time dependence. Here, x and $\dot{x} \in \mathbb{R}^D$ represent the Cartesian position and velocity of the end effector. The function $f : \mathbb{R}^D \mapsto \mathbb{R}^D$ (initially unknown, but implicit in the demonstrated behavior) is a continuous and continuously differentiable function stable only at the attractor $\overset{*}{x}$. The nonlinear behavior of function f is encoded using a mixture of k Gaussians, specified by a vector $\theta_x^k = [\pi_x^k, \mu_x^k, \Sigma_x^k]$, representing the parameters of the GMMs (priors, means, covariance matrices), such that $P(x, \dot{x}|\theta_x^k)$ represents the dynamics of system 1. Based on this encoding, the velocity $\dot{x}$ is thus computed as $\dot{x} = E\{p(\dot{x}|x; \theta_x^k)\}$. The model is learned through maximization of likelihood under stability constraints (see [33] for details).

In our case, the absolute position of the attractor in each segment is estimated from the initial set ξ_0 (in R_0) as the average of all the points from the N demonstrations, on each dimension d, at the segmentation time t_s, resulting in: $\overset{*}{x}_d = \underset{1 \to N}{\text{avg}}\,(x_d(t_s))$. The motion is encoded in the attractor's reference frame R_*, such that the attractor becomes $\overset{*}{x} = 0$. The axes of the attractor's reference frame are not necessarily aligned with those of R_i, and the origin is located at $\overset{*}{x}$. In a grating task, for example, there are two attractors with respect to the grating surface: the *top* (initial point touched on the grater) and the *bottom* (after passing the blade).

Similarly, we encode the rotation specified by an axis-angle representation $r \in \mathbb{R}^4$, as $P(r, \dot{r}|\theta_r^k)$, with respect to an estimated attractor $\overset{*}{r}$. Finally, $P(\gamma(x), r|\theta_c^k)$ represents a coupling function between the two systems, learned using maximization of likelihood. During the execution, the system updates the dynamics of system 1 through GMR; second, the coupling is updated, and this determines updating the second system (in this case, the orientation) (see Algorithm 2).

Algorithm 2 Constraint–based task execution

```
FSM Execution(ψ_i, C_{ψ_i}, i = 1 : s)
  do
  read robot current position ξ_{d,1} and EE force ξ_{d,2}
  read objects positions
  for all task segments s do
    Use current state's constraints ψ_s = [t_s, R_s, ξ_l^d, λ_s]
    Transform data to R_s
     % Compute the next desired robot position {x(t+1), r(t+1)},
     % using CDS [33]
    if current attractor x*, r* not reached then
        % Compute next end effector position
            ẋ = E{p(ẋ|x; θ_x^k)};
            x(t+1) = x(t) + ẋ(t)Δt
        % Infer orientation based on current position
            r̃ = E{p(r|γ(x); θ_c^k)};
        % Compute next end effector orientation
            ṙ = E{p(ṙ|β(r − r̃); θ_r^k)};
            r(t+1) = r(t) + αṙ(t)Δt
         % Determine stiffness modulation based on current position
            λ = E{p(λ|x)}
        if Force is important on dimension d then
            % Predict force based on current position
                F = E{p(F|x)}
        end if
        Transform all data back to RF_0
        Update robot's motion (according to eq. 7)
    end if
    Else Go to the next state
  end for
  until Task completed
end
```

The model can be further parameterized to control the speed and amplitude of the robot's behavior under perturbation, using two scalars α and β. While in the original implementation in [33], these parameters are learned from recording good trials and perturbed demonstrations; here, we can estimate them based on the variance information, such that in regions with high variability, the adaptation is slower than in regions with low variability. Thus, in the proposed impedance controller, the reference trajectory for the reaching segments is given by the learned CDS model. This ensures that the learned model follows the original dynamics of the demonstrated motion, and it is stable at the target. The synchronous evolution is ensured through the coupling function. The complete CDS encoding of the motion in a subpart of the task is thus specified by the vector: $C_x = [\theta_x^k, \theta_r^k, \theta_\xi^k, \overset{*}{x}, \overset{*}{r}, \alpha, \beta]$.

2) Learning the Force Profile: For segments of the task, and across dimensions in which the force becomes important, we use GMM to learn a joint distribution of the variables F and x. We choose to encode the force profile with respect to the axis in which we see noticeable changes in position. In the grating task, for example, force control is performed along the *Z*-axis of the object; there is no modification in position along the *Y*-axis (i.e., along the width of the grater, but the highest variance is observed with respect to the motion along the *X*-axis (the grater's length),

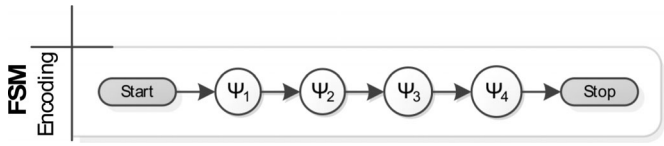

Fig. 6. FSM used for executing the task. Each state encodes the determined constraints. We consider that the order of the demonstrated actions is implicit for the task flow.

which therefore becomes the variable with respect to which we encode the force profile.

We use a model comprising a mixture of K Gaussian components, such that: $p(F,x) = \sum_{k=1}^{K}(\pi_F^k \cdot p(F,x;\mu_F^k,\Sigma_F^k))$, where π_F^k, μ_F^k, and Σ_F^k represent the priors, the mean, and the covariance matrix for the Gaussian model. These parameters are learned through Expectation-Maximization algorithm. The vector $\theta_F^k = [\pi_F^k, \mu_F^k, \Sigma_F^k]$ is added to the $C_{\psi_s} = [\theta_F^k]$. During the execution, GMR is used for predicting the force to be applied based on the current position: $E\{p(F|x)\}$. Unlike the encoding of position, for the force, there is no attractor, as force control is performed along a trajectory.

3) Learning the Stiffness Profile: We encode the stiffness modulation factor λ similarly to encoding the force, by learning a joint distribution $p(\lambda, x)$ using a mixture of k Gaussians. The model is parameterized by the vector $\theta_\lambda^k = [\pi_\lambda^k, \mu_\lambda^k, \Sigma_\lambda^k]$, representing the priors, means, and covariance matrices.

C. Constraint-Based Execution

We assume that the flow of atomic actions is implicit in the demonstration; thus, the reproduction is based on the determined sequence of $\psi_{1:S}$ points. An FSM containing the inferred states is generated, as shown in Fig. 6. A state is generated for each change of constraints and contains: 1) the extracted constraints, and 2) the learned motion models, as they are summarized in Table I, Column 3.

Typically, the transition between states occurs when the attractor of the current state is reached. This implies that reaching the determined relative frame is the main factor for advancing the execution. However, we make the assumption that the variables that were not determined as important for control might still hold complementary information, useful for state transitioning. The execution of the task based on the extracted constraints is presented in Algorithm 2.

IV. Robot Experiments

This approach was validated on two robot experiments performed using a 7-DOF KUKA Light Weight Robot arm, with the provided Cartesian Impedance controller. The controller takes as parameters the desired position, force, and stiffness, and it automatically adjust the damping and dynamics terms for stability. The two experiments consisting of a kitchen task, *grating vegetables*, and an office task *removing a battery from a charging stand*, differ in duration, number of variables used for control, and objects involved. We performed a quantitative evaluation of the extracted constraints with respect to the learned models, and a qualitative assessment with respect to the task performance.

A. Carrot Grating: Task Description

The task consisted of several atomic actions, presented in Fig. 7: reaching from the initial position to the slicer (the motion takes around 3–5 s), a repetitive slicing motion (on average, around 30 s), a reaching motion from the slicer to the trashing container (on average, 2 s).

Two objects were used: a grater (o_1) and a bowl (o_2). Data were recorded from the robot at 100 Hz, using kinesthetic demonstration and consisting of end effector position $x \in \mathbb{R}^3$ and orientation ($r \in \mathbb{R}^{3\times3}$), and external forces estimated at the end effector ($F \in \mathbb{R}^3$). The objects were tracked at 100 Hz using an OptiTrack motion capture system.

The variability of the task consisted in: 1) starting each demonstration from a different initial position of the robot, and placing the objects in different positions in the reachable space of the robot (we recorded data for three different positions of the objects, placed on average 30, 45, and 65 cm apart from the initial position); 2) using vegetables of different sizes and types (we recorded data for three types of vegetables (carrots, zukinis, and cucumbers). The vegetables varied in length, from a minimum of 10 cm for a carrot to a maximum of 35 cm for a cucumber, and with about 2 cm in diameter); the variability of the manipulated object affected the force applied by the user when providing demonstrations and the duration of the demonstration. The task lasted until the vegetable was fully grated; 3) inherent user variability between demonstrations. A total of $N = 18$ demonstrations were recorded, six for each vegetable type, using three different objects poses.

1) Extracted Constraints: For extracting the task constraints, we evaluated the 3-D measurements of position and force projected in the reference frame of each object. Following the approach described in Section III, the criterion on each axis was evaluated in a time window of width $w = 200$ time steps (2 s) for determining the reference frame. This resulted in one segmentation point. The motions of reaching and grating were expressed in the reference frame of object 1, the grater, and the motion of reaching the trash container was expressed in the reference frame of object 2, the bowl.

Similarly, we evaluated the criterion on each dimension, using a time window of width $w = 300$ time steps (3 s) for determining the variable of interest. The results showed that the force on the vertical axis became important in the second part of the task (grating and trashing), while only position was important in the first part of the motion (corresponding to reaching the grater). The change in the variable of interest determined a new segmentation point. A final point concludes the motion. Therefore, three segmentation points ψ_s were determined for this task (see Fig. 8), involving the three different states.

Two attractors were determined relative to the grater: one near the handle (*Grater top*) and one at the bottom of the grater (*Grater bottom*), after passing over the blade. Similarly, at the end of the motion, the positioning was relative to the trashing bowl. This allowed us to have an attractor-based encoding of

Fig. 7. Atomic actions in the *Vegetable Grating Task*. The user demonstrates the task, using different starting configurations of the objects and the robot. (a) Reaching the grater. (b) Grating the whole vegetable. (c) Trashing the remains.

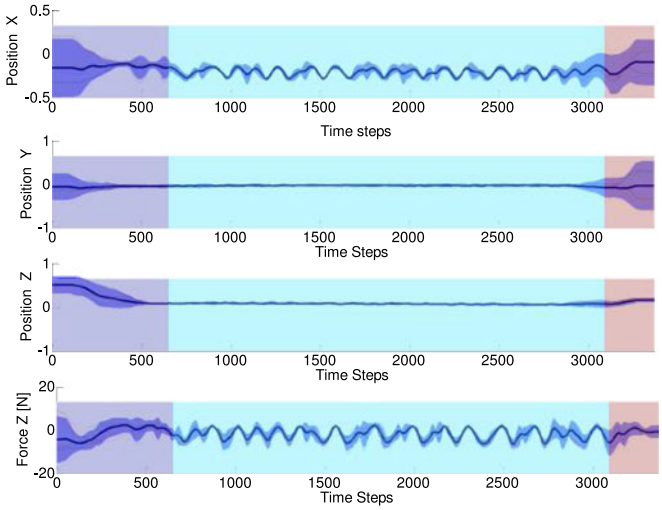

Fig. 8. Obtained segmentation overlapped on the demonstration data.

the task. The learned dynamics for reaching the grater o_1 and the trash o_2, respectively, are shown in Fig. 9(a) and (b), with generalization across different starting postures. Generalization with respect to a moving target is shown in Fig. 9(c); the force and stiffness modulation are presented in Fig. 10.

An FSM was generated as described in Section III-C. The advancement of the FSM happened when the current attractor was reached, or when the number of grating passes was completed. For evaluation purposes, the number of times the grating was performed was an additional condition for the transition between states ψ_1 and ψ_2.

2) Task Evaluation: We performed both a qualitative and quantitative assessments and evaluated 1) the correct extraction of task constraints; and 2) the ability of the system to generalize to new object locations and different vegetables.

a) Evaluation of the extraction of constraints: We validated whether the model had correctly extracted the dimensions onto which to provide either force or position control, by comparing the robot's quantitative performance in executing the task when using the proposed approach or other simple control schemes. For the quantitative assessment, we measured the effects of the determined variables as the determined constraints.

For evaluating the framework, we compared our approach with standard control modes: a position controller and an impedance controller with fixed stiffness values. For these two control modes, $N = 5$ different demonstrations were provided, using gravity compensation mode (*gcp*) and robot's execution was evaluated during motion replays ($\text{Rep}_i, i = 1 \dots 5$) in the different setups: position control (*pos*) and impedance control (*imp*). The performance under these control modes was compared with the developed approach (*amp*). Several replays were performed for each demonstrated motion. We constantly compensated for the decrease of the vegetable's height, during replays. Each group of one demonstration followed by five replays were performed on the same vegetable. A single vegetable type was used, and the task was demonstrated using five passes over the grating surface during each trial.

For all the trials, we measured the original and final weight of the vegetable ($u_{\text{init}}, u_{\text{fin}}[g]$) and the original and final height ($h_{\text{init}}, h_{\text{fin}}[\text{cm}]$). The original values were measured before the demonstration was performed, while the final values were measured at the end of the last replay round. For each round of demonstration and replay, we measured the weight of the grated part ($\Delta u[g]$) with a precision of $\pm 1g$ and counted the number of successful passes (*SP*).

We evaluated the task performance with respect to the following computed measures:

1) $u_{\text{ratio}}[\%]$, the ratio of the grated vegetable ($u_{\text{grated}} = \sum \Delta u$) as a percentage of the initial weight.
2) $h_{\text{ratio}}[\%]$, the percentage of the vegetable length being grated ($h_{\text{init}} - h_{\text{fin}}$) with respect to the initial length.
3) $\mathbf{SP}_{\text{ratio}}[\%]$, the percentage of SP out of the total passes performed.

Results are presented in Table II. Using a standard position controller (Trials 1–5) for replaying the motion gave good results in a very low number of cases: mean (M) = 12% and standard deviation (SD) = 10.95 SP, while the amount of vegetable grated was bellow one gram per trial ($M = 0.80$ g, SD = 0.83). When replaying the recorded motion using an impedance controller the number of SP increased ($M = 52.5\%$, SD = 25.16).

These results were compared against the proposed approach (see Table II, Trial 6), using the parameterization learned from demonstrations. The grating performance was assessed using the same performance metrics as for the standard control modes. The overall performance was better with respect to the amount of grated vegetable, and the number of SP.

b) Evaluation of the generalization ability: We tested whether the automatic segmentation of the task and the extraction of reference frame was correct and led to a correct reproduction when

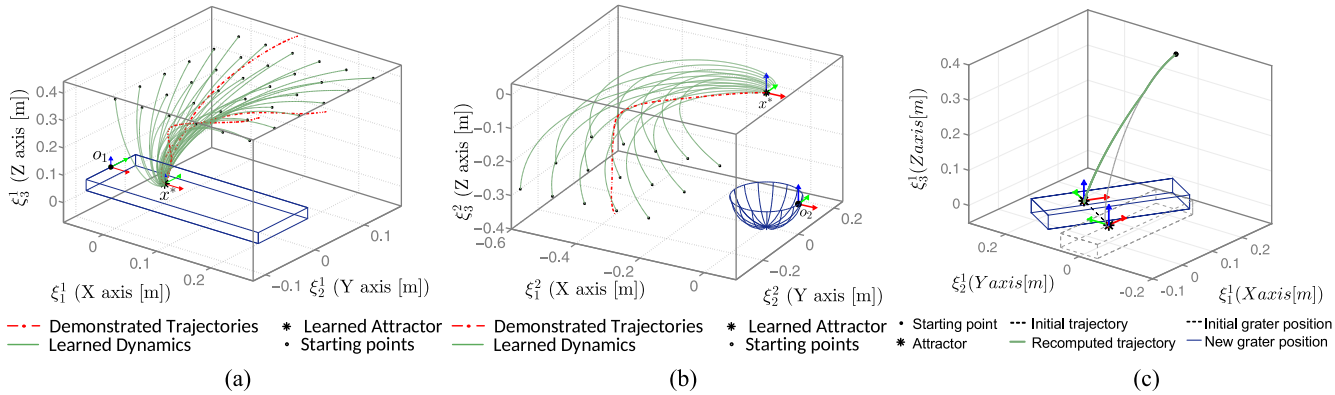

Fig. 9. Learned dynamics. (a) and (b) show generalization with respect to different starting locations; (c) shows the automatic adaptation of the trajectory with respect to changing positions of the object.

TABLE II
EVALUATION OF THE CONTROL MODES

		Trial 1			Trial 2			Trial 3			Trial 4			Trial 5			Trial 6 – amp		
Type	Control	Δu [g]	SP	SP_{ratio} [%]	Δu [g]	SP	SP_{ratio} [%]	Δu [g]	SP	SP_{ratio} [%]	Δu [g]	SP	SP_{ratio} [%]	Δu [g]	SP	SP_{ratio} [%]	Δu [g]	SP	SP_{ratio} [%]
N1	gcp	4	5	100	7	5	100	9	5	100	6	5	100	6	5	100	7	5	100
Rep1	pos	1	1	20	2	1	20	1	1	20	0	0	0	0	0	0	4	4	80
Rep2	imp	2	3	60	2	2	40	7	4	80	5	4	80	2	2	40	5	4	80
Rep3	imp	3	2	40	5	3	60	6	4	80	3	2	40	1	1	20	8	5	100
Rep4	imp	4	4	80							1	1	20	1	1	20	9	5	100
Rep5	imp	7	4	80							1	1	20	5	4	80			
u_{ratio} [%]		21.00			21.62			31.08			17.78			18.07			35.86		
h_{ratio} [%]		42.06			35.65			35.00			42.30			26.92			48.15		

For Trials 1–5, we compared the demonstrated motion D_i provided using the robots gravity compensation mode (*gcp*), with a standard position control mode (*pos*), and with an impedance controller with fixed stiffness (*imp*). Trial 6 illustrates the performance of the proposed controller, learned from the $N = 18$ demonstrations (*amp*).

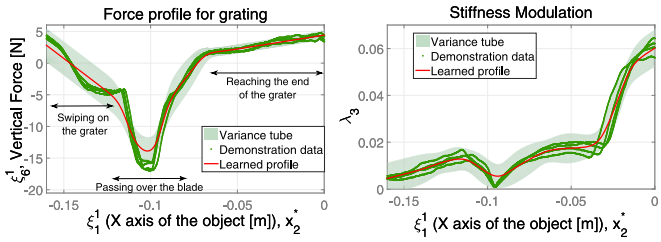

Fig. 10. Force profile and stiffness modulation used when grating.

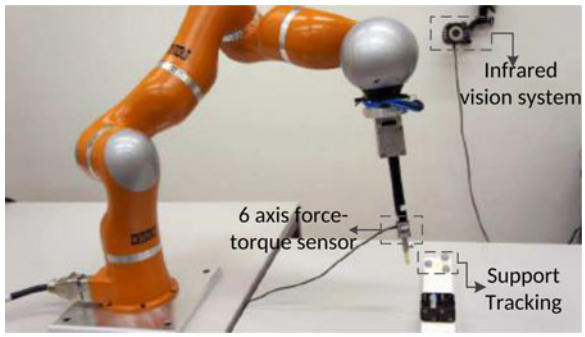

Fig. 11. Experimental setup for removing a battery from a charging stand.

the position of the objects was changed. The robot regenerated the complete sequence and managed to complete the overall task comprising the three segments even when the objects were located in arbitrary positions and orientations, none of which were seen during training.

The importance of being able to change the reference frame is illustrated in Fig. 1, when using different positions and orientations of the two objects. In this case, we performed a pure qualitative assessment by placing the objects in random positions and orientations in the robot's reachable space, and using different vegetables. We measured the number of SP over the grater's surface. Similarly, we tested the functionality over a larger grating surface.

B. Battery Charger Task

We tested the ability of the proposed method to properly extract constraints on a second task, i.e., removing a battery from a charging stand (see Fig. 11). The task was very fast paced. From the first segment to the last segment, the task lasted on average less than 5 s.

In this example, we used a single object o, the battery stand. We recorded data at 1 kHz, from human demonstrations by using vision to track the motion of the tool and of the object. Additionally, we mounted a six-axis force torque sensor on the tool to record precise interaction forces (see Fig. 11). The steps

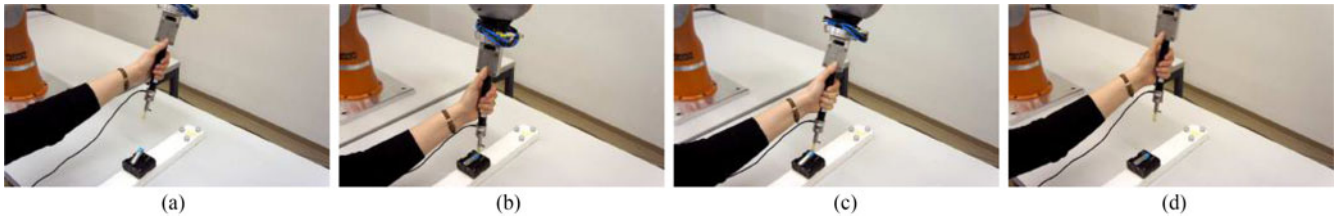

Fig. 12. Atomic actions in performing the task. The task typically consists of reaching for the battery stand, applying a force that tensions the spring inside the support (pushing), taking out the battery (lifting), and reaching away. (a) Reaching. (b) Pushing. (c) Lifting. (d) Reaching back.

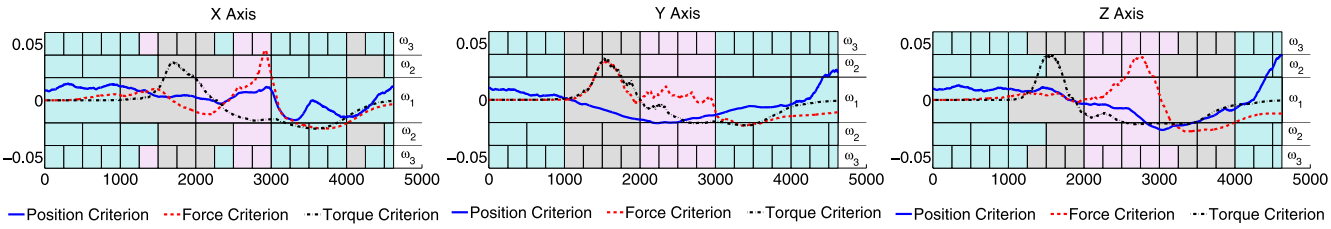

Fig. 13. Computed Criteria. We contrast the contribution of position, force, and torque as variables of interest. Segmentation obtained using time window of various sizes: $\omega_1 = 1000$ time steps; $\omega_2 = 500$ time steps; $\omega_3 = 250$ time steps. We retain the segments obtained after using the time window ω_1.

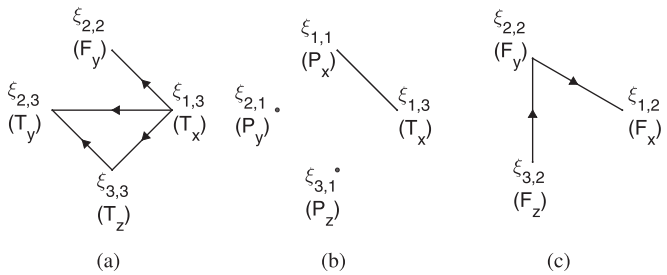

Fig. 14. Causality relationships corresponding to the "push" and "lift" segments. The oriented arrow shows the start variable to be causal for the end variable. An unoriented edge shows double causality.

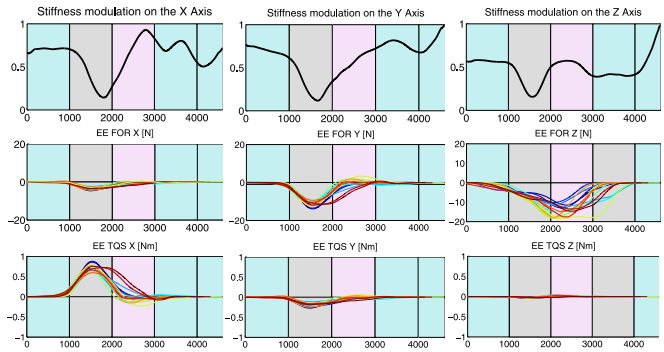

Fig. 15. Segmentation for the battery extraction task.

of the task are shown in Fig. 12. The dataset consisted of nine variables, 3-D measurements of end effector position, force, and torque. We computed the criterion as described in Section III, using a time window of 1000 time steps, for position, force, and torque on each axis of the object (see Figs. 13 and 15).

For the first part of the task (reaching), the criterion for position was dominant on all axes. For the following part (pushing and lifting the battery), there was a clear separation between the two segments on the *X*- and *Z*-axis of the object (see Fig. 13(a) and (c), marked by changing the variable of interest (i.e., torque than force), while on the *Y*-axis (perpendicular to the object), torque and force were equally important with a small relative difference in their criterion [see Fig. 13(b)].

In this case, the proposed approach does not offer a clear decomposition of the task, suitable for hybrid force–position control. The method offers a relative weighting between the importance of different variables acting on the same axis. However, to still use a hybrid controller, we should determine the relative importance between axes, as in this case, the forces and torques acting sideways were just a reaction to the motion of taking out the battery. For this, we propose studying the causal relation between the variables of interest determined above, across all the N demonstrations. This allows us to:

1) have a relative weighting of the axis's importance;
2) determine on which of the other axis the motion should be conditioned on.

For analyzing the causality in the data, we have used an existing MATLAB toolbox [34]. Fig. 14(a) shows the relationship between the variables of interest determined on each axis for the "push" segment. The force component on the *X*-axis along the object (corresponding to torque around the *X*-axis of the end effector) was causal for the force components around other axes. The amplitude of the causal interaction was 0.37 for the torque around the *Y*-axis and 0.1793 for the force on the *Z*-axis, thus proving that the interaction is stronger in the *XY* plane of the battery charger. Second, we studied the connectivity of the most important variable with all the other secondary variables [i.e., the change of position on all axis; see Fig. 14(b)], which showed a causality relation in both ways. This allowed us to reduce the number of axis on which we perform force control in this segment to one (the *X*-axis) and to automatically determine that this should be encoded based on a change of position

along that axis (it also determined a change of position on this axis). Similarly analyzing the causal structure in the data for the "lifting" segment [see Fig. 14(c)] allowed us to reduce the dimensionality of this model.

Fig. 17 shows robot reproduction and generalization to different positions of the battery charger stand.

V. Discussion

Our approach of extracting continuous soft constraints from human demonstration was tested on a cooking task encompassing three segments and on an office task with four segments. The tasks differed in duration and the set of important variables. The proposed method extracted the necessary information for performing the tasks and encoded the task flow without any prior knowledge. The tasks were reproduced from a time-invariant encoding, using an impedance controller parameterized by the continuous constraints.

From a human–robot interaction perspective, this method can facilitate teaching interactions as it allows the user to demonstrate the whole task rather than individual actions. A fragmented representation can be demanding when the user has to actively teach the robot how to perform the task. As multiple demonstrations are required for generalization, it is more convenient for the user to demonstrate the whole task, rather than individual actions, such as reaching movements.

We further discuss several aspects that could influence the behavior of our approach.

A. Influence of Other Variables on Segmentation Points

The approach presented above can be extended by taking into account other variables. For the grating task, we computed the variance over trials and time window for two other measures: the torques sensed at the end effector, and the end effector velocity (a total of $L = 4$ variables). The analysis, using the same approach presented in Section III, showed that using the extra information provided by the velocity, or torque, data did not significantly modify the segmentation points.

B. Choice of Window Size

In the current implementation, the window size was chosen by the user. The size of the window might influence the number and location of the segmentation points obtained; therefore, an automatic way of obtaining an adaptive time window was proposed in Section III. This, however, required to set a threshold of the minimum amount of change and hence introduced yet another open parameter. In Fig. 16, we show how various time window sizes affect the variance computation.

C. Task Generalization

Automatically determining the object of interest and expressing the motion in this frame allows us to easily generalize to new positions and orientations of the object, as when the object position changes, this frame changes with it, ensuring proper execution. However, the proposed approach has a two-level specification of the reference frame, by determining an object of interest and one or more attractors (i.e., relative positioning), with respect to this object. There can be multiple attractors with respect to a single object. For example, in the grating task, we needed to reach the grater at a certain point above the blade, but the grating motion ended at a point just after passing the blade. The two points determined different actions performed with respect to the same object, and thus refining the task encoding. Additionally, relating attractors to the initial frame, rather than storing only the attractor points with respect to the world, allowed us to implicitly capture properties of the object, such as rigidity.

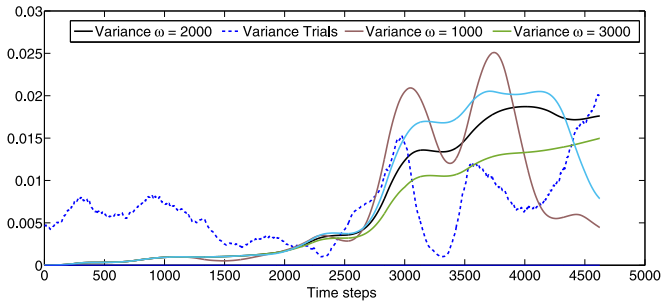

Fig. 16. Change of time window variance with respect to the window size. This representation corresponds to the criterion for position on the *X*-axis of the battery task.

Additionally, for the first task, we tested the developed controller for a different grating surface and a softer vegetable. This resulted in proper grating. However, in the current implementation, the choice of modeling the force as conditioned on the position was ad-hoc prior information. The possibility to learn and extract automatically that there is a correlation between these two variables and the directionality of the correlation was explored in the second robotic experiment, in Section III-B.

D. Stiffness Modulation

Modulating the arms stiffness is important for several reasons: 1) it allows us to apply the determined decomposition of force and position control. 2) Proper stiffness contributes to successfully executing the task. For example, a robot that is too stiff in the grating segment would break the carrot or other soft vegetables (like a cucumber) during grating, while a robot that lacks sufficient stiffness would get stuck in the graters blade and, therefore, not manage to perform the task. 3) Stiffness modulation is also important with respect to safety issues: a stiff robot is required to be able to perform some parts of the task, but a less stiff robot when reaching the trash, for example, is safer, in case of colliding with a human. This aspect of changing the arm stiffness is observed in humans as well, when performing various tasks and additionally helps in reducing the energy consumption at the joint level.

E. Ending Condition for Repetitive Tasks

In the carrot-grating task, for example, while reaching the first attractor was important for starting the grating, still looking at the complementary information of force indicated that at the end of the segment the end effector was in contact.

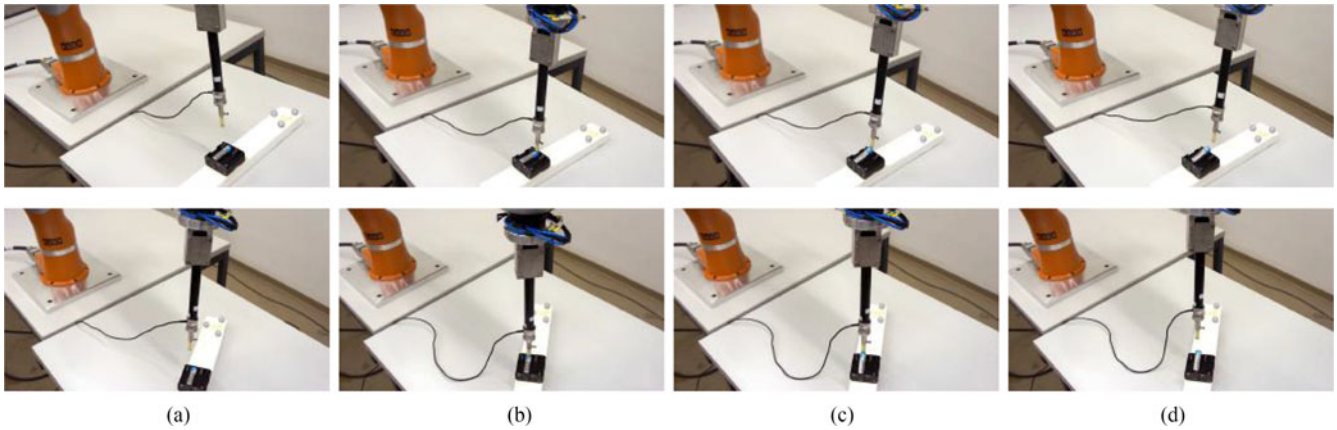

Fig. 17. Robot task execution using different configurations of the object with respect to the robot's base. (a) Reaching. (b) Pushing. (c) Lifting. (d) Reaching back.

Similarly, during the grating motion, mainly the vertical force was the important variable for control. However, the vertical position of the end effector with respect to the grater held complementary information for ending this action. Namely, for each grater pass, we observed a decrease in height by approximately 2 mm. If the task was demonstrated until the vegetable is fully grated, this implied finishing the grating action at the same end effector height above the grater. Therefore, we could consider this information as an ending condition for the repetitive motion.

VI. Conclusion

The presented approach for extracting task constraints takes advantage of the existing variance in the demonstrated data and proposes a criterion for detecting regions of coherence across demonstrations. Objects upon which an action was performed are determined. The action is further encoded in the local frame of reference, in a time-invariant manner, preserving the task flow of actions.

In particular, we compared different measurements (like position and force) and modulated their contribution to the controller used in reproducing the motion, by using a weighting factor that adapts the robot's stiffness. In addition, by weighting the relative importance of each of the task variables when expressed in the reference system of the objects involved in the task, we can determine the suitable reference frame to be used in each segment. Finally, a set of segmentation points were obtained by splitting the motion whenever a change in the reference frame or in the variables of interest occurred. The approach was validated on a kitchen task (grating vegetables), and an office task (removing a battery from a charging stand) achieving good generalization results, and managing to capture the dynamics of a fast task.

The advantages of using this segmentation and feature extraction method are first decreasing the task complexity by focusing on learning just the variables that are important for each region of the task (i.e., encode just end effector position for a reaching motion versus accounting for position and force in manipulation subtasks) and second achieving efficient generalization when the position of the objects is changed.

It determines the proper chain of actions in the task and the conditions for the transition between actions. Automatically extracting this information can contribute to both simplifying the control and to automating this part of the Learning from Demonstration procedure, which was usually done manually.

A limitation of the current approach is the fact that the method relates only the arm behavior to how the manipulation should be performed on an object, while interacting with it, but is not suitable for modeling effects on the object. A second limitation is that it does not directly apply to tasks where the "lack of change" is important (like controlling for zero force on one axis) as being key to task completion. However, this can be addressed by studying the relative importance of each axis, extending the approach proposed in Section III-B.

References

[1] L. Villani and J. De Schutter, "Force control," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin, Germany: Springer, 2008, pp. 161–185.
[2] M. Howard, S. Klanke, M. Gienger, C. Goerick, and S. Vijayakumar, " A novel method for learning policies from variable constraint data," *Auton. Robots*, vol. 27, no. 2, pp. 105–121, 2009.
[3] A. F. Bobick and A. D. Wilson, "A state-based approach to the representation and recognition of gesture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 12, pp. 1325–1337, Dec. 1997.
[4] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Trans. Syst., Man Cybern. B, Cybern.*, vol. 37, no. 2, pp. 286–298, Apr. 2007.
[5] S. Calinon, F. Guenter, and A. Billard, "On learning the statistical representation of a task and generalizing it to various contexts," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2006, pp. 2978–2983.
[6] M. Muhlig, M. Gienger, J. Steil, and C. Goerick, "Automatic selection of task spaces for imitation learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 4996–5002.
[7] M. Stilman, "Task constrained motion planning in robot joint space," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2007, pp. 3074–3081.
[8] V. Sukhoy, V. Georgiev, T. Wegter, R. Sweidan, and A. Stoytchev, "Learning to slide a magnetic card through a card reader," in *Proc. Int. Conf. Robot. Autom.*, 2012, pp. 2398–2404.
[9] P. Kormushev, S. Calinon, and D. G. Caldwell, "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input," *Adv. Robot.*, vol. 25, no. 5, pp. 581–603, 2011.
[10] S. Calinon, I. Sardellitti, and D. Caldwell, "Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies," in *Proc. IEEE/RSJ Int. Conf. Int. Robots Syst.*, 2010, pp. 249–254.
[11] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern Recog.*, vol. 36, no. 3, pp. 585–601, 2003.

[12] J. Lin and D. Kulic, "Automatic human motion segmentation and identification using feature guided HMM for physical rehabilitation exercises," in Robotics for Neurology and Rehabilitation, Workshop at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011.
[13] J. Shim and A. L. Thomaz, "Human-like action segmentation for option learning," in *Proc. IEEE RO-MAN*, 2011, pp. 455–460.
[14] O. Mangin and P.-Y. Oudeyer, "Learning to recognize parallel combinations of human motion primitives with linguistic descriptions using non-negative matrix factorization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2012, pp. 3268–3275.
[15] L. Tao, E. Elhamifar, S. Khudanpur, G. Hager, and R. Vidal, "Sparse hidden Markov models for surgical gesture classification and skill evaluation," in *Information Processing in Computer-Assisted Interventions* (ser. Lecture Notes in Computer Science), vol. 7330, P. Abolmaesumi, L. Joskowicz, N. Navab, and P. Jannin, Eds. Berlin, Germany: Springer, 2012, pp. 167–177.
[16] D. Kulic, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura, "Incremental learning of full body motion primitives and their sequencing through human motion observation," *Int. J. Robot. Res.*, vol. 31, no. 3, pp. 330–345, 2012.
[17] W. Takano and Y. Nakamura, "Humanoid robot's autonomous acquisition of proto-symbols through motion segmentation," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2006, pp. 425–431.
[18] D. Kulic, W. Takano, and Y. Nakamura, "Combining automated on-line segmentation and incremental clustering for whole body motions," in *Proc. Int. Conf. Robot. Autom.*, 2008, pp. 2591–2598.
[19] D. H. Grollman and O. C. Jenkins, "Incremental learning of subtasks from unsegmented demonstration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 261–266.
[20] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, "Learning motion patterns of people for compliant robot motion," *Int. J. Robot. Res.*, vol. 24, pp. 31–48, 2005.
[21] S. H. Lee, H. K. Kim, and I. H. Suh, "Incremental learning of primitive skills from demonstration of a task," in *Proc. 6th Int. Conf. Human-Robot Interaction*, 2011, pp. 185–186.
[22] T. Matsuo, Y. Shirai, and N. Shimada, "Automatic generation of HMM topology for sign language recognition," in *Proc. 19th Int. Conf. Pattern Recog.*, 2008, pp. 1–4.
[23] D. Kulic and Y. Nakamura, "Scaffolding on-line segmentation of full body human motion patterns," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 2860–2866.
[24] G. Oriolo and M. Vendittelli, "A control-based approach to task-constrained motion planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 297–302.
[25] Ye, Gu, and Ron Alterovitz, "Demonstration-guided motion planning," International Symposium on Robotics Research (ISRR). vol. 5. 2011.
[26] M. Beetz, L. Mösenlechner, and M. Tenorth, "CRAM—A cognitive robot abstract machine for everyday manipulation in human environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 1012–1017.
[27] N. Dantam and M. Stilman, "The motion grammar: Analysis of a linguistic method for robot control," *IEEE Trans. Robot.*, vol. 29, no. 3, pp. 704–718, Jun. 2013.
[28] J. D. Schutter, T. D. Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *Int. J. Robot. Res.*, vol. 26, no. 5, pp. 433–455, 2007.
[29] S. Niekum, S. Chitta, A. G. Barto, B. Marthi, and S. Osentoski, "Incremental semantically grounded learning from demonstration," presented at the Robot. Sci. Syst. Conf., Berlin, Germany, 2013.
[30] A. Arsenio, "Learning task sequences from scratch: Applications to the control of tools and toys by a humanoid robot," in *Proc. IEEE Int. Conf. Control Appl.*, 2004, vol. 1, pp. 400–405.
[31] R. Jäkel, S. R. Schmidt-Rohr, M. Lösch, and R. Dillmann, "Representation and constrained planning of manipulation strategies in the context of programming by demonstration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 162–169.
[32] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto, "Robot learning from demonstration by constructing skill trees," *Int. J. Robot. Res.*, vol. 31, no. 3, pp. 360–375, 2012.
[33] A. Shukla and A. Billard, "Coupled dynamical system based arm–hand grasping model for learning fast adaptation strategies," *Robot. Auton. Syst.*, vol. 60, no. 3, pp. 424–440, 2012.
[34] A. K. Seth, "A MATLAB toolbox for Granger causal connectivity analysis," *J. Neurosci. Methods*, vol. 186, pp. 262–273, 2010.

Ana Lucia Pais Ureche received the B.S. and M.S. degrees in automation control and systems engineering from Polytechnical University of Bucharest, Bucureşti, Romania, in 2009 and 2011, respectively. She is currently working toward the Ph.D. degree with the Learning Algorithms and Systems Laboratory, Swiss Federal Institute of Technology in Lausanne, Lausanne, Switzerland.

Her research interests include programming by demonstration and machine learning techniques for improving human–robot interaction and task learning.

Keisuke Umezawa received the B.S. and M.S. degrees in mechanoinformatics from University of Tokyo, Tokyo, Japan, in 2012 and 2014, respectively.

Since 2014, he has been a Quantitative Analyst with Bank of Tokyo Mitsubishi UFJ, Tokyo, Japan.

Yoshihiko Nakamura (F'11) received the Doctor of Engineering degree from Kyoto University, Kyoto, Japan.

He is currently a Professor with the Department of Mechano-Informatics, University of Tokyo, Tokyo, Japan. Before joining University of Tokyo in 1991, he was with Kyoto University as an Assistant Professor and with University of California Santa Barbara as an Assistant and Associate Professor. His research interests include humanoid robotics, cognitive robotics, neuromusculoskeletal human modeling, biomedical systems, and their mathematical problems.

Dr. Nakamura is a Fellow of the Japan Society of Mechanical Engineers, the Robotics Society of Japan, and the World Academy of Art and Science. He serves as the President of IFToMM (2012-2015). He is a Foreign Member of the Academy of Engineering Science of Serbia and TUM Distinguished Affiliated Professor of Technische Universität München.

Aude Billard received the M.Sc. degree in physics from École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 1995, and the M.Sc. degree in knowledge-based systems and the Ph.D. degree in artificial intelligence from University of Edinburgh, Edinburgh, U.K., in 1996 and 1998, respectively.

She is currently a Professor of micro and mechanical engineering and the Head of the LASA Laboratory, School of Engineering, EPFL. Her research interests include machine learning tools to support robot learning through human guidance. This extends also to research on complementary topics, including machine vision, and its use in human–robot interaction and computational neuroscience to develop models of motor learning in humans.

Dr. Billard received the Intel Corporation Teaching Award, the Swiss National Science Foundation Career Award in 2002, the Outstanding Young Person in Science and Innovation from the Swiss Chamber of Commerce, and the IEEE Robotics and Automation Society (RAS) Best Reviewer Award in 2012. She served as an Elected Member of the Administrative Committee of the IEEE RAS for two terms (2006–2008 and 2009–2011). She is the Chair of the IEEE-RAS Technical Committee on Humanoid Robotics.