

Force-based Robot Learning of Pouring Skills using Parametric Hidden Markov Models

Leonel Rozo and Pablo Jiménez and Carme Torras

Abstract—Robot learning from demonstration faces new challenges when applied to tasks in which forces play a key role. Pouring liquid from a bottle into a glass is one such task, where not just a motion with a certain force profile needs to be learned, but the motion is subtly conditioned by the amount of liquid in the bottle. In this paper, the pouring skill is taught to a robot as follows. In a training phase, the human teleoperates the robot using a haptic device, and data from the demonstrations are statistically encoded by a parametric hidden Markov model, which compactly encapsulates the relation between the task parameter (dependent on the bottle weight) and the force-torque traces. Gaussian mixture regression is then used at the reproduction stage for retrieving the suitable robot actions based on the force perceptions. Computational and experimental results show that the robot is able to learn to pour drinks using the proposed framework, outperforming other approaches such as the classical hidden Markov models in that it requires less training, yields more compact encodings and shows better generalization capabilities.

I. INTRODUCTION

Robot learning from demonstration¹ (**LfD**) aims to endow robots with skills learned from human examples [1]. The teacher can demonstrate the task by teleoperation [2], kinesthetic teaching [3], or by using motion sensors or markers attached to the body [4]. During the demonstration phase, the perception system of the robot gathers all the information about the state of the robot and its environment. In this context, force sensing has recently opened the door to learning tasks where contact with the surroundings, objects and even with a human partner may occur.

Force-based data may enrich the perception system of a robot by providing additional information about the task, sometimes missed by other kind of sensors. For instance, Falco *et al.* used force measurements to improve the observation of the human hand motion [5], which may be of particular interest in LfD applications. Specifically, force inputs have been exploited in two kinds of scenarios, namely: (i) tasks involving contact and/or manipulation of objects, and (ii) physical human-robot interaction (**pHRI**). In [6], control policies based on goal-driven dynamical systems are learned to encode interaction force skills for a robot in

This research is partially sponsored by the European projects STIFF-FLOP (FP7-ICT-287728), IntellAct (FP7-269959), and the Spanish project PAU+ (DPI2011-27510). L. Rozo was supported by the CSIC under a JAE-PREDOC scholarship.

L. Rozo is research fellow at the Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genoa, Italy. leonel.rozo@iit.it. P. Jiménez and C. Torras are with the Institut de Robòtica i Informàtica Industrial CSIC-UPC, C. Llorens i Artigas, 4-6, 08028 Barcelona, Spain. {jimenez,torras}@iri.upc.edu.

¹Also known as programming by demonstration or imitation learning.

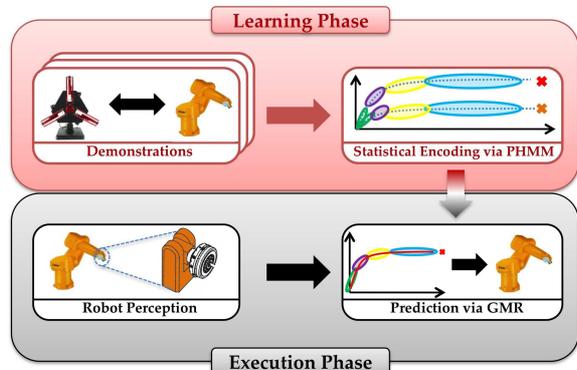


Fig. 1. Proposed learning framework. At the *demonstration* phase, the human provides examples of the pouring task by teleoperating the robotic arm using a haptic device. The force perceptions sensed over the course of the task, along with their initial value (i.e., the task parameter), are used to train a PHMM. At the *reproduction* stage, the robot carries out the pouring skill using GMR to retrieve joint-level commands given the force-torque inputs at each time step.

contact with compliant environments. Schmidts *et al.* [7] propose to learn human grasping skills from motion and force data, which are encoded by a hidden Markov model (**HMM**) and reproduced by a Gaussian mixture regression (**GMR**) later. A similar structure was previously used in the context of pHRI for learning a collaborative lifting task, wherein the haptic communication was exploited to extract leading and following robot behaviors [8]. Force control policies have been also learned through reinforcement learning for compliant manipulation [9].

In this paper we propose to teach a robot to pour drinks using force-based perceptions exclusively, where the key contribution is to exploit the haptic input information perceived at the beginning of each demonstration to parameterize force-based tasks and encode them through a compact probabilistic model (see Figure 1). In a real situation, a human carrying out the same task must turn the bottle in such a way that the fluid is poured, which definitely depends on the quantity of fluid inside it, as shown in Figure 2. Such situation of having a task parameter that modifies the execution accordingly is often found in several scenarios. For instance, a point-to-point reaching task may depend on the location of the object to be grasped, which would modify the trajectory followed by the robot to reach the goal. A similar case was tackled in [10] where the authors propose to learn reaching movements through HMMs projected onto several position-varying landmark frames (the task parameters), and then to estimate a resulting model in the form of products of Gaus-

sians. Similarly, Cederborg *et al.* used the notion of *framings* to incrementally learn various tasks that can be defined in different frames (also viewed as task parameters), where the system infers from demonstrations which particular frame should be used for each task [11].

In contrast to the aforementioned works, we propose a learning framework where a parametric version of the HMM (PHMM) [12] is used to represent the demonstrations and parameters of force-based skills, which is further used in a reproduction phase where GMR is implemented to retrieve the robot movements. We show the benefits of this approach in a *pouring* task scenario. Note that Krüger *et al.* [13] proposed to synthesize and recognize robot reaching movements – action primitives – by using PHMMs, where the task parameters correspond to the initial and final location of the grasped object. Nonetheless, this task may also be solved using the frames-based approaches described above, unlike our case where the parameters are not position dependent. The advantage of using a PHMM is that the parameters can come from different kinds of information source (i.e., not only from position data), which makes it more suitable to be used in force-based robot learning.

This paper is organized as follows: Section II describes our experimental setup and the pouring task taught to the robot. Sections III and IV explain the learning and reproduction phases, respectively, first illustrating the statistical parametric encoding of the demonstrations by using a PHMM and then, showing how GMR is implemented at the execution stage. Section V shows computational and robot execution results. Finally, the conclusions of this paper and future work are presented in Section VI.

II. EXPERIMENTAL SETUP

We constructed an experimental setup to teach a robotic manipulator to carry out the pouring task using exclusively haptic data. In such scenario a human user holding the end-effector of a 6-DoF haptic interface (Delta device from Force Dimension) teleoperates a robotic arm (RX60 from Stäubli) which has a force-torque sensor (Schunk FTC-050) placed on its wrist (see Figure 3). The sensed forces-torques are fed back to the teacher in order to establish a bidirectional communication channel during the demonstration stage. This implies to work at a minimum frequency of 1000 Hz to have a high fidelity force reflection and a stable teleoperation system, which greatly depends on the executed processes between the position sensing of the haptic device and when the sensed force is reflected on it. Our experimental setup takes such requirement into account and guarantees a high bandwidth communication in the haptic loop.

In our experimental scenario the robot holds a 1 liter plastic bottle full of tiny metallic spheres, which play the role of a fluid (this solution was adopted to avoid spilling liquid during tests and, in the end, what we like to learn is a given fluid-like dynamics, no matter which). The teacher teleoperates the robot in order to demonstrate how to pour 100 ml drinks into a plastic glass. In contrast to the illustrative example shown in Figure 2 where the human moves its arm in order to pour

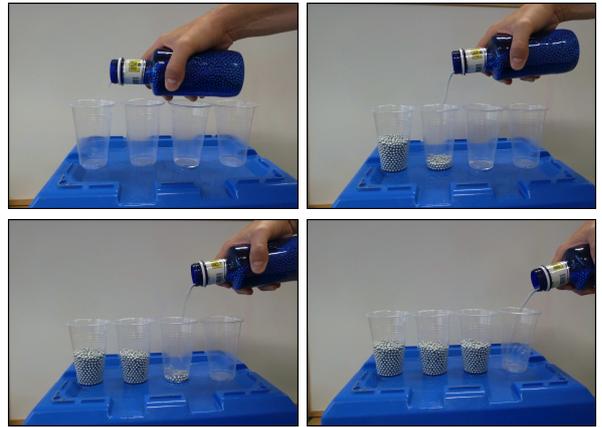


Fig. 2. Illustration of pouring task carried out by a human in a real situation. The less quantity of fluid inside the bottle, the more the human turns the bottle. Thus the initial conditions at each execution influence how much the bottle needs to be turned to accomplish the task successfully.

into all the glasses placed at different locations, here every sample of the task starts from a unique predefined initial pose of the bottle, which is also the stop configuration once the robot has poured a drink. Initially, the bottle is completely full, and the teacher carries out several demonstrations until the bottle is empty. Thus, the initial force-torque values for each example vary according to how much fluid has been poured previously, which, at the time, is the task parameter conditioning the robot movements.

Similarly to our previous work [14], a dynamic compensation model was used here for removing the bottle mass effects from the sensor readings, in order to feed back the teacher with only the external forces-torques generated by the fluid at the demonstration phase. Note that in this task, the teacher is able to watch the scene directly (see Figure 3), thus he/she can know the location of the glass in the robot workspace. Such information is not provided to the robot during the execution phase because the glass position is predefined in advance and fixed across the examples.²

Let us mention that the proposed task is challenging and has aroused the research community's interest recently. Tamosiunaite *et al.* [15] tackled the same problem using reinforcement learning, which was applied to improve the initial encoding obtained from human demonstrations modeled through dynamic motion primitives. Cakmak and Thomaz [16] taught a humanoid robot to pour through an active learning framework, wherein the robot was allowed to ask questions regarding the task at hand. Our approach significantly differs from these works in that the human-robot interaction is through a haptic device, the demonstrations are encoded by a probabilistic model that exploits the task parameters and the perception system senses only the forces-torques generated over the execution of the skill.

²Note that a camera system may also be used to know the location of the glass in the robot frame, so that the demonstrations would also be dependent on this parameter.



Fig. 3. Experimental scenario of the *pouring task*. The teacher demonstrates the robot how to pour 100 ml drinks into a glass by teleoperation.

III. PARAMETRIC ENCODING OF THE SKILL

The hidden Markov model has been extensively used for encoding or representing the teacher examples in LfD applications [2], [3], [8]. An HMM can be interpreted as an extension of the Gaussian mixture model (GMM) in which the choice of the mixture component for each observation depends also on the choice of the component for the previous observation. Moreover, this method has the advantage of containing time as an implicit variable, which can be exploited when dealing with force-based data because they tend to show very large time discrepancies.

Specifically, a classical HMM λ of N components is defined by its initial distribution $\pi = \{\pi_i\}$, the transition probability distribution $\mathbf{A} = \{a_{ij}\}$ and the continuous observation probability distribution $\mathbf{B} = \{b_i\}$, with $i, j = 1, 2, \dots, N$. For continuous observation densities, the b_i are often represented by a Gaussian distribution with mean μ_i and covariance matrix Σ_i . The model parameters are estimated through the *Baum-Welch* method given a training dataset $\{\mathbf{d}_p^m\}$, with $m = 1, 2, \dots, M$ and $p = 1, 2, \dots, P$, where M is the number of demonstrations and P is the number of datapoints collected along demonstration m (for a general explanation of the HMM please refer to [17]). It should be noted that this classic model does not handle task parameters explicitly, and if a parameter exists for every demonstration, a possible solution would be to add it as an additional (likely constant) input variable with the cost of increasing the dataspace dimensionality.

Instead, we propose to use a parametric version of the HMM, namely PHMM [12]. This technique models the dependence on the parameter of interest in an explicit way through the output densities b_i . Formally, the observation probability distributions are now a function of the demonstration and an associated parameter³ θ_m : $b_i(\mathbf{d}^m; \theta_m)$. In this paper we adopt the linear dependence of the mean of the Gaussian distributions on θ , wherein for each component

³The dimension of the parameter depends on its degrees of freedom, for instance θ would be a three dimensional vector if representing the location of an object in the space.

i of the model, we have:

$$\hat{\mu}_i(\theta_m) = \mathbf{W}_i \theta_m + \mu_i, \quad (1)$$

where \mathbf{W}_i describes the linear variation. Eq. (1) can be expressed in a matrix fashion as $\hat{\mu}_i(\theta_m) = \mathbf{Z}_i \Omega_m$, where

$$\mathbf{Z}_i \equiv [\mathbf{W}_i \quad \mu_i], \quad \Omega_m \equiv [\theta_m \quad 1]^T. \quad (2)$$

The former linear formulation allows to estimate only one additional model parameter, namely \mathbf{Z}_i , from which the means of the model are computed for a specific value of θ_m (the readers are referred to [12] in order to look up the estimation equation for \mathbf{Z}_i). Note that this parametric model allows us to provide a compact probabilistic encoding of the demonstrations, handling the task parameters through a linear dependence that modifies the location of the output densities in the dataspace, without influencing their shape (i.e., the covariance information). Results of the parametric encoding of the pouring skill are given and explained in Section V.

IV. REPRODUCTION OF THE SKILL

Once the PHMM has been trained, it is necessary to find a suitable way to reconstruct the output commands given a force perception and the resulting parametric model. To achieve this goal, we resort to use Gaussian mixture regression for computing the robot actions to be sent to the controller as the desired robot state to be achieved. Note that Calinon [19] proposed to use GMM/GMR for learning tasks at trajectory level, where the main idea is to model data from a mixture of Gaussians and to compute predictions for a given set of queries through a regression that uses mean and covariance information to retrieve the estimated output.

In this work, GMR uses the information encapsulated by the observation probability distributions of the PHMM. Formally, for each Gaussian component i of the model, input \mathbf{x} and output \mathbf{y} are separated by expressing the mean and covariance matrix as:

$$\hat{\mu}_i = \begin{bmatrix} \hat{\mu}_i^x \\ \hat{\mu}_i^y \end{bmatrix}, \quad \Sigma_i = \begin{bmatrix} \Sigma_i^{xx} & \Sigma_i^{xy} \\ \Sigma_i^{yx} & \Sigma_i^{yy} \end{bmatrix}. \quad (3)$$

Then, the conditional expectation $\hat{\mathbf{y}}$ given the input vector \mathbf{x} , for a mixture of N Gaussians is:

$$\hat{\mathbf{y}} = \sum_{i=1}^N h_i [\hat{\mu}_i^y + \Sigma_i^{yx} (\Sigma_i^{xx})^{-1} (\mathbf{x} - \hat{\mu}_i^x)], \quad (4)$$

where

$$h_i = \frac{p(i)p(\mathbf{x}|i)}{\sum_{j=1}^N p(j)p(\mathbf{x}|j)} \quad (5)$$

is a weight exclusively based on the input variables, mainly force-torque data in our task. Hence, we aim at predicting the necessary robot commands as a function of its force-based perceptions in order to follow the taught movements to carry out the pouring process. Computational and experimental results of the reproduction phase implementing GMR are shown and explained in the next section.

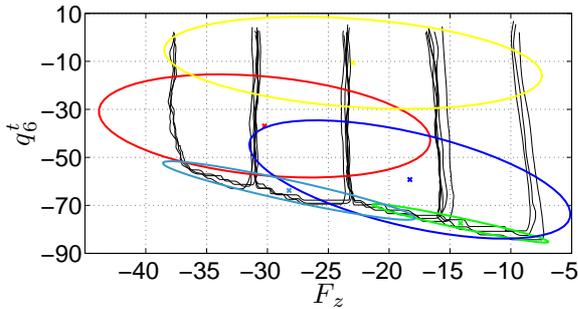


Fig. 4. Resulting 5-components HMM trained with three demonstrations of the pouring task. Each subset of provided samples of the skill shows different initial force-torque values given by the quantity of fluid inside the bottle. The less quantity, the more the robot rotates the bottle. Robot joint values are given in degrees while forces are in Newtons.

V. RESULTS

In the proposed experimental task, namely the pouring skill, the robot perceptions are the wrench $\Delta = \{F_x \ F_y \ F_z \ T_x \ T_y \ T_z\}$, i.e., the sensed forces and torques in the robot’s frame. However, several of these variables do not play a relevant role for learning the skill at all, thus we subjected the training data to a Mutual Information analysis in order to select the most relevant perceptions, reducing significantly the input dimensionality and the computational cost of further processes (i.e., encoding and reproduction phases), similarly to our previous work [2]. The resulting “most informative” set of inputs was $\delta = \{T_x \ F_z \ T_y\}$, of which T_x and F_z may also be directly extracted from an analysis of the task at hand.⁴

The whole input space is composed of the selected subset of variables δ and the current joint value q at time step t . Output is the desired robot joint position to be achieved at $t + 1$. Thus, each training datapoint is defined as $\mathbf{d}_p^m = \{T_x \ F_z \ T_y \ q_1^t \ \dots \ q_{N_q}^t \ q_1^{t+1} \ \dots \ q_{N_q}^{t+1}\}$, where N_q is the number of joints of the robot. The fact of having included the robot state into the input vector is aimed at encapsulating the task dynamics, so that the next robot state depends not only on the force-based perceptions but also on the current robot joint values, which is in the line of a consistent representation of the skill [18].

In order to teach the robot to pour drinks, three “complete executions” ($M = 3$) of the task (i.e., serving four drinks consecutively) are provided to the robot by teleoperation as described in Section II. Each demonstration m consists in starting with the bottle full of fluid and pouring four 100 ml drinks. Note that after each drink is poured, the initial force-torque value changes for the next one, which conditions the robot movements as shown in Figure 4 where the black lines represent the teacher’s demonstrations. Observe that the less quantity of fluid, the more the robot rotates the bottle. Using this training dataset, two different models were trained for comparison purposes, one of them corresponds to the

⁴A simple analysis should convince the reader that the bottle with the fluid inside produces a load on the sensor because the gravitational force, generating a force along the axis z and a torque about x .

classic HMM and the other one to the PHMM. Encoding and reproduction results are presented next, where the projection of the models is done on the space defined by F_z and q_6 , which are one of the most relevant variables of the task and the robot joint turning the bottle, respectively.

A. Classical HMM

The resulting training dataset was used to train a 5-components HMM ($N = 5$) by applying the *Baum-Welch* method until convergence. The number of Gaussians was chosen according to the Bayesian Information Criterion [19]. Figure 4 shows the model encoding the pouring skill, where the yellow component covers the beginning and the end of all the executions, whereas the light blue and green ellipses are encapsulating the phases when the fluid is coming out of the bottle. The other two Gaussians can be considered as intermediate phases of the task.

In order to test the reproduction performance of the model, one demonstration (serving four 100 ml drinks) was removed from the training data to be used as query datapoints. All the joint trajectories executed by the robot were quite similar to the ones obtained from the teacher examples as well as the input-output pattern. After this, the following tests were aimed at evaluating the generalization capabilities of the trained HMM. In this case, the bottle contained quantities of fluid different from the ones used at the demonstration phase. For all the tests where the starting force-torque perception was covered by the initial HMM component (i.e., the yellow ellipse in Figure 4), the robot performed successfully. The robot joint trajectories and the input-output pattern for one of these tests is shown in Figure 5(a). Nevertheless, as the starting perception significantly differs from the values encapsulated by the initial component, the robot performance deteriorates considerably (see Figure 5(b)). In other words, the actual model shows good interpolation competences but a poor extrapolation performance, which constrains the range of situations where the robot would perform successfully without retraining the HMM.

B. PHMM

A PHMM of three Gaussians ($N = 3$) was trained using only one demonstration ($M = 1$) of the pouring skill, i.e., four examples of pouring 100 ml drinks. The parameter was set to be the initial force-torque values directly representing the quantity of fluid at the beginning of the demonstration m , that is $\theta_m = [F_z^{m,t} \ T_x^{m,t}]^T$ with $t = 1$. Therefore, $\Omega_m = [F_z^{m,t} \ T_x^{m,t} \ 1]^T$ and $\dim(\mathbf{Z}_i) = (N_i + N_o) \times (\dim(\theta_m) + 1)$, where $N_i = 9$ and $N_o = 6$ are the number of inputs and outputs of the task, respectively. Figure 6 shows the resulting model, where it is possible to see how the PHMM is able to encode the task through a simple *left-to-right* topology. Note how the provided parameter translates the model components to cover the corresponding demonstration data due to the linear relationship between the task parameter and the Gaussian means (see Eq. (1)). It should be noted that neither the Gaussians shape nor their orientation are modified. In this task – and those also depending on parameters – it may be

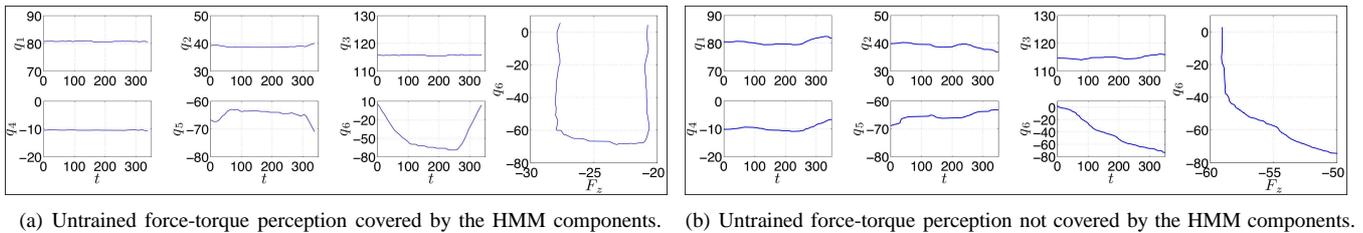


Fig. 5. Reproduction for two quantities of fluid using the HMM. The robot joint trajectories during the execution are shown in the three first columns at each graph. Last plot displays the reproduction pattern for q_6 as a function of F_z . Robot joint values are given in degrees while forces are in Newtons.

useful to shape the Gaussians to improve the reproductions. For instance, the first component of the model may be explicitly shaped according to how much the robot rotates the bottle for a given force parameter. Approaches like the proposed by Herzog *et al.* [20] or Calinon *et al.* [21] may be taken into consideration in future for improving the current results. The core of these works is to also parameterize covariance of the Gaussian components, which is crucial because covariance encodes the local relationships among the variables that are of interest for the task, as well as the expected variations during its execution.

The reproduction performance of the PHMM was satisfactory when faced to force-torque values previously observed during the demonstrations, showing joint trajectories very similar to the ones provided by the HMM. Regarding the generalization capabilities, different quantities of fluid from the ones used in the demonstrations were given to the robot, using the same bottle and a larger one. For both cases, the robot successfully poured a drink of 100 ml approximately (see Figure 7). Figure 7(b) displays a robot execution when a larger bottle full of fluid was used, where it is possible to observe how the robot joint q_6 follows a trajectory quite similar to the demonstrated ones, performing the skill as expected. This result shows that the robot is able to pour drinks using different types of bottle without retraining the model, because the force-based parameter allows to displace the model so that it can cover the subspace where the data are expected to be, which is not possible using the classic HMM (see Figure 5(b)). Thus, the PHMM provides better generalization capabilities than those observed using the HMM, with fewer components and a simpler topology.

Figure 8 displays the Gaussian distribution and the influence of the model components for the reproduction shown in Figure 7(b). One can observe how the model translates to cover a different data subspace given the new task parameter. It is also possible to see how the GMR weights h_i (Eq. (4)) evolve over time, showing how the model components influence the reproduction. A video accompanying this paper is available at <http://programming-by-demonstration.org/RoMoCo13>, where executions of the skill for observed and unobserved situations are shown.

VI. CONCLUSIONS AND FUTURE WORK

This work is aimed at exploiting force parameters to obtain a compact encoding of skills that are exclusively

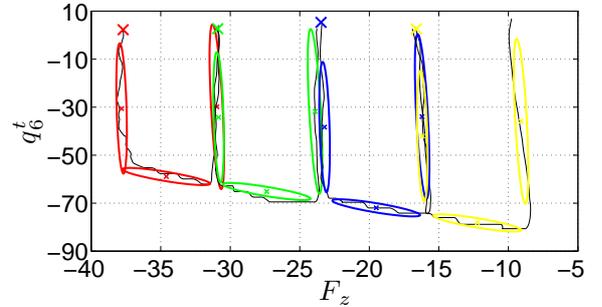
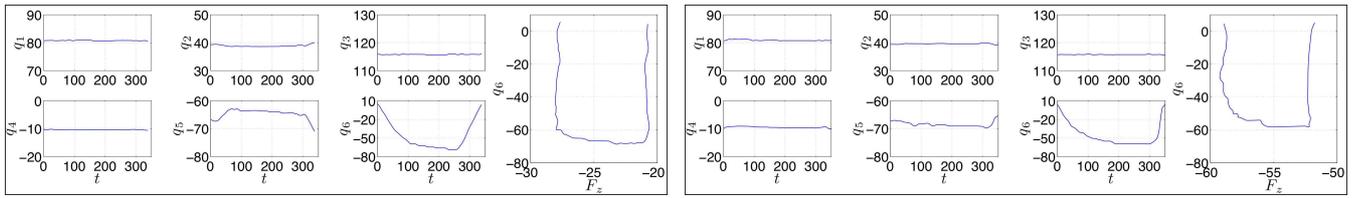


Fig. 6. Resulting 3-components PHMM trained with one demonstration of the task (four 100 ml drinks poured). Crosses show the skill parameter, for which the corresponding model is displayed using the same color. The PHMM satisfactorily encodes the task using a simple *left-to-right* topology. Robot joint values are given in degrees while forces are in Newtons.

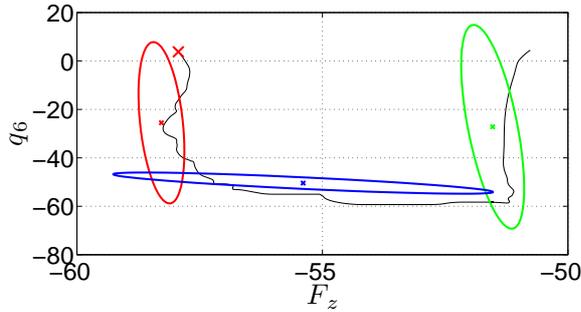
based on haptic perceptions. We propose to use a learning framework based on a parametric version of the classic HMM and Gaussian mixture regression to encapsulate the demonstrations and reproduce the task, respectively. One of the contributions of this paper is to show how the PHMM can offer a compact and simpler model for representing force-based skills by taking advantage of the task parameters, in comparison to other learning algorithms commonly used as the HMM. Results demonstrate that the PHMM provide better generalization capabilities than those observed when the HMM was used.

It should be noted that fewer demonstrations were needed to train the PHMM than when training the classic version. This is because the HMM components must cover the entire space that the provided samples occupy, thus needing more information to converge to a good encoding in terms of the resulting likelihood. In contrast, the PHMM exploits the task parameters to linearly translate its components so that they cover a specific region of the dataspace, requiring less information – and probably fewer components – to achieve a good representation of the task. On the other hand, it is worth mentioning that in literature task parameters often considered are the location – and possibly the orientation – of objects to be grasped. Thus, other scenarios are ignored, where the parameters come from different sources, like force-based information as we propose. Therefore, in the line of our previous work on force-based LfD [2], we consider this work as a step ahead for achieving a more generic and versatile framework to learn tasks in the force domain.

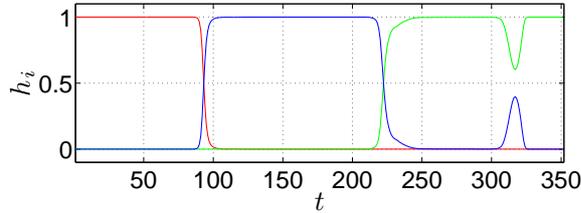


(a) Untrained force-torque perception lying in the space covered by the demonstrations. (b) Untrained force-torque perception lying out of the space covered by the demonstrations.

Fig. 7. Reproduction for two quantities of fluid using the PHMM. The robot joint trajectories during the execution are shown in the three first columns at each graph. Last plot displays the reproduction pattern for q_6 as a function of F_z . Robot joint values are given in degrees while forces are in Newtons.



(a) Resulting PHMM for the given unobserved force perception



(b) PHMM component activation over the course of the skill

Fig. 8. *Top*: Component distribution and input-output profile of a reproduction for unobserved force-torque data, where the cross represents the given parameter. *Bottom*: Component influence profiles during the reproduction.

As future work, we plan to include vision to the robot perception system in order to exploit multi-modal sensing in robot learning from demonstration, where both vision and force parameters may condition the robot behavior. For instance, in the proposed experimental setting a camera may provide the location of the glass so that the robot can approach the bottle to the glass and then pour the drink. For both phases of the task there will be a different parameter conditioning the robot actions, which may be exploited to encode the whole task compactly. Moreover, we will study techniques to allow the task parameter to shape and reorient the model Gaussians, which would provide an even more generic framework and better reproduction results.

REFERENCES

- [1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, *Springer Handbook of Robotics*. Springer, 2008, ch. 59. Robot Programming by Demonstration, pp. 1371–1394.
- [2] L. Rozo, P. Jiménez, and C. Torras, “Robot learning from demonstration of force-based tasks with multiple solution trajectories,” in *Intl. Conf. on Advanced Robotics*, 2011, pp. 124–129.
- [3] D. Lee and C. Ott, “Incremental kinesthetic teaching of motion primitives using the motion refinement tube,” *Autonomous Robots*, vol. 31, pp. 115–131, 2011.
- [4] M. Riley, A. Ude, C. Atkeson, and G. Cheng, “Coaching: An approach to efficiently and intuitively create humanoid robot behaviors,” in *Intl. Conf. on Humanoid Robots*, 2006, pp. 567–574.
- [5] P. Falco, R. Jäkel, C. Natale, and R. Dillmann, “Improvement of human hand motion observation by exploiting contact force measurements,” in *Intl. Conf. on Humanoid Robots*, 2011, pp. 141–146.
- [6] V. Koropouli, D. Lee, and S. Hirche, “Learning interaction control policies by demonstration,” in *Intl. Conf. on Intelligent Robots and Systems*, 2011, pp. 344–349.
- [7] A. Schmidts, D. Lee, and A. Peer, “Imitation learning of human grasping skills from motion and force data,” in *Intl. Conf. on Intelligent Robots and Systems*, 2011, pp. 1002–1007.
- [8] S. Calinon, P. Evrard, E. Gribovskaia, A. Billard, and A. Kheddar, “Learning collaborative manipulation tasks by demonstration using a haptic interface,” in *Intl. Conf. on Advanced Robotics*, 2011, pp. 1–6.
- [9] M. Kalakrishnan, L. Righetti, P. Pastor, and S. Schaal, “Learning force control policies for compliant manipulation,” in *Intl. Conf. on Intelligent Robots and Systems*, 2011, pp. 4639–4644.
- [10] S. Calinon, F. D’halluin, E. Sauser, D. Caldwell, and A. Billard, “Learning and reproduction of gestures by imitation: An approach based on hidden Markov model and Gaussian mixture regression,” *Robotics and Automation Magazine*, vol. 17, no. 2, pp. 44–54, 2010.
- [11] T. Cederborg, L. Ming, A. Baranes, and P.-Y. Oudeyer, “Incremental local online gaussian mixture regression for imitation learning of multiple tasks,” in *Intl. Conf. on Intelligent Robots and Systems*, 2010, pp. 267–274.
- [12] A. Wilson and A. Bobick, “Parametric hidden Markov models for gesture recognition,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 884–900, 1999.
- [13] V. Krüger, D. Herzog, S. Baby, A. Ude, and D. Kragic, “Learning actions from observations: Primitive-based modeling and grammar,” *Robotics and Automation Magazine*, vol. 17, no. 2, pp. 30–43, 2010.
- [14] L. Rozo, P. Jiménez, and C. Torras, “Sharpening haptic inputs for teaching a manipulation skill to a robot,” in *Intl. Conf. on Applied Bionics and Biomechanics*, 2010, pp. 370–377.
- [15] M. Tamosiunaite, B. Nemeč, A. Ude, and F. Wörgötter, “Learning to pour with a robot arm combining goal and shape learning for dynamic movement primitives,” *Robotics and Autonomous Systems*, vol. 59, no. 11, pp. 910–922, 2011.
- [16] M. Cakmak and A. Thomaz, “Designing robot learners that ask good questions,” in *Intl. Conf. on Human-Robot Interaction*, 2012, pp. 17–24.
- [17] L. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” in *Proceedings of the IEEE*, 1989, pp. 257–286.
- [18] D. Pardo, L. Rozo, G. Alenyà, and C. Torras, “Dynamically consistent probabilistic model for robot motion learning,” in *Intl. Conf. on Intelligent Robots and Systems - Workshop on Learning and Interaction in Haptic Robots*, 2012.
- [19] S. Calinon, *Robot Programming by Demonstration: A Probabilistic Approach*. EPFL/CRC Press, 2009.
- [20] D. Herzog, V. Krüger, and D. Grest, “Parametric hidden Markov models for recognition and synthesis of movements,” in *British Machine Vision Conf.*, 2008, pp. 163–172.
- [21] S. Calinon, Z. Li, T. Alizadeh, N. Tsagarakis, and D. Caldwell, “Statistical dynamical systems for skills acquisition in humanoids,” in *Intl. Conf. on Humanoid Robots*, 2012, pp. 323–329.