

# Learning adaptive dressing assistance from human demonstration

Emmanuel Pignat and Sylvain

*Idiap Research Institute, Martigny, Switzerland*  
{emmanuel.pignat, sylvain.calinon}@idiap.ch

---

## Abstract

For tasks such as dressing assistance, robots should be able to adapt to different user morphologies, preferences and requirements. We propose a programming by demonstration method to efficiently learn and adapt such skills. Our method encodes sensory information (relative to the human user) and motor commands (relative to the robot actuation) as a joint distribution in a hidden semi-Markov model. The parameters of this model are learned from a set of demonstrations performed by a human. Each state of this model represents a sensorimotor pattern, whose sequencing can produce complex behaviors. This method, while remaining lightweight and simple, encodes both time-dependent and independent behaviors. It enables the sequencing of movement primitives in accordance to the current situation and user behavior. The approach is coupled with a task-parametrized model, allowing adaptation to different users' morphologies, and with a minimal intervention controller, providing safe interaction with the user. We evaluate the approach through several simulated tasks and two different dressing scenarios with a bi-manual Baxter robot.

---

## 1. Introduction

One of the key abilities that will allow a wider spread of assistive robotics is learning and adaptation. In the case of assisting a person to dress, the robot should for example be able to adapt to different morphologies, pathologies or stages of recovery, implying different requirements for movement generation and physical interaction. This behavior is person-dependent, and cannot be pre-engineered (it is not fixed in time). It must instead continuously adapt to the user by considering acclimating or rehabilitation periods and aging. This requires the robot to represent dressing skills with a flexible model allowing adaptation both at the level of the movement and impedance parameters (e.g., based on the height of user or preferred forces), and at the level of the procedure (e.g., reorganization of the sequence of actions).

This assistance is currently provided by healthcare workers, which is not always convenient. From the worker perspective, there is a lack of employees dedicated to this service, while the activity takes time and is not particularly gratifying. From the patient's perspective, such assistance is often viewed negatively because it drastically reduces the sense of independence of the person (e.g., the person cannot go out of her own free will because of this

dependence to another person for getting dressed). Providing robots with dressing assistance capabilities would have benefits on both sides, but it is not possible to pre-program all the dressing behaviors and requirements in advance. In this context, the programming by demonstration (PbD) paradigm provides a human-oriented solution to transfer such assistive skills from a non-expert user to the robot. This can be achieved by means of kinesthetic teaching or motion capture system, where several demonstrations of the task executed in various situations can be used to let the robot acquire the person-specific requirements and preferences rapidly.

In PbD, skills are generally decomposed into elementary building blocks or movement primitives (MPs) that can be recombined in parallel and in series to create more complex motor programs. They are particularly suitable to generate motion trajectories. However, in order to tackle the highly multimodal interaction involved in assistive tasks, the notion of movement primitives should be enlarged to a richer set of behaviors including reaction, sensorimotor and impedance primitives.

In particular, such a model should also be able to encode both time-independent and time-dependent behaviors. A typical example of time-independence in this human-centric context arise when holding a coat and waiting for someone to come; the duration of the associated movement primitive to help the person should here be triggered by the proximity and attention of the user, and is thus, in this case, time-independent. Other parts of the skill are in contrast time-dependent when a movement

needs to be completed after being initiated, which typically appears when more dynamic features are required, such as slipping on pants. Often, a relative time dependence is required to guarantee a cohesive evolution of the movement (i.e., with a local time instead of an absolute time).

The above issue is closely related to the problem of organizing the action primitives in series and in parallel, as well as deciding which one to choose and when to switch between them. The contribution of this paper is to present a probabilistic approach to learn the movement primitives, their temporal characteristics as well as the switching rules determining their sequences. The proposed approach is based on a generative model to encode the sensorimotor patterns observed during the demonstrations. We demonstrate the capability of this human-centric approach with different simulated scenarios as well as with a Baxter robot, by considering the task of helping a user to put on the sleeve of a jacket.

Supplementary material, including source code and video, is available at <http://programming-by-demonstration.org/RAS2017>.

### 1.1. Related Work

Dressing is an everyday activity that is representative of the multifaceted challenges behind learning and adaptation in assistive robotics and is the topic of many works. In [4], Gao *et al.* concentrate on the challenge of modeling the user and optimizing personalized paths. In [19], Tamei *et al.* concentrate on optimizing joint angle trajectories, described by via-points, with the aim of generalizing the skill to different postures through reinforcement learning. The approach is based on the use of topological coordinates for the encoding of dressing skills, with a predetermined reward function describing the specific task to handle. In [3], trajectory planning with waypoints is used for dressing tasks and several techniques are investigated for failure detection.

In [13], a risk-sensitive control approach for physical human-robot interaction is proposed. It takes into account uncertainty information to tune the control gains (the more confident the robot is, the more force it can apply). This principle was applied in [2] in the PbD context, by inferring control gains from the variability observed in a set of demonstrations. In these approaches, variable impedance is exploited as a key element for both perceived and actual safety.

In [12], movements are represented as trajectory distributions, where the extension to a mixture model enables the encoding of multiple human-robot interaction behaviors. Our approach shares a similar strategy, by using a hidden Markov model (HMM) structure to encode multiple trajectories with components shared across different tasks. The HMM structure allows us to replace the time-dependent structure of trajectory distributions (usually represented as temporal radial basis functions) with

a more elaborated encoding enabling the retrieval of both time-independent and time-dependent behaviors.

In [15], demonstrations are first segmented into motion categories with HMM. They are then split into semantic units, allowing MPs to be used multiple times in the skills for different purposes. In our work, we seek to learn directly the semantic units as sensorimotor states with the HMM.

Modeling human-robot interaction with an HMM was also proposed in [11] and [5]. In [11] learning motion and interaction are achieved in two separated stages. The model described by [5] is the closest to ours. The approach was demonstrated in learning responsive behaviors for tasks with joint dynamics, such as high-five gestures.

We adopt a similar perspective in that the sequence of states to control the robot is retrieved from observations of human behavior. However, instead of observing the behavior of the human and controlling the robot in two successive stages, we focus on continuous interactions in which the sequence of states should be updated as often as possible, by taking into account the present and history of actuation (state of the robot) and perception (human's state).

A similar use of an HMM in the field of speech processing consists of exploiting Gaussian conditioning to provide adaptation [7]. Two times series (joint distributions with silent articulation as input, and audible speech as output) are encoded in a full-covariance HMM. Based on the silent articulation time series, the goal is to retrieve the corresponding speech signal. Viterbi algorithm is used to retrieve the sequence of states, where the output of each state is then computed by Gaussian conditioning. The approach that we propose follows a similar scheme, with the position of the human user used as input, and with projections of the hand of the robot in multiple coordinate systems used as output.

The use of hidden semi-Markov models (HSMM) have already been investigated for encoding robotic skills [20]. It allows for a more precise retrieval of the time constraints of the tasks than standard HMMs, by explicitly modeling duration distributions. In [20], the HSMM is only encoding robot actuation. Thus, this approach does not solve the problem of learning when to trigger actions, react to the human or choose between multiple options.

## 2. Proposed Approach

In this work, movement primitives (MPs) are defined at a low level and correspond to the states of an HSMM [14]. We will consider the special case of discrete operational space attractors, but the approach is not limited to this form of position attractors and can be readily extended to other forms of reference signals including velocities, accelerations and forces, or their equivalence in joint space. Fig. 1 gives an overview of the proposed approach.

The task-parametrized version of HSMM is used to be robust to varying situations, in this case related to dif-

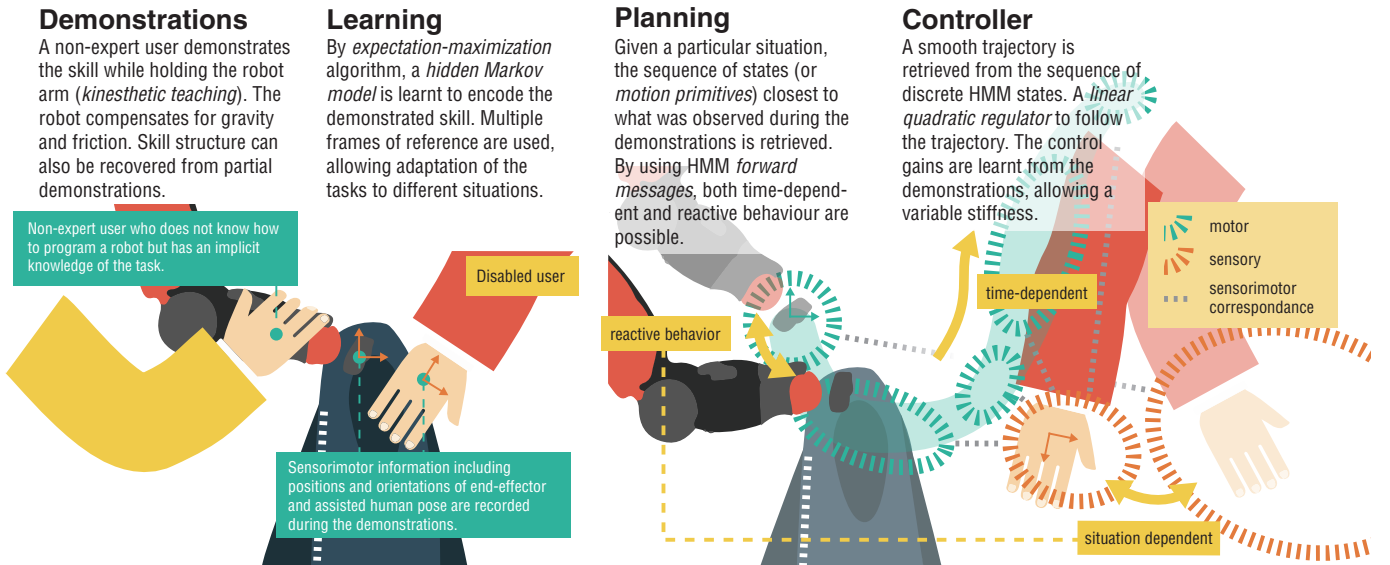


Figure 1: Overview of the proposed approach.

ferent users’ morphologies. It consists of encoding an observed movement projected onto multiple coordinate systems attached to landmarks of potential interest for the task (e.g., cloth, user’s hand or shoulder). The model parameters are learned from human demonstrations data using an expectation-maximization (EM) algorithm.

The main contribution of this paper is to augment the MPs with sensory data, that are usually encoding only motor commands, and model them in an HSMM. We show that it provides a solution for the problems mentioned above, namely, encoding both time-dependent and independent behaviors, as well as learning how to make transitions between the MPs.

In order to reproduce the demonstrated behaviors, the robot has to choose which state of the HSMM to activate at each time step. This decision should depend on what happened at the previous time steps, the temporal characteristics of each MP and the probability of activating each MP given the currently observed sensory stream. We show that the forward messages [17] provide a formal approach to this end. It allows the robot to compute marginals of the state indicator given measurement (sensory information), transition and duration probabilities.

In order to control the robot, we retrieve a continuous distribution representing the reference to track from the HSMM discrete states using a method similar to [10]. These targets are tracked using linear quadratic tracking (LQT) based on a minimal intervention principle. It exploits the covariance of the targets as a penalty term that provides variable stiffness for a desired minimal control cost [2]. In the context of dressing assistance, such formulation provides a safe interaction with the user as the robot is only stiff when this is strictly required for the success of the task.

### 2.1. Hidden Semi-Markov Model (HSMM)

A hidden semi-Markov model (HSMM) is used to model the complex task that the robot should learn. Each state of the HSMM corresponds to a low-level sensorimotor movement primitive that encodes both sensory information about the external world and physical motor actuation. The HSMM is considered in discrete time steps, as most conventionally used.

The first challenge is to define a well-suited representation of the sensorimotor stream required for the dressing assistance, which should be estimated from a set of partial demonstrations of the skill (i.e., not time-aligned nor complete demonstration). The second challenge is to use this model for regenerating appropriate behaviors on the robot.

An HMM [17] is a stochastic process composed of discrete states, denoted  $z_t$  at time step  $t$ . For the first time step, the states are distributed according to an *initial probability distribution*  $\mathbf{\Pi}$ ,

$$z_0 \sim \mathbf{\Pi}.$$

The transitions then follow a Markovian structure where each next state is drawn from a state-specific *transition distribution*,

$$z_t | z_{t-1} \sim \boldsymbol{\pi}_{z_{t-1}}.$$

These states emit observations following a state-specific *emission or observation distribution* that are independent given the sequence of states,

$$\mathbf{y}_t | z_t \sim F(\theta_{z_t}),$$

where  $F$  is a family of distributions parametrized by  $\theta_{z_t}$  for state  $z_t$ . These properties are primordial for tractable computation of some essential marginals using forward-backward algorithm [17].

Robotics applications often require some duration information for each MP. However, standard HMM only allows a crude modeling of this duration as a geometric distribution. In order to explicitly model these durations, a hidden semi-Markov model (HSMM) can instead be used, see [14] for details. In a variable-duration HMMs (a type of HSMM), each state emits a sequence of conditionally independent observations, corresponding to a segment. The length of the segment is drawn from a state-specific duration distribution, for example  $\mathcal{N}(\mu_{z_t}^D, \Sigma_{z_t}^D)$ , where  $\mu_{z_t}^D$  denotes the mean of the duration,  $\Sigma_{z_t}^D$  the variance and  $z_t$  the index of hidden state at time  $t$ . In [20], the authors presented a robotic application of the HSMM. For concision, we describe our methods through an HMM but the same holds for an HSMM with some modifications of the forward-backward algorithm for computing marginals such as  $p(z_n | \mathbf{y}_1, \dots, \mathbf{y}_n)$ .

If an HMM still allows a crude modeling of the durations, it does not model at all the aspects of time dependence or independence that we discussed before. In the HSMM, the variance  $\Sigma_i^D$  precisely models this concept. A MP subject to precise timing constraints will have a very small variance. On the opposite, a very large variance assigned to a MP will correspond to a time-independent behavior, where the transition to a next step can be triggered in another manner (e.g., from perception by exploiting the conditioning property).

## 2.2. Observation model

From human demonstrations, we get the observations  $\mathbf{y}_t = \{\{\xi_t^{(j)}\}_{j=1}^P, \xi_t^S\}$  of the skill to transfer to the robot.  $\{\xi_t^{(j)}\}_{j=1}^P$  is the position and velocity of the robot end-effector from the perspective of  $P$  Cartesian coordinate systems. These coordinate systems are defined by  $\{\mathbf{b}_{t,j}, \mathbf{A}_{t,j}\}_{j=1}^P$ , where  $\mathbf{b}_{t,j}$  is the position of their origin and  $\mathbf{A}_{t,j}$  their rotation matrix. They are associated with landmarks of (potential) interest for the task.  $\xi_t^S$  denotes sensory data such as the position of the user's hand or head.

The observations are modeled by a multivariate Gaussian distribution, with the probability of an observation given the state indicator  $z_t$  given by

$$p(\mathbf{y}_t | z_t) = \prod_{j=1}^P \mathcal{N}(\xi_t^{(j)} | \mu_{z_t}^{(j)}, \Sigma_{z_t}^{(j)}) \mathcal{N}(\xi_t^S | \mu_{z_t}^S, \Sigma_{z_t}^S). \quad (1)$$

Thus, the model is compactly defined by

$$\Theta = \left\{ \pi_i, \Pi_i, \mu_i^S, \Sigma_i^S, \{\mu_i^{(j)}, \Sigma_i^{(j)}\}_{j=1}^P \right\}_{i=1}^K,$$

where  $K$  is the number of states of the HMM.

These parameters can be estimated with an EM algorithm, see for example [17]. With EM, the initialization is very important. Pure spatial initialization such as k-means can be considered. If the skill can be represented

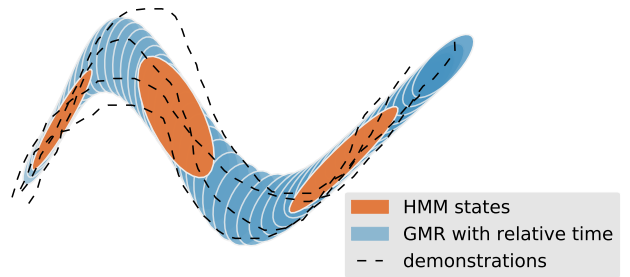


Figure 2: Demonstrations are encoded using a discrete state HMM (orange). With a local time variable, it is possible to retrieve the corresponding time-dependent trajectory using GMR (blue). This approach brings together the advantages of GMR and HMM (i.e., without the need to realign the demonstration, and by keeping the capability to encode multiple options or recurring patterns).

by a left-to-right model (an HMM with a unique sequence of state), it is possible to split the data in  $K$  bins of equal duration, that are then used to initialize the observation parameters of the  $K$  states. The number of hidden states can be determined empirically (the approach we used in the experiment), by cross-validation or with Bayesian non-parametric techniques [9, 6].

## 2.3. Relative time encoding

Most learning by demonstration approaches model a direct dependence between temporal and spatial data, with either radial basis functions (RBFs) [16] or Gaussian mixture regression (GMR) [1]. This requires a realignment in time of the different demonstrations. For simple discrete (point-to-point) tasks, this can be achieved by dynamic time warping (DTW), but for tasks implying loops or time-independent parts, such preprocessing is not possible.

The HSMM can partially handle such a problem by modeling the duration distribution of each state, but it fails to encode the local spatio-temporal correlations, as in the case of GMR. Different approaches have been proposed to deal with the discreteness of the HMM states in synthesis problems. An option is to augment the observation model with dynamic features, see e.g. [18]. Another option is to couple the approach with an optimal control strategy by adding a cost penalizing acceleration, see e.g. [20]. An alternative approach suggested by [10] is to combine the HMM with a local version of GMR, encoding relative time instead of the time from the onset of the motion. This approach, that we will exploit, encodes locally in each state the correlation between the spatial and temporal data.

We define a relative time variable  $\phi$  that takes the value  $-1$  when we enter a state and  $1$  when we leave it. In HSMMs, this variable can be linked to the existing timer, indicating the remaining duration of the current state and deterministically decremented to zero [14]. Once the HSMM is trained over spatial data, we retrieve the most probable states sequence for each demonstration

using Viterbi algorithm. From these sequences of states, we augment the observation with a relative time variable for each state encountered in the demonstrations. We then augment the observation model in each state to encode the local correlation between the relative time and the spatial data. The observation model is then described by

$$\boldsymbol{\mu}_i = \begin{bmatrix} \boldsymbol{\mu}_i^\phi \\ \boldsymbol{\mu}_i^\circ \end{bmatrix}, \quad \boldsymbol{\Sigma}_i = \begin{bmatrix} \boldsymbol{\Sigma}_i^\phi & \boldsymbol{\Sigma}_i^{\phi\circ} \\ \boldsymbol{\Sigma}_i^{\circ\phi} & \boldsymbol{\Sigma}_i^\circ \end{bmatrix}, \quad (2)$$

where  $\circ$  denotes the spatial data. These parameters relate to those in (1). The block-diagonal elements of  $\boldsymbol{\Sigma}_i^\circ$  are  $\boldsymbol{\Sigma}_i^s$  and  $\{\boldsymbol{\Sigma}_i^{(j)}\}_{j=1}^P$ . The concatenation of  $\boldsymbol{\mu}_i^s$  and  $\{\boldsymbol{\mu}_i^{(j)}\}_{j=1}^P$  forms  $\boldsymbol{\mu}_i^\circ$ . The additional information is given by  $\boldsymbol{\Sigma}_i^{\phi\circ}$ , which is the covariance between the relative time  $\phi$  and the spatial data.

When reproducing a skill, a sequence of states first needs to be retrieved from the HSMM and the corresponding relative time variable computed. Then, Gaussian conditioning can be used on the augmented states to retrieve a more precise estimation of the observation  $\{\hat{\boldsymbol{\mu}}_i^\circ, \hat{\boldsymbol{\Sigma}}_i^\circ\}$  with

$$\hat{\boldsymbol{\mu}}_i^\circ(\boldsymbol{\xi}_{t,i}^\phi) = \boldsymbol{\mu}_i^\circ + \boldsymbol{\Sigma}_i^{\circ\phi} \boldsymbol{\Sigma}_i^{\phi-1} (\boldsymbol{\xi}_{t,i}^\phi - \boldsymbol{\mu}_i^\phi), \quad (3)$$

$$\hat{\boldsymbol{\Sigma}}_i^\circ = \boldsymbol{\Sigma}_i^\circ - \boldsymbol{\Sigma}_i^{\circ\phi} \boldsymbol{\Sigma}_i^{\phi-1} \boldsymbol{\Sigma}_i^{\phi\circ}, \quad (4)$$

given corresponding phase variable  $\boldsymbol{\xi}_{t,i}^\phi$ .

Fig. 2 illustrates the differences between the discrete HSMM states and the continuous distribution that can be retrieved with this technique. In Sec. 4.2 the advantages of this technique for synthesis purpose is shown. This keeps the simplicity of the time-based GMR approach while offering more flexibility in the skill representation (including loops, partial demonstrations and options in the movement). We can see this model as a set of GMR building blocks that can be rearranged at will.

### 3. Skills Reproduction

The reproduction of skills can be decomposed in three steps. They are re-executed as often as possible, in order to provide continuous adaptation to environment changes and perturbations. In our experiments, these steps run at approximately 100Hz. The three steps consist of: 1) from the encoding of demonstrations in multiple coordinate systems, we first need to find a compromise between the information encoded in the different coordinate systems; 2) we then have to address the problem of planning an appropriate sequence of states for the near future, according to what the robot needs to do, and based on the sensory data that will influence the transitions and durations of the MPs; and 3) the last step is to generate a smooth and safe controller for the robot according to the generated sequence of MPs.

The controller is given as a feedback between current robot state (position, velocity) and a desired state, with varying gains. The gains and targets are updated by the three aforementioned steps, and used by a controller running at a higher, constant frequency (500Hz in our experiments).

#### 3.1. Task-parametrized model

Our model represents redundantly the skill as Gaussian distributions in multiple coordinate systems (frames of reference). In order to bring together information coming from these multiple coordinate systems, the first step is to express all the distributions in a common coordinate system. It is achieved by applying linear transformations on the means and covariance matrices using the current position and transformation matrix describing the coordinate systems  $\{\mathbf{b}_{t,j}, \mathbf{A}_{t,j}\}_{j=1}^P$ , namely

$$\hat{\boldsymbol{\mu}}_{t,i}^{(j)} = \mathbf{A}_{t,j} \boldsymbol{\mu}_i^{(j)} + \mathbf{b}_{t,j}, \quad \hat{\boldsymbol{\Sigma}}_{t,i}^{(j)} = \mathbf{A}_{t,j} \boldsymbol{\Sigma}_i^{(j)} \mathbf{A}_{t,j}^\top. \quad (5)$$

These transformed distributions are multiple views of the same MP. The product of Gaussians can then be used to retrieve the exact compromise between them, with

$$\mathcal{N}(\hat{\boldsymbol{\mu}}_{t,i}, \hat{\boldsymbol{\Sigma}}_{t,i}) \propto \prod_{j=1}^P \mathcal{N}(\hat{\boldsymbol{\mu}}_{t,i}^{(j)}, \hat{\boldsymbol{\Sigma}}_{t,i}^{(j)}), \quad (6)$$

computed with

$$\hat{\boldsymbol{\Sigma}}_{t,i} = \left( \sum_{j=1}^P \hat{\boldsymbol{\Sigma}}_{t,i}^{(j)-1} \right)^{-1}, \quad (7)$$

$$\hat{\boldsymbol{\mu}}_{t,i} = \hat{\boldsymbol{\Sigma}}_{t,i} \sum_{j=1}^P \hat{\boldsymbol{\Sigma}}_{t,i}^{(j)-1} \hat{\boldsymbol{\mu}}_{t,i}^{(j)}. \quad (8)$$

The product of Gaussians computes the distribution that minimizes the distance to the mean of multiple Gaussians, weighted by the inverse of their covariance matrix, the precision matrix.

In [1] a detailed presentation of this approach and comparison to other approaches are given. The more naive approach to learn adaptable skills is to encode position and orientation of some objects, together with robot actuation, as Gaussian distributions. Using Gaussian conditioning on known state of the objects, it is then possible to retrieve an adapted robot actuation. Compared to this approach, the task-parametrized model exploits the coordinate systems structure to provide better interpolation and extrapolation performances, especially regarding the orientation of the path in the approaching phase.

This model is particularly relevant to cope with the different morphologies of the users. For the scenario of putting on a jacket sleeve, a coordinate system can be assigned to each important user's body-part (wrist, elbow, shoulder), see Fig. 3. Demonstrating this skill with different morphologies or user's poses, the model can learn that the beginning exhibits less variance in the coordinate system of the wrist, then in the coordinate system of the elbow and finally in the coordinate system of the shoulder.

#### 3.2. Generating a sequence of states

As the robot should be able to interact with the user and autonomously recover from perturbations, it is essential to continuously adapt and re-plan the sequence of

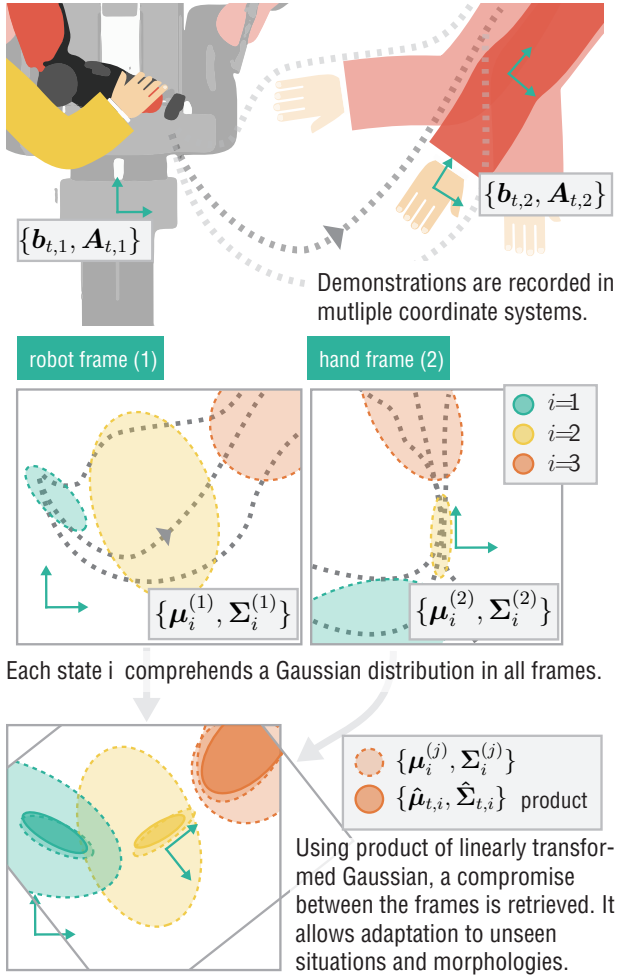


Figure 3: Illustration of the task-parametrized model for the dressing assistance skill.

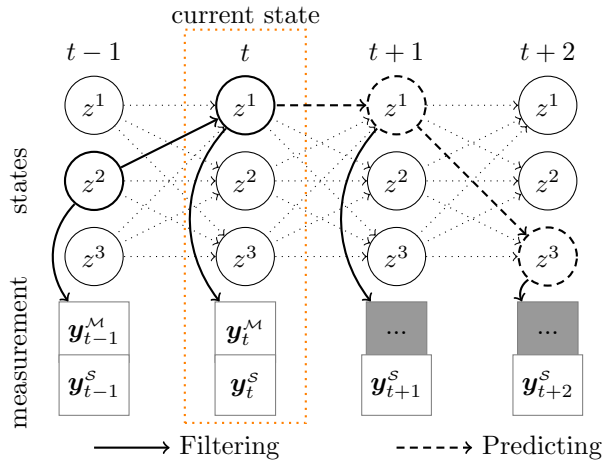


Figure 4: For interaction with the user and to cope with perturbations, the future sequence of states needs to be recomputed at each time step. This can be split in two related problems. (left) Filtering, to recover the current state from the past sensorimotor measurement. (right) Predicting, to plan a sequence of states, either based on interpolated sensory data or on temporal characteristics.

states. This can be decomposed in two closely related problems. The first is to find the latent sensorimotor state for the current time step  $t$ , denoted by the state indicator variable  $z_{n=t}$ . The second is to plan an adequate sequence of states for  $n = t + 1, \dots, t + T$  up to a planning horizon  $T$ . This plan is then cleared and recomputed at next update loop, in order to take into account varying sensory data  $\mathbf{y}^s$  and the new robot state. These two problems are illustrated in Fig. 4. They are commonly encountered in the field of HMMs and correspond to the filtering  $p(z_n | \mathbf{y}_1, \dots, \mathbf{y}_n)$  and prediction  $p(z_{n+M} | \mathbf{y}_1, \dots, \mathbf{y}_n)$  problems.

To compute these marginals, *forward messages*, defined as  $\alpha_n(z_n) \triangleq p(\mathbf{y}_1, \dots, \mathbf{y}_n, z_n)$ , are used [17]. They can be recursively computed with

$$\alpha_{n+1}(z_{n+1}) = p(\mathbf{y}_{n+1} | z_{n+1}) \sum_{z_n} \alpha_n(z_n) p(z_{n+1} | z_n) \quad (9)$$

for an HMM.

The first problem can then simply be solved with

$$p(z_n | \mathbf{y}_1, \dots, \mathbf{y}_n) = \frac{\alpha_n(z_n)}{\sum_z \alpha_n(z)}. \quad (10)$$

This means that, at each update loop, we only need to apply the recurrence relation (9) to get current forward messages, and then normalize them. The current state is then retrieved with

$$z_t = \arg \max_z \alpha_t(z), \quad (11)$$

as a maximum-likelihood solution.

Two approaches are possible for the prediction problem. The first is to compute recursively the forward messages, starting from current messages  $\alpha_t(z)$ , up to  $n = t + T$ , end of the planning horizon, based only on transition information

$$\alpha_{n+1}(z_{n+1}) = \sum_{z_n} \alpha_n(z_n) p(z_{n+1} | z_n), \quad (12)$$

as the observations are not yet available.

The second approach, adopted in this work, is to consider that the sensory data up to  $n = t + T$  is known from current observation. In this case, forward messages are computed with (9) where

$$p(\mathbf{y}_{n+1} | z_{n+1}) = \mathcal{N}(\mathbf{y}_{n+1}^s | \boldsymbol{\mu}_{z_{n+1}}^s, \boldsymbol{\Sigma}_{z_{n+1}}^s), \quad (13)$$

meaning that only the sensory data is used. This is particularly relevant for skills with low or predictable dynamics, so that we can consider the sensory part as constant with  $\mathbf{y}_{n+M}^s = \mathbf{y}_n^s$  for  $M > 0$ . It should be noted that this assumption is only considered within the planning horizon, which is discarded and updated at each update loop, following a receding horizon control principle. The predicted sequence of states is then retrieved by taking the most probable state for each time step as in (11).



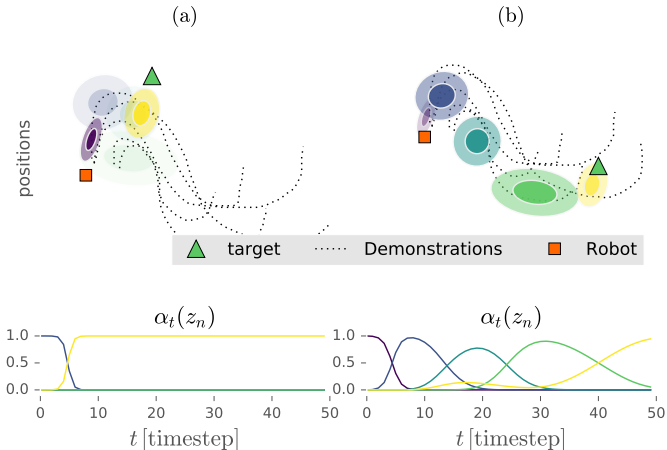


Figure 5: Example of skill requiring a wide range of adaptation. The robot should go to the target while doing an S shape. (a) When the target is close to it, only two MPs (displayed in purple and yellow) are sufficient to adequately encode the trajectory. Eq. (13) gives low probability of the other MPs in this situation. (b) When the target is further, more MPs are required.

The first approach then tries to predict changes in the sensory data as well, based on transition distribution of the model. In dressing assistance applications, where the sensory data relate to the human user, this will typically let the robot anticipate the human’s behavior. In the risk-sensitive context of dressing assistance, this is not a particularly desirable feature, and we prefer the more conservative behavior brought by the second approach, where the robot can be slightly late, but can avoid undesired parasite anticipatory motions.

Fig. 5 and 6 illustrate how the object position (sensory information) influences the forward messages through (13), for planning a sequence of state. In Fig. 6, an obstacle avoidance behavior is learned. When no obstacle is present, the simple trajectory can be represented by two MPs, but when in front, two additional MPs are required to appropriately encode the avoidance trajectory. Eq. (13) influences the forward message because of the low probability that an avoidance MP was observed in the demonstration without an obstacle in front. It should be noted that this behavior was learned in the EM process without explicitly indicating the different options.

The same approach holds for HSMM. Only the forward messages computation needs to be adapted. Following [21], (14) becomes

$$\alpha_{n+1}(z_{n+1}) = \sum_{z_n} \sum_{d=1}^{\min(D, n)} \alpha_n(z_{n+1-d}) p(z_{n+1}|z_n) \cdot p(d|z_{n+1}) \prod_{s=n+1-d}^{n+1} p(y_s|z_{n+1}), \quad (14)$$

where  $p(d|z_{n+1}) = \mathcal{N}d | \mu_{z_{n+1}}^D, \Sigma_{z_{n+1}}^D$  and  $D$  bounds the computation to a finite number of iterations.

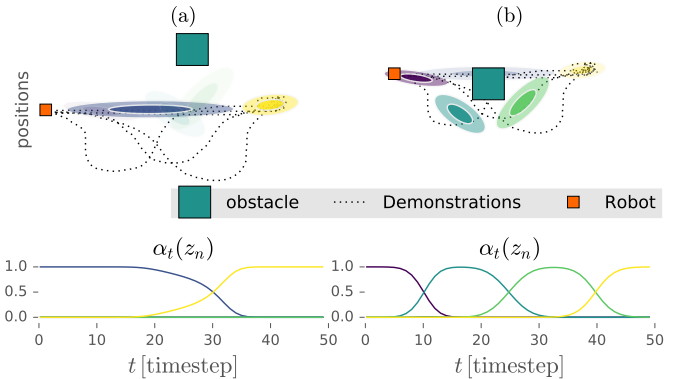


Figure 6: Example of a learned obstacle avoidance behavior. The obstacle position influences the forward messages (bottom graphs) through (13). (a) Without an obstacle in front, two MPs were not used during the demonstrations and thus have low probabilities in (13). (b) They were observed when the obstacle is in front and are thus activated in this situation (displayed in light blue and green).

### 3.3. Controller

From the retrieved sequence of states, we compute the relative time sequence within each state. Using our augmented model and the GMR approach with relative time, it is possible to retrieve a set of normal distributions  $\{\hat{\mathbf{x}}_t, \hat{\Sigma}_t\}_{t=1}^T$  representing state targets, up to a planning horizon  $T$ . As described in [2], the variability of the target  $\hat{\Sigma}_t$  is exploited to derive a minimal intervention controller. A cost function

$$c = \sum_{t=1}^T (\hat{\mathbf{x}}_t - \mathbf{x}_t)^\top \hat{\Sigma}_t^{-1} (\hat{\mathbf{x}}_t - \mathbf{x}_t) + \mathbf{u}_t^\top \mathbf{R}_t \mathbf{u}_t, \quad (15)$$

used in linear quadratic tracking (LQT), is defined. It penalizes distance from the target distributions and control input  $\mathbf{u}_t$ , weighted by  $\mathbf{R}_t$ . In the presented application,  $\mathbf{u}_t$  is defined as a force command in operational space. The sequence of input  $\mathbf{u}_t$  minimizing this cost function can be retrieved as a feedback controller

$$\mathbf{u}_t = \hat{\mathbf{K}}_t^p (\hat{\mathbf{x}}_t^p - \mathbf{x}_t^p) + \hat{\mathbf{K}}_t^v (\hat{\mathbf{x}}_t^v - \mathbf{x}_t^v), \quad (16)$$

where  $\hat{\mathbf{K}}^p$ ,  $\hat{\mathbf{K}}^v$  are respectively feedback gains on  $p$  position and  $v$  velocity. In order to solve this minimization problem, the relation between  $\mathbf{x}$  and  $\mathbf{u}$ , namely the dynamic model of the robot, should be expressed as

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t. \quad (17)$$

This relation can be retrieved either by linearizing the dynamics of the system or by considering a simpler virtual system, such as a double integrator [20]. The force command  $\mathbf{u}_t$  is used to control the robot through the torque command

$$\boldsymbol{\tau}_t = \mathbf{J}^\top(\mathbf{q}_t)(\mathbf{q}_t) \mathbf{u}_t + \mathbf{g}(\mathbf{q}_t) + (\mathbf{I} - \mathbf{J}^\top(\mathbf{q}_t)\mathbf{J}^{\dagger}(\mathbf{q}_t)) \mathbf{K}(\hat{\mathbf{q}} - \mathbf{q}_t) \quad (18)$$

where  $\mathbf{J}(\mathbf{q}_t)$  is the robot end-effector Jacobian at current joint space configuration  $\mathbf{q}_t$ , and  $\mathbf{g}$  is the gravity compensation torques. The last term is a null space control term [8] where  $\dagger$  denotes the pseudo-inverse,  $\hat{\mathbf{q}}$  is a preferred neutral pose and  $\mathbf{K}$  is a tracking gain (scalar in our implementation).

#### 4. Experiments and results

The proposed approach is tested with the task of helping a user to put on the sleeve of a jacket, see Fig. 1. The robot should learn from demonstration how to hold the jacket, how to comfortably bring the opening of the sleeve toward the user’s hand, and when it is appropriate to initiate the dressing movement once the hand is in the sleeve.

First, the task is simulated using synthetic data. The reproductions are performed assuming a perfect model of the robot, so that the quantitative evaluation only reflects the performance of the model in simulation condition.

Then, the approach is tested on a real robotic platform, using the Baxter robot from Rethink Robotics with two 7 DOFs arms. We consider a dressing experiment in which  $\{\boldsymbol{\mu}_i^{(j)}, \boldsymbol{\Sigma}_i^{(j)}\}_{j=1}^2$  encode the position of the robot end-effector from the coordinate system of the robot ( $j = 1$ ) and from a coordinate system related to the hand of the user ( $j = 2$ ).  $\{\boldsymbol{\mu}_i^s, \boldsymbol{\Sigma}_i^s\}$  encode the position of the user’s hand in the robot coordinate system. The same task is then used to evaluate the benefits of the relative time encoding for trajectory synthesis (Sec. 2.3).

Then, we present an experiment where the robot should learn a task consisting of a periodic and a discrete motion, where the robot needs to determine when to switch from one to the other based on the user’s behavior.

Finally, we consider the assistive task of putting on a shoe. The robot, holding the shoe, should in this case bring it to the user’s foot. An obstacle is placed between the starting position of the robot and the foot, so that the robot should avoid it.

##### 4.1. Experiment with Synthetic Data

*Dressing.* Several demonstrations are performed for different positions of the hand, see Fig. 7. Two different options are shown corresponding to the desired behavior. These are not explicitly indicated to the robot. The HSMM is trained on these demonstrations using expectation-maximization. In Fig. 8 the Gaussian distributions from the resulting states are shown. One state is used to describe the option where the hand is not reachable while the others are used to describe the dressing movement.

Fig. 9 shows reproduction attempts for two different positions of the user’s hand. The evaluation consists of comparing our method with the HSMM encoding only motor commands, by testing if it performs as well as the former for time-dependent tasks. The results show that the

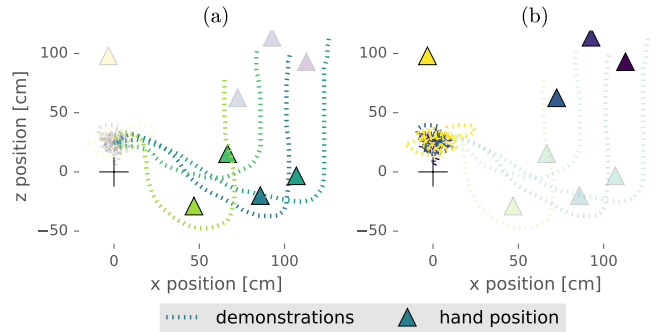


Figure 7: Set of demonstrations for learning the dressing task. Two different options are shown but not explicitly indicated to the robot. The color code indicates the correspondence between the demonstrations and the hand position. (a) Dressing option. (b) Waiting option (i.e., the hand is not reachable).

Table 1: RMSE between demonstrations and reproduction for option (a) of the simulated dressing task. (*only motor*) denotes the HSMM with only transition and duration information. (*sensorimotor*) denotes the model presented in this paper. (*testing set*) denotes leave-one-out cross-validation. This shows that for time-dependent skills, the original motor HSMM is not degraded.

RMSE [cm]	training set	testing set
only motor	$1.37 \pm 0.21$	$1.51 \pm 0.16$
sensorimotor	<b><math>1.26 \pm 0.31</math></b>	$1.48 \pm 0.21$

proposed additional feature in the model could be added without degrading the original model.

As comparison metric, the root-mean-square error (RMSE) between the reproduction and the human demonstration in the same situation is computed. For the testing set, we used a leave-one-out evaluation, where the reproduction is tested in an unseen situation. As time-dependent task, we used the dressing movement of Fig. 9(b).

For this particular task and model, the results show that there is no significant difference for the testing set and the results are slightly better for the training set with our model (see Table 1). The analysis of Fig. 8-(c) provides an explanation. We can see that the sensory distributions are almost the same for the states corresponding to the dressing movement. Thus, the probability  $\mathcal{N}(\mathbf{y}_n^s | \boldsymbol{\mu}_{z_n}^s, \boldsymbol{\Sigma}_{z_n}^s)$  is almost the same for all  $z$ , which induces negligible changes in the forward messages computation. If important degradation occurs with the sensorimotor model, regularization on the sensory dimensions is possible. It will equalize the marginal probabilities of the sensory dimensions, leading to a “only-motor HSMM”.

Overfitting may also occur, where in the extreme case, separate states are used for each situation. In this case, the model acts as a library of skills, where no generalization occurs in-between the demonstrations. Fig. 10 illustrates this problem.

Such an overfitting is also present and even more problematic in a standard task-parametrized model. In sensori-



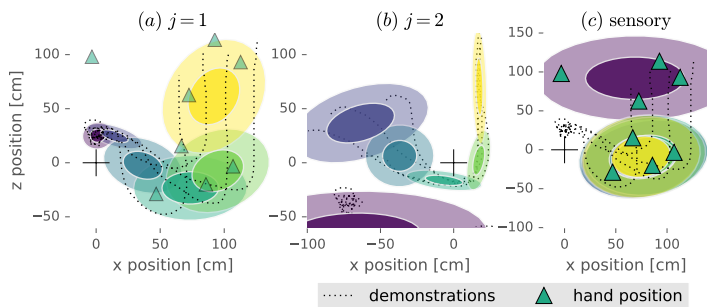


Figure 8: Gaussian distributions from the different states learned by expectation-maximization. The color code indicates the different states. (a)  $\{\mu_i^{(1)}, \Sigma_i^{(1)}\}$  position of the end-effector in the robot coordinate system. (b)  $\{\mu_i^{(2)}, \Sigma_i^{(2)}\}$  position of the end-effector in the user's hand coordinate system. (c)  $\{\mu_i^S, \Sigma_i^S\}$  position of the user's hand in the robot coordinate system.

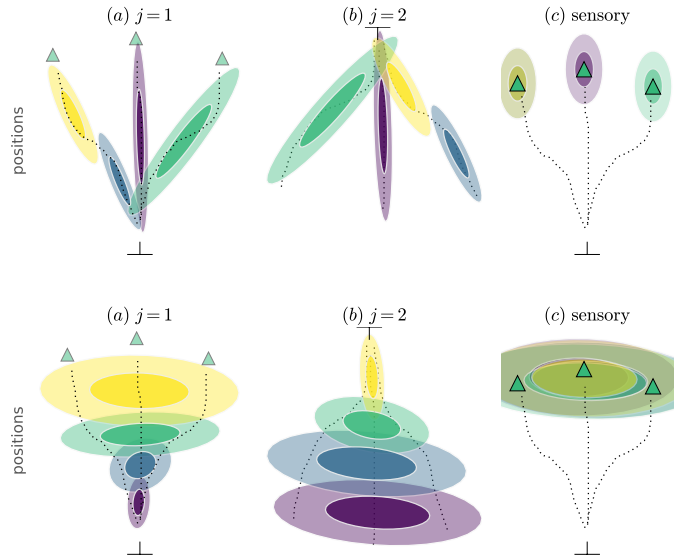


Figure 10: Illustration of over-fitting problems that may occur with sensorimotor and task-parametrized model. (a-b-c) See legend of Fig. 8. *Top*: Overfitting with each of the three demonstrations encoded using separate states, where generalization cannot occur in this case (the sensory dimensions indicate that three clearly distinct situations were detected). *Bottom*: With proper regularization or initialization, the model converges to a desired solution (the situations are considered as the same, and generalization occurs).

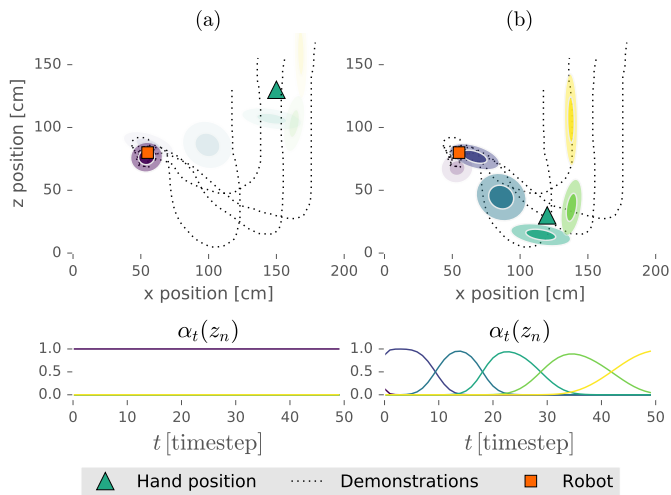


Figure 9: Illustration of reproduction for the dressing task. The transparency indicates the marginal probabilities of the states given the observation of the user's hand. (a) Low probability of states corresponding to the dressing movement when observing the hand. The forward variable (bottom graph) computed with this information indicates that the robot should stay in the waiting state. (b) Same probability of each other states when the hand is lower. The forward variable (bottom graph) indicates that the robot should follow the depicted sequence of states.

motor model, encoding the position of the object in  $\mathcal{N}(\mathbf{y}_n^S | \mu_{z_n}^S, \Sigma_{z_n}^S)$  allows the system to indicate which states are more probable given the situation. Thanks to this information, it is still possible to retrieve the desired trajectory in known situations.

These problems could be avoided, for example, by using regularization terms in the sensory distributions or by proper HMM parameters initialization. In this work, Tikhonov regularization is used for the observation parameters, where an isotropic covariance is added to the covariance of the data with

$$\Sigma \leftarrow \Sigma + \sigma^2 \mathbf{I}, \quad (19)$$

where  $\sigma$  is different for position, velocity or sensory data. This regularization is added at each M step of the EM fitting process. Later in this work, an analysis of the effect of regularization parameters is proposed.

#### 4.2. Experiment with Real-World Data

*Garment.* The dressing assistance was then performed on the robotic platform. We used Alvar, an open source AR tag tracking library, to get the position of the user's hand. A wristband with a  $6 \times 6$  cm tag was worn by the user. The camera from the robot free hand was used, such that, knowing the kinematic model of the robot, the position of the hand could be retrieved in a global coordinate system. Fig. 12 illustrates the setup.

In Fig. 11 the demonstration and reproduction processes are shown. In total, 8 demonstrations of putting on

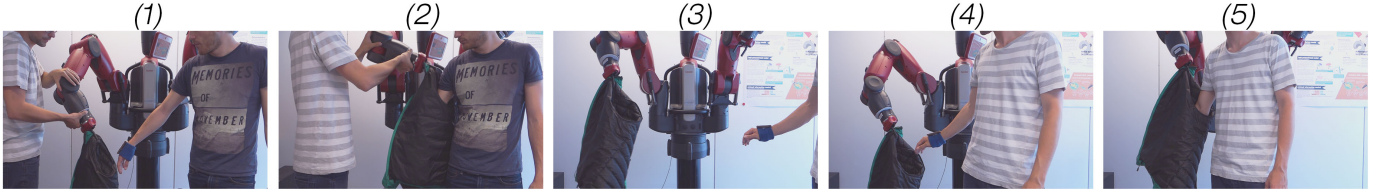


Figure 11: Demonstration and reproduction process of the dressing task on Baxter robot. (1) *Demonstrations*: interaction when the user brings his hands towards the robot, which initiates the same movement. (2) Putting on the sleeve. (3) *Reproduction*: The user approaches the robot. (4) The robot initiates a movement down, toward the user’s hand. (5) Once the hand is at the entrance of the sleeve, the robot initiates the movement upwards, moving toward the shoulder.

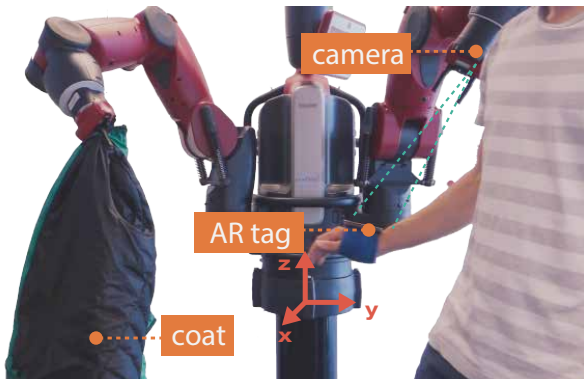


Figure 12: Setup for the dressing assistance. The robot holds a jacket with the right arm and tracks the human’s hand with the camera embedded in the other arm. An AR tag on a wristband is used.

a jacket sleeve in different positions, 3 demonstrations of the interaction when the human brings his hand toward the robot and a demonstration of the whole task were performed.

We compared the proposed method to two other approaches. The first is to consider the position (or velocity) of the robot as only dependent on the position of the user’s hand. Namely, computing  $p(\{\xi_t^{(j)}\}_{j=1}^P | \xi_t^S)$  with GMR, and using the retrieved Gaussian distribution as target in the LQT formulation. The problem with this approach is that given a sensory observation  $\xi_t^S$ , the target of the robot is a single Gaussian distribution. Even if we augment this distribution with new information (e.g. velocity or acceleration), we lose the capability of HSMM to handle multiple distributions and time-dependent behavior. In the dressing scenario, the behavior corresponding to the reaction to the user’s hand coming close is well encoded with this model. However, for the dressing movement, the robot stops at the most probable position which is near the middle of the arm; the time-dependent skills required to follow the configuration of the user’s arm is lost.

The other test we considered is to compute the sequence of states from the HSMM only based on the transition and duration probabilities. This corresponds to a pure time-dependent model where the robot behavior is defined by  $p(\{\xi_t^{(j)}\}_{j=1}^P | t)$ . The results show that, as expected, the

model is blind to the environment changes (sensory information, perturbations) and is thus not able to synchronize well with the human user.

*Benefits of relative time encoding.* To evaluate the benefits of relative time encoding, the same setup and task as in the previous experiment are used. As we are interested only in the actuation, no sensory data is used this time and the user’s arm is considered to be, from the beginning, in a pose to be dressed. The motion is regenerated based only on duration and transition distributions, thus providing a purely time-dependent behavior. For this experiment, the velocity of the end-effector is added to the observation model, providing during reproduction a velocity feedback bringing back the robot to velocities observed during the demonstrations.

We compare the synthesis of motion using: (a) the relative time encoding with velocity; (b) the discrete HSMM states without velocity; and (c) the discrete HSMM states with velocity. Resulting trajectories are displayed in Fig. 13.

In (b), the trajectory exhibits heavy stepwise motions, with velocity peaks when switching between states. To get smooth trajectories with this technique, a very fine tuning of the control cost weight  $\mathbf{R}_t$  from (15) needs to be performed, so that forces (thus accelerations) are limited. Such fine tuning interferes with the first goal of this cost: controlling the robot compliance for safe interaction. indeed, tuning  $\mathbf{R}_t$  so that a smooth trajectory is retrieved will usually result in the robot becoming overly compliant. In (c), feedback on the velocity allows for smoother trajectories and closer reproduction of the dynamic features (velocity). However, the stepwise characteristics are still perceived. The relative time encoding in (a) allows for a more precise and smooth encoding of the target distribution, using Gaussian conditioning. The smoothness relies much less on  $\mathbf{R}_t$ , which can be freely modified to obtain a desired compliance. The dynamic features are also better reproduced.

*Learning skill structure.* In this experiment, we demonstrate that our model can learn a skill with both discrete and periodic movements. We also show how regularization can influence the structure discovered by the learning

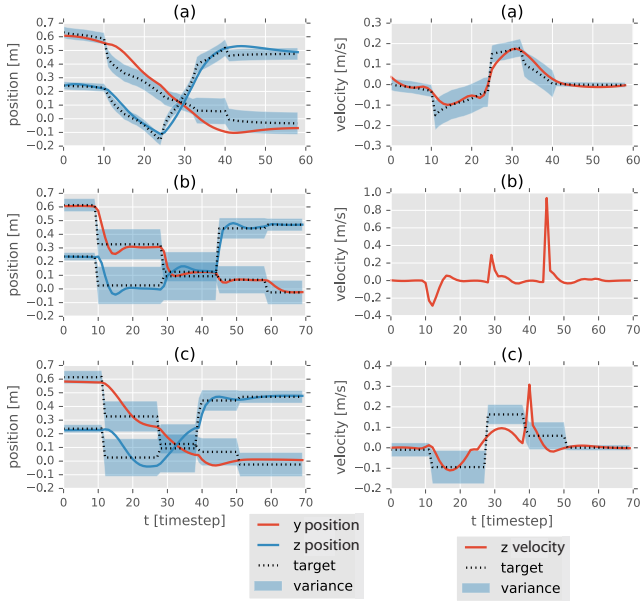


Figure 13: Comparison of the retrieved targets and the generated trajectories using: (a) the relative time encoding and velocity; (b) discrete HSM states without velocity; and (c) discrete HSM states with velocity. *Left*: Y and Z position profiles. *Right*: Z velocity profile.

process, which we believe to be a very important indicator of its success. Fig. 14 displays, as a graph, different structures of skills that can be learned and which are encoded in the transition matrix. The nodes represent the HSM states and the arrows indicate that a transition between the connected states exists.

This solution is contrasted to the model in [12], sharing links with our approach, but where the use of linear basis function induces an explicit dependency between state variables  $\mathbf{y}$  and a time or phase variable  $z$ , global to the trajectories.

Fig. 14(a) illustrates a typical structure of skills in [12], where a mixture of robot-human trajectory distributions is learned. In this case, the trajectory to execute is only selected at the beginning, given the observation of an incomplete human trajectory. This dependency can restrict the variety of skills that can be learned. In particular, for human-centered assistive applications, the robot needs to cope with partial demonstrations, time-dependent and time-independent behaviors, multiple paths or options, as well as cyclic or discrete movements. An HSM representation can handle these different scenarios. Fig. 14(b) illustrates the HSM structure that we propose in the context of human-robot interaction. The structure represents in this case a skill in which the robot should perform a periodic task and, once the user comes close to the robot, should complete the loop and switch to another non-periodic task.

The same setup as in the dressing experiment was used, except that we added the velocity of the end-effector in the

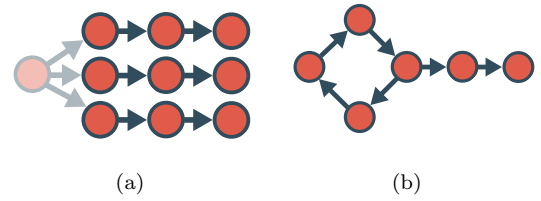


Figure 14: Illustration of the HSM structure as a graph. Each node represents a state and an arrow indicates that there is a transition between two states. (a) represents the typical structure of skills in the case of trajectory distributions (see e.g. [12]), where the trajectory to reproduce is decided at the beginning of the movement, based on the observation of partial human trajectory. (b) illustrates a different structure, potentially containing loops and options that can be represented by an HSM.

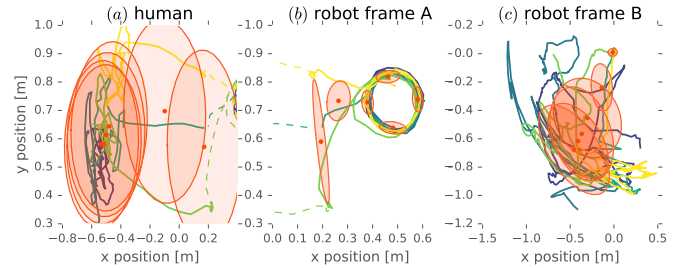


Figure 15: Demonstrations and learned model of a task requiring periodic and discrete motions. The user moves randomly in two demonstrations, and approaches the robot in the three other demonstrations. (a) Trajectories of the user's hand. (b) Trajectories of the robot end-effector in a global reference frame. The periodic motion is demonstrated, as well as three handover motions. (c) Trajectories of the robot end-effector in the coordinate system of the human's hand. The three trajectories (corresponding to handover motions) terminate in a precise location (top-right).

model. The task was demonstrated 5 times. In two demonstrations, the periodic motion was shown by repeating multiple circular movements with the robot arm. Meanwhile, the user was moving randomly at approximately 1 meter distance. In the three other demonstrations, we proceeded in the same way but, at some point, the user brought his hand close to the robot, and the robot was taught to bring its end-effector toward it (handover motion). The demonstrations ended when the robot end-effector was just above the user's hand. The demonstrations are shown in Fig. 15. In (a), the human trajectories are displayed (corresponding to the sensory dimensions). (b) and (c) show the trajectory of the end-effector in the global reference frame and the coordinate system of the human's hand, respectively.

The model is learned from these demonstrations without any information about the structure of the skill. In Fig. 15, the ellipses indicate the observation model of the learned HSM, for sensory and actuation parts, in two coordinate systems. The learned structure is depicted in Fig. 14(b) and corresponds to the skill we wanted the robot to learn. However, since the result of the EM process depends on the initialization of the parameters, it may not

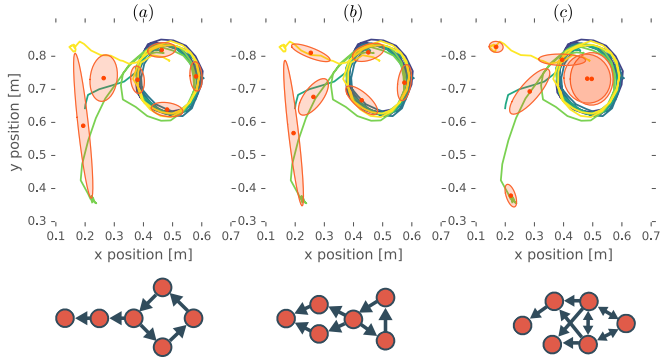


Figure 16: Different classes of structure to which the learning process can converge. (a) The structure corresponds to the desired skill (with generalization between the demonstrations). (b) A part of the skill was divided in two paths, providing satisfying reproduction in known situations but limited generalization capability. (c) The structure does not correspond to the desired skill.

always converge to the model with an adequate structure. For this experiment, with 980 datapoints of 15 dimensions, convergence is fast and takes about 3 seconds, so multiple learning processes with different initializations can be run. By starting from 20 different initializations, EM converges to three different classes of structure (three local optima), depicted in Fig. 16. In (a), all the demonstrations corresponding to the same part of the skill are assigned to the same sequence of states. For the particular examples of the demonstrations, a single structure is extracted. In (b), since the handover motion was divided in two paths, the robot would be able to perform the task in known situations, but would not be able to correctly generalize it to new situations. In (c), the structure does not correspond to the original structure of the skill: the robot would not be able to reproduce it well, even in a known situation.

We now analyze how regularization can influence the discovered structures. As regularization, we used  $\sigma_p = 0.01[m]$  for position data,  $\sigma_v = 0.05[m/s]$  for velocity data, as described in (19). We let the regularization for the sensory data vary between  $\sigma_p = 0.025[m]$  and  $\sigma_p = 0.225[m]$ , and ran 20 initializations for each conditions. For each initialization, we classify the discovered structure between (a), (b) and (c) as presented in Fig. 16. For each regularization parameters, the proportion of convergence to the different classes of structure is reported in Fig. 17. Decreasing the regularization over sensory dimensions would tend, in a first place, to increase the proportion of class (b) and then of class (c). This arises from the fact that our model is fully generative and tends to encode precisely the user trajectory at the expense of the robot actuation. This is similar to what happens in Fig. 10, where all demonstrations are treated as separate options, and no generalization occurs.

This problem can be solved in two different ways. First, as the learning process is very fast, multiple initializations can be run and the structure providing the best generaliza-

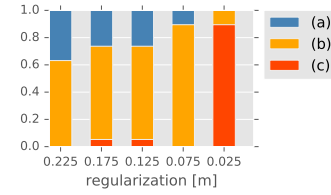


Figure 17: Proportion of convergence to the classes of structures shown in Fig. 16, for different levels of regularization in the sensory dimensions. The model parameters were randomly initialized in each trial, and updated by an EM process until convergence.



Figure 18: Variant (a) of the task to put on a shoe while facing an obstacle. The obstacle is kept static. The robot learns to avoid it from the right or the left side based on what is more convenient given the position of the user's foot.

tion can be chosen with a testing set (e.g. one demonstration left apart). Secondly, more demonstrations could be used, so that the continuity between the different phases of the task would be emphasized. It would help the system generalize instead of considering each demonstration as a separate path. In this experiment, five demonstrations is a very low number, since the robot needs to learn both the periodic and discrete parts of the motion, how to switch between them, as well as how to adapt to the position of the user's hand in the handover movement. Additional tests by increasing this number to 8-10 confirmed this point, with substantially improved results compared to Fig. 17.

*Putting on a shoe with obstacle.* Two variants are explored in the task of putting on a shoe while facing an obstacle. In the first variant, the obstacle is static and the foot can move to different sides of the obstacle, see Fig. 18. The robot should learn to avoid it from the right or left side depending on what is more convenient. The robot should also wait for the foot to be in a reachable zone. For this task,  $j = 1$  is the robot coordinate system and  $j = 2$  is a coordinate system attached to the user's foot. Each Gaussian  $\mathcal{N}(\mu_i^s, \Sigma_i^s)$  encodes the position of the user's foot in the robot coordinate system.

In the second variant of the experiment, the foot is considered as fixed and the obstacle can move, see Fig. 19.



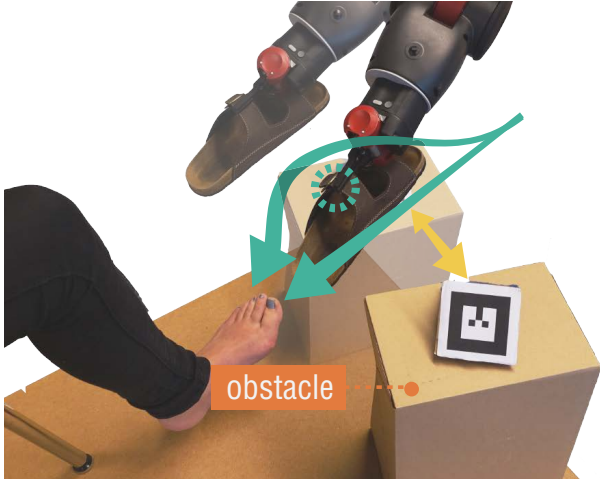


Figure 19: Variant (b) of the task to put on a shoe while facing an obstacle. The foot of the user is static. Depending on the position of the obstacle, the robot should either take a straight path to the foot or a more complex one, involving obstacle avoidance. The robot learns to distinguish between these two cases by estimated MPs that are specifically suited for avoiding the obstacle.

Depending on the position of the obstacle, the robot should then either make a straight motion to the foot or a more complex one, involving obstacle avoidance. For this task,  $j = 2$  is a coordinate system attached to the obstacle. Note that the above scenarios could alternatively be combined together by considering  $P = 3$  coordinate systems (robot, obstacle and foot).

For the first task, 11 demonstrations were collected in total: 5 for avoiding the obstacle on one side and 4 on the other side. Two demonstrations also show that nothing should be done if the foot is not at a reachable distance. We recall that this information is not explicitly given for learning. We tested the approach by leave-one-out cross-validation with  $K = 7$  Gaussians. The results show that the desired behavior was acquired (see Fig. 20). The robot stays still when the foot is on the right, and avoids the obstacle either on one side or the other. However, we noticed that with one of the situation, an ambiguity remained: the reproduction did not avoid the obstacle from the same side as that of the demonstration. After careful analysis, we noticed this was due to the fact that one of the estimated Gaussian could not clearly encode the decision boundary.

For task (b), a quantitative evaluation is presented. The root-mean-square error (RMSE) between demonstrations and reproductions is used as a metric. Reproductions are performed on a simulated robot with known dynamic model. As reference, the mean RMSE between multiple demonstrations in the same situation is computed. It indicates the typical error allowed in the task. Thus, we use

$$\text{relative RMSE} = \frac{\text{RMSE}_{(\text{demo-repro})}}{\text{RMSE}_{(\text{demos})}}. \quad (20)$$

This task may also be encoded with a standard task-parametrized approach [1], which was used for baseline

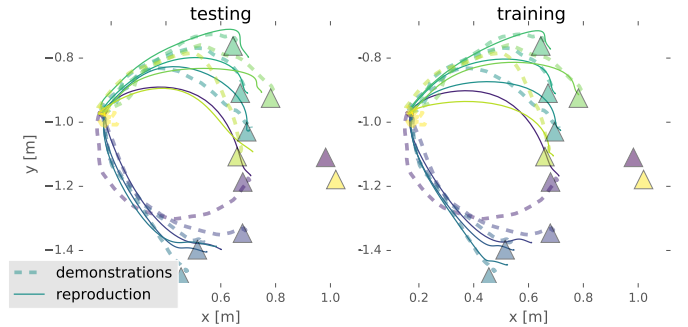


Figure 20: Demonstrations and reproductions for task (a) (see Fig. 18) with  $K = 7$  Gaussians. (left) Leave-one-out cross-validation. (right) All the demonstrations are used to train the model.

Table 2: Relative RMSE between demonstrations and reproduction for task (b). (*Sensorimotor*) denotes the model presented in this paper. (*TP model (a)*), standard task-parametrized model [1] with spatial initialization of HSMM parameters. (*TP model (b)*), standard task-parametrized model with time-dependent initialization of the parameters, resulting in a *left-to-right* HSMM. (*testing set*) leave-one-out cross-validation.

relative RMSE	training set	testing set
sensorimotor	<b>1.26</b> $\pm$ 0.45	<b>1.48</b> $\pm$ 0.47
TP model (a)	2.69 $\pm$ 1.96	2.99 $\pm$ 1.65
TP model (b)	1.35 $\pm$ 0.49	2.23 $\pm$ 1.59

comparison. Quantitative results are presented in Table 2.

Two different methods for initializing the HSMM parameters were used in the task-parametrized model. The first method (a) is based purely on spatial data, by using k-means on the whole dataset. In this case, the HSMM converges to a structure with two distinguishable paths: the first corresponds to the case where no obstacle is present (the robot should just perform a straight motion) and the second corresponds to the path with obstacle avoidance. If this structure encodes the motor actuation well, the standard task-parametrized does not encode the influence of the task parameters over the sequences of states. It leads to very poor results as the robot is unable to determine which path to take in the HSMM. The second initialization method considered (b) uses a time-dependent initialization, where each demonstration is divided in  $K$  temporal bins,  $K$  corresponding to the number of HSMM states. The observation model of state  $i$  is initialized with the data in the  $i$ th bin of each demonstration. It results in a *left-to-right* HSMM, where each demonstration is encoded with the same sequence of states, canceling the problem in (a) of adapting the sequence of states.

Fig. 21 shows the reproductions with time-dependent initialization. In this case, the motions with and without the obstacle are considered as the same option. It prevents the robot from learning that there is a dependency

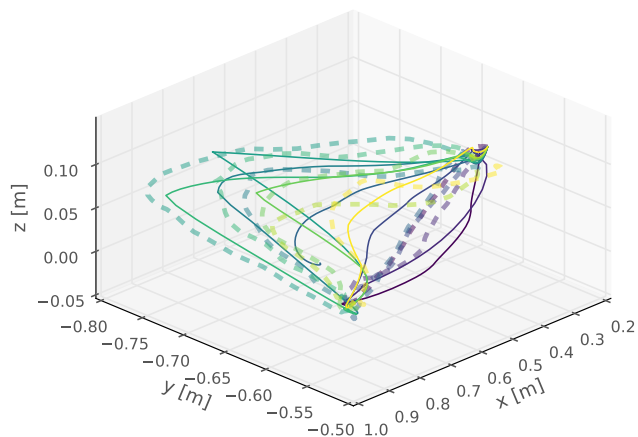


Figure 21: Demonstrations and reproductions for task (b) (see Fig. 19) with a standard task-parametrized model with  $K = 8$  states. A time-dependent initialization of the HSMM parameters is used. With this initialization, the model is a *left-to-right* HSMM, in which a unique sequence of states is considered. The motion with and without the obstacle are thus considered as the same option. Such approach results in less precise reproduction than the proposed sensorimotor model (see Fig. 22), where the skill is encoded as having two paths. With two paths, the model encodes that there is a dependency between the position of the obstacle and the motion only when the obstacle is nearby. In contrast, with a *left-to-right* HSMM, the two behaviors are not distinguished, and the dependency exists even when the obstacle is far, resulting in the robot being drawn towards the obstacle (purple trajectories).

between the position of the obstacle and the motion only when the obstacle is nearby. Even when the obstacle is outside of the trajectory, it slightly pulls the robot toward it. In contrast, the proposed sensorimotor HSMM splits this task in two clearly distinct options, resulting in a clear non-dependency to the obstacle when it is far (see Fig. 22). The benefit of encoding a model with two options is quantified in Table 2. The sensorimotor model provides better results, especially regarding generalization, compared to the *left-to-right* task-parametrized model (b).

## 5. Conclusion and Future work

In this paper, we presented a method to encode assistive tasks with hidden semi-Markov model. The parameters of this model are learned by expectation-maximization on a set of unstructured and unlabeled demonstrations, allowing quick and intuitive re-programming of skills, with adaptation to new situations. We showed that an explicit modeling of state durations in HSMM can encode the time dependence or independence of the learned behavior primitives. By considering additional sensory data in the HSMM, we showed that forward messages could be used to generate adaptive behaviors, where different sequences of states are planned online according to the current situation. A minimal intervention control strategy based on this representation was then derived, by relying on a linear quadratic tracking formulation, resulting in a safe interaction with the user.

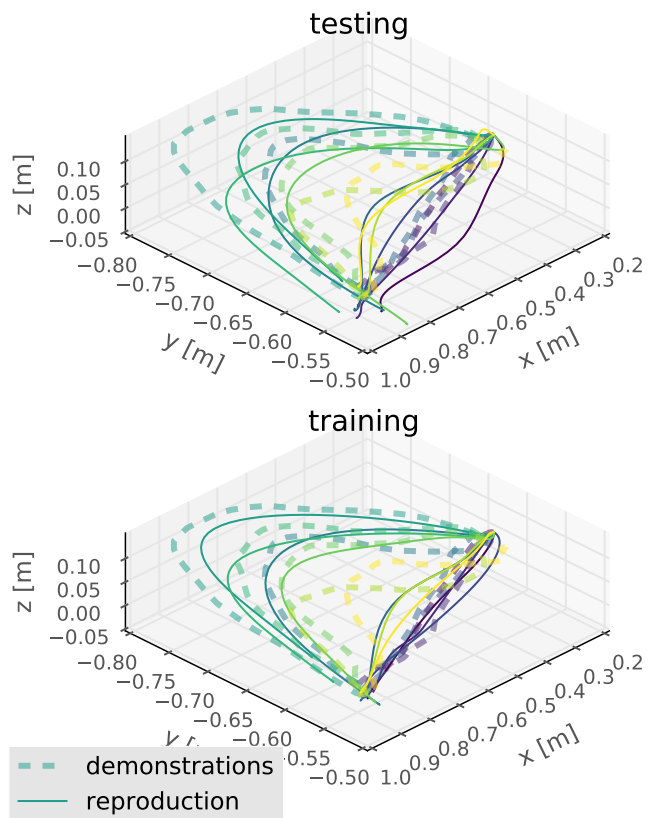


Figure 22: Demonstrations and reproductions for task (b) (see Fig. 19) with  $K = 8$  states, using the proposed sensorimotor model. (top) Leave-one-out cross-validation. (bottom) All the demonstrations are used to train the model. The structure of the learned HSMM has two paths: one for the straight motion and one for the obstacle avoidance. It results in notably better regeneration than the standard task-parametrized model, see Fig. 21 and Table 2 for quantitative evaluation.



By coupling the proposed approach with a task-parametrized model, we then showed that this method can be used by the robot to adapt its behavior to the user's pose or morphology. We showed the robustness of the approach in the context of obstacle avoidance, by achieving better generalization results than with the standard task-parametrized model.

This paper presented tasks exclusively defined by operational space position commands. Our future work will explore if the model can incorporate commands in other data spaces. In particular, operational space velocities, accelerations and forces will be considered together with joint space commands.

The HSMM used in this work was composed of sensorimotor states, which means that a normal distribution over motor commands is always linked to a unique perception distribution, forming together a semantic unit. This one-to-one relationship is somehow restrictive. For example, the same motor primitive may have multiple semantic meanings, corresponding to well-distinguished sensory states. Similarly, multiple motor states may correspond to a unique perception, as illustrated in the dressing assistance scenario with the jacket. We plan to investigate this issue in future work, with the goal of improving the flexibility of the model to enable shared sensory and motor distributions across states.

## Acknowledgements

This work was supported by the Swiss National Science Foundation through the I-DRESS project (20CH21-160856, ERA-net CHIST-ERA).

## References

- [1] S. Calinon. A tutorial on task-parameterized movement learning and retrieval. *Intelligent Service Robotics*, 9(1):1–29, January 2016.
- [2] S. Calinon, D. Bruno, and D. G. Caldwell. A task-parameterized probabilistic model with minimal intervention control. In *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, pages 3339–3344, Hong Kong, China, May-June 2014.
- [3] G. Chance, A. Camilleri, B. Winstone, P. Caleb-Solly, and S. Dogramadzi. An assistive robot to support dressing - strategies for planning and error handling. In *IEEE Intl Conf. on Biomedical Robotics and Biomechatronics (BioRob)*, pages 774–780, June 2016.
- [4] Y. Gao, H. J. Chang, and Y. Demiris. Iterative path optimisation for personalised dressing assistance using vision and force information. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 4398–4403, 2016.
- [5] Ben Amor H., Vogt D., Ewerton M., Berger E., Jung B., and Peters J. Learning responsive robot behavior by imitation. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 3257–3264, Nov 2013.
- [6] I. Havoutis and S. Calinon. Supervisory teleoperation with on-line learning and optimal control. In *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, Singapore, May-June 2017.
- [7] T. Hueber and G. Bailly. Statistical conversion of silent articulation into audible speech using full-covariance HMM. *Comput. Speech Lang.*, 36(C):274–293, March 2016.
- [8] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):243–253, February 1987.
- [9] B. Kulis and M. I. Jordan. Revisiting k-means: New algorithms via Bayesian nonparametrics. In *Proc. Intl Conf. on Machine Learning (ICML)*, pages 1–8, Edinburgh, Scotland, UK, 2012.
- [10] D. Lee and C. Ott. Incremental motion primitive learning by physical coaching using impedance control. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 4133–4140, Taipei, Taiwan, October 2010.
- [11] D. Lee, C. Ott, and Y. Nakamura. Mimetic communication model with compliant physical contact in human-humanoid interaction. *Intl Journal of Robotics Research*, 29(13):1684–1704, 2010.
- [12] G. J. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, O. Kroemer, and J. Peters. Probabilistic movement primitives for coordination of multiple human-robot collaborative tasks. *Autonomous Robots*, pages 1–20, 2016.
- [13] J. R. Medina, D. Lee, and S. Hirche. Risk-sensitive optimal feedback control for haptic assistance. In *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, pages 1025–1031, May 2012.
- [14] K. P. Murphy. Hidden semi-Markov models (HSMMs). Technical report, MIT, 2002.
- [15] S. Niekum, S. Osentoski, G. Konidaris, S. Chitta, B. Marthi, and A. G. Barto. Learning grounded finite-state representations from unstructured demonstrations. *The International Journal of Robotics Research*, 34(2):131–157, 2015.
- [16] A. Paraschos, C. Daniel, J. Peters, and G. Neumann. Probabilistic movement primitives. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 2616–2624. Curran Associates, Inc., 2013.
- [17] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77:2:257–285, February 1989.
- [18] K. Sugiura, N. Iwahashi, H. Kashioka, and S. Nakamura. Learning, generation, and recognition of motions by reference-point-dependent probabilistic models. *Advanced Robotics*, 25(6-7):825–848, 2011.
- [19] T. Tamei, T. Matsubara, A. Rai, and Shibata T. Reinforcement learning of clothing assistance with a dual-arm robot. In *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*, pages 2733–2738, Oct 2011.
- [20] A. K. Tanwani and S. Calinon. Learning robot manipulation tasks with task-parameterized semi-tied hidden semi-markov model. In *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, Stockholm, Sweden, May 2016.
- [21] S.-Z. Yu and H. Kobayashi. Practical implementation of an efficient forward-backward algorithm for an explicit-duration hidden Markov model. *IEEE Trans. on Signal Processing*, 54(5):1947–1951, 2006.