Associative Skill Memory Models

Hakan Girgin and Emre Ugur

***Abstract*— Associative Skill Memories (ASMs) were formulated to encode stereotypical movements along with their stereotypical sensory events to increase the robustness of underlying dynamic movement primitives (DMPs) against noisy perception and perturbations. In ASMs, the stored sensory trajectories, such as the haptic and tactile measurements, are used to compute how much a perturbed movement deviates from the desired one, and to correct the movement if possible. In our work, we extend ASMs: rather than using stored single sensory trajectory instances, our system generates sensory event models and exploits those models to correct the perturbed movements during executions with the aim of generalizing to novel configurations. In particular, measured force and the torque trajectories are modelled using Parametric Hidden Markov Models, and then reproduced by Gaussian Mixture Regression. With Baxter robot, we demonstrate that our proposed force feedback model can be used to correct a trajectory while pushing an object with a mass never experienced before, and which otherwise slips away from the gripper because of noise. In the end, we discuss how far this skill can be generalized using the force model and possible future improvements.**

I. INTRODUCTION

Learning from Demonstration (LfD) [1] has been suggested as an efficient and intuitive way to teach new skills to the robots, where the robot observes, learns and imitates the actions demonstrated by the human tutors. LfD has been applied to various robotic learning problems including object grasping and manipulation [2]–[6]. Among others, learning methods that are based on dynamic systems [7] and statistical modeling have been popular in the recent years.

Dynamic Movement Primitives (DMPs) [7], for example, encode the demonstrated trajectory as a set of differential equations, and offers advantages such as one-shot learning of non-linear movements, real-time stability and robustness under perturbations with guarantees in reaching the goal state, generalization of the movement for different goals, and linear combination of parameters. The parameters of the system can be learned with different advanced algorithms such as Locally Weighted Regression [8] and Locally Weighted Projection Regression [9]. Statistical modeling, which can model the statistical regularities and important features of the demonstrated motions, has also been influential in learning the skills [2], [10].

After encoding the action, the robot is generally required to refine the parameters of the learned control policy [11].

This research has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 731761, IMAGINE; and was partially supported by TUBITAK BIDEB under agreement No 117C016 and by Bogazici Resarch Fund (BAP) startup project no 16A01SUP2.

Bogazici University, Department of Computer Engineering, Istanbul, Turkey `firstname.lastname@boun.edu.tr`

Fig. 1: Experimental setup showing Baxter robot pushing 7.5 kg battery pack.

Memorized force and tactile profiles can also be used to modulate learned Dynamic Movement Primitives (DMPs) [12], [13]. Memorized force and tactile profiles have already been successfully utilized in modulating learned movement primitives in difficult manipulation tasks that contain high degrees of noise in perception such as flipping a box using chopsticks However, we believe that rather than memorizing one single haptic profile for a skill, learning general multi-model sensory models might provide us with more generalizable and robust manipulation skills.

Chu et al. learned such multi-modal models based on Hidden Markov Models from temperature, pressure and fingertip information for exploratory object classification tasks [14], however the learned models were not used to adapt any further action execution. Latent Drichlet Allocation [15] and recently deep networks [16] were used to learn multi-modal models from different sensory information such as temperature, pressure, fingertip, contacts, proprioception, and speech; however these models were used only to categorize the sensory data without any effect on action execution. More recently, Kramberger et al. investigated the same problem of generalization of force/torque profiles for contact tasks [17]. In their work, these profiles are modeled by Locally Weighted Regression (LWR) which has local generalization capabilities, hence successful at intermediate query points.

However, the learned models did not extrapolate from the training queries.

In this paper, our system generates sensory event models, and exploits those models to correct the perturbed movements during executions with the aim of generalizing to the novel configurations. In particular, measured force and the torque trajectories are modeled using Parametric Hidden Markov Models (PHMM), and then reproduced by Gaussian Mixture Regression (GMR). PHMM was previously used in robotics application for a liquid pouring task to create a model that links the joint space and sensory feedback information of the robot to the amount of the liquid [18]. However, their work consisted of learning the position and force trajectory together by PHMM and therefore was not robust to dynamical changes as is DMP. With Baxter robot (Fig. 1), we demonstrate that our proposed force feedback model can be used to correct a trajectory while pushing an object of mass never experienced before, which otherwise slips away from the gripper because of noise in perception. The rest of the paper is structured as follows: Section II. provides the proposed method that use PHMM to model the force feedback term of DMPs, Section III. gives experimental results of pushing of varying weighted objects task with some analysis on the adaptation limitations of the chosen feedback model and Section IV and provide discussion and conclusion.

II. METHODS

The formulation of DMP allows the robot to learn a stereotypical skill from demonstration. Adding a sensory feedback to the system enhances the capabilities of the robot in the learned skill by sending corrective signals to low-level controllers [4]. In an open-loop execution of the skill pushing a cup, for example, the dynamics of the environment can not be always known to the robot and also due to the noise in perception and uncertainties in the environment, and the cup can slide from its end-effector from time to time. The forces that the robot should feel during the execution, namely the desired forces, help the robot the orient and position its end-effector so that it prevents sliding of the cup.

However, storing and using force trajectory instance does not allow generalization to new situations in the long run. Pastor et al. called this storage of data coupling with movements as Associative Skill Memories [12]. In this paper, instead of memorizing how to feel during each execution, we propose to model the forces experienced by the robot during its successful action executions for each primitive movement by using PHMMs and reproduce them using GMR. For a typical movement, we argue that the relation between parameters of the environment and experienced force feedback can be learned via linking at least two obtained force models by means of Gaussian centers of their hidden states, and that the desired forces for a new movement at the proximity of these demonstrations can be predicted from this parametric model.

A. Dynamic Movement Primitives

For one degree of freedom, DMP is composed of the following set of differential equations:

$$\begin{aligned} \tau\dot{v} &= K(g-x) - Dv - K(g-x_0)s + Kf(s) \\ \tau\dot{x} &= v \end{aligned} \tag{1}$$

where,

- x and $\dot{x}$ are the position and the velocity,
- x_0 and g are the initial and goal positions,
- v and $\dot{v}$ are the velocity and acceleration scaled by the duration of the demonstration τ,
- K and D are the proportionality and the damping constants,
- $f(s)$ is a nonlinear function of the phase variable s

K and D are selected so that there is critical damping and the damping constant is taken as $D = 2\sqrt{K}$. The phase variable makes DMP temporal invariant by encoding time in its canonical system defined as

$$\tau\dot{s} = -\alpha s \tag{2}$$

where α is a constant representing the convergence rate of the phase variable from 1 to 0. Starting each DMP with the same phase variable and integrating with the same canonical system ensures their simultaneous evaluation.

The shape of the trajectory $f(s)$, is encoded as normalized weighted sum of radial basis functions, (ψ_i) as in equation:

$$f(s) = \frac{\sum_i w_i \psi_i(s) s}{\sum_i \psi_i(s)} \tag{3}$$

where $\psi_i = exp(-h_i(s-c_i)^2)$ are the radial basis functions and c_i and h_i are respectively the mean and the variance of these functions. The weights w_i specific to each movement primitive are learned by linear regression [19].

B. Sensory Feedback Extension to DMPs

In ASMs, a coupling term is integrated into the original DMP formulation Eq. 1 to compensate for the generalized forces that the robot senses during the execution of a task, since each movement primitive should capture the entire dynamics of the skill. This coupling term is given by

$$\zeta = \mathbf{K_1}\mathbf{J}^{\mathbf{T}}_{\mathbf{sensor}}\mathbf{K_2}(\mathbf{F} - \mathbf{F_{des}}) \tag{4}$$

where $\mathbf{K_1}$ and $\mathbf{K_2}$ are positive definite gain matrices, $\mathbf{J}^{\mathbf{T}}_{\mathbf{sensor}}$ is the transpose of the Jacobian with respect to sensors by which the forces are measured. $\mathbf{F}$ and $\mathbf{F_{des}}$ are the current and desired generalized forces which, in task-space, is the end-effector's 6D wrench.

Coupling term incorporated in DMP formulation Eq. 1 is then given by

$$\tau\dot{v} = K(g-x) - Dv - K(g-x_0)s + Kf(s) + \zeta \tag{5}$$

C. Encoding Force Feedback by PHMMs

In this paper, we propose to construct temporal probabilistic models to encode force feedback trajectories measured from the same movement primitive that is executed several times. For this we propose to use PHMM.

Hidden Markov Model (HMM), λ, of N hidden states, is composed of the prior distributions π_i, the transition probabilities a_{ij} and the observation probability distributions b_i, represented as: $\lambda = \{\pi_i, a_{ij}, b_i\}_{i,j=1}^{N}$

When the observation data is continuous, we assume generally that each state produces a multivariate Gaussian distribution with mean $\boldsymbol{\mu_i}$ and covariance matrix $\boldsymbol{\Sigma_i}$. These parameters are learned by an extension of Expectation Maximization (EM) Algorithm, called the Baum-Welch Algorithm. For further explanations, reader is invited to refer to the paper of Rabiner [20]. However, HMMs are ignoring the environmental parameter information of the demonstrations. Therefore, in this paper, we propose to use the parametric version of HMM, Parametric Hidden Markov Models (PHMM). In these models, the observational probability distributions are functions of the parameters $\boldsymbol{\theta} = \{\theta_k\}_{k=1}^{K}$ of the demonstrations k through the means of the Gaussian distribution functions where K represents the number of the demonstrations. Each of the Gaussian means produced by all hidden states are linear functions of θ_k and can be expressed as follows:

$$\hat{\boldsymbol{\mu}}_i(\boldsymbol{\theta}_k) = \boldsymbol{W}_i\boldsymbol{\theta}_k + \boldsymbol{\mu}_i \tag{6}$$

This way the Gaussians corresponding to a state in each demonstration are tied linearly by their means. For further information, refer to [21]. In PHMMs, to determine the observational probability distributions, in addition to $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma_i}$, we also need to find $\boldsymbol{W}_i$. However, in Baum-Welch Algorithm, the drawback of updating the values of $\boldsymbol{W}_i$ and $\boldsymbol{\mu}_i$ separately is that while updating one, the non-updated value of the other one is used. These values can be written as in Eq. 7, so that we only need to update $\boldsymbol{Z}_i$ and $\boldsymbol{\Sigma_i}$ for observational probability distributions. For further information, refer to [21].

$$\boldsymbol{Z}_i \equiv [\boldsymbol{W}_i \;\; \boldsymbol{\mu}_i], \;\; \boldsymbol{\Omega}_k \equiv \begin{bmatrix} \boldsymbol{\theta}_k \\ 1 \end{bmatrix} \tag{7}$$

D. Trajectory Generation

The weights and values of the learned radial basis functions and phase variable can be put inside the Eq. 3 to find the value of the shaping function. With these values, Eq. 1 and 2 can be used to calculate the position and the velocity.

E. Desired Forces Generation

When a new parameter is given to the model retrieved by PHMM, each hidden state produces new multivariate Gaussian distributions. The mean vector of these distributions $\boldsymbol{\mu}_i$ and the covariance matrix $\boldsymbol{\Sigma_i}$, can be expressed as partitioned matrices by splitting the input $\boldsymbol{x}$ and output $\boldsymbol{y}$ as in:

$$\boldsymbol{\mu_i} = \begin{bmatrix} \boldsymbol{\mu_i^x} \\ \boldsymbol{\mu_i^y} \end{bmatrix} \quad \boldsymbol{\Sigma_i} = \begin{bmatrix} \boldsymbol{\Sigma_i^{xx}} & \boldsymbol{\Sigma_i^{xy}} \\ \boldsymbol{\Sigma_i^{yx}} & \boldsymbol{\Sigma_i^{yy}} \end{bmatrix} \tag{8}$$

According to the GMR, the output vector can be found by inserting the input vector and the Gaussian distribution acquired from the PHMM into the following equation [22].

$$\boldsymbol{y} = \sum_{i=1}^{N} h_i[\boldsymbol{\mu_i^y} + \boldsymbol{\Sigma_i^{yx}}(\boldsymbol{\Sigma_i^{xx}})^{-1}(\boldsymbol{x} - \boldsymbol{\mu_i^x})] \tag{9}$$

Here h_i are the weights of the marginal distribution of the input and are calculated according to the Eq. 10 where $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu_i^x}, \boldsymbol{\Sigma_i^{xx}})$ is the multivariate Gaussian density function of the input.

$$h_i = \frac{\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu_i^x}, \boldsymbol{\Sigma_i^{xx}})}{\sum_{i=1}^{N} \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu_i^x}, \boldsymbol{\Sigma_i^{xx}})} \tag{10}$$

III. EXPERIMENTS

A. Experimental Setup

Our experimental setup is composed of a Baxter robot which has two 7-DoF anthropomorphic arm, each actuated by a series elastic actuators enabling to measure torque output directly from the actuators (see Figure 1). The arm has a electric, parallel jaw gripper that is used in closed state with 4 cm wide open during the experiments. The experiments are conducted on a flat table, using 1.75 kg small batteries and a 2.25 kg big battery with enough surface area to avoid gripper from slipping away in the execution phase. We successively attached the small batteries to the big one with a tape and let the robot interact with the surface of the large battery in order to increase the interaction surface area.

B. Task: Push

The task of the robot is to push the box from an initial position to a final position with approximately linear in the x-direction at the robot's frame, i.e. only the x-position of the end-effector is changing with time. We selected the task of "pushing an object to a goal position" task in the experiments as this task requires exploitation of the learned force feedback model when the object is not moving as expected during the execution in response to the learned and reproduced movement of the end-effector. Such unexpected behavior can be observed through introducing different types of noise and perturbations: by incorrect perception of the intial location of the object and initiating the push trajectory from a slightly shifted position; or by physically perturbing the object while being pushed. In this paper, we simulated noise in perception, initiated the push trajectory from a slightly different position (around $6.25cm$ maximum), and called this setup as 'misplaced object'.

The movement is demonstrated by kinesthetic teaching, using the gravity compensating mode of the Baxter arm. Because holding the end effector while kinesthetic teaching affects the force/torque measurements, the recorded trajectory is re-executed without human intervention, and modelled with a set of DMPs in task space. Note that even though DMP in quaternion space has a different set of equations [4] because of the unity constraints on quaternions, we still modeled the trajectory with the standard DMPs.

TABLE I: Experimental conditions.

Condition	Explanation
Misplaced-No-Force	object is misplaced, control with no force feedback coupling term
Misplaced-Memory-Force	object is misplaced, control with closest memorized force feedback coupling term
Misplaced-PHMM-Force	object is misplaced, control with PHMM model feedback coupling term

C. Force feedback model

After the trajectory is encoded as DMP as described above, the DMP model without force feedback is executed with 4 different object masses (2.25kg, 4kg, 5.75kg, 7.5 kg), three times each, and the wrench data provided by the Baxter robot are saved. The last object of mass 7.5 kg is used as test data. The wrench data obtained from the original end-effector trajectory were modeled by PHMM of 10 hidden states with the parameters $\theta_k = m_k$ (see Eq. 6) , where m_k is the mass of each object to push. Wrench data corresponding to these executions with the PHMM model fitted, and the reference forces for the novel environment is shown by Fig. 2 in only x-direction since the position trajectory is not changing much in directions other than x. This produces significantly different forces in x-directions but slightly different ones in other directions. With the new means of the Gaussians of hidden states when the parameter of the novel environment is given to the PHMM model, we can predict the desired forces using GMR. The predicted desired force trajectory is shown by solid black line in Fig. 2. The reference and the predicted force trajectories are both around 15 N and demonstrates our model's generalization capability.

D. Robot Execution

Since the task is to push an object on a table, we decided to neglect the effect of the forces orthogonal to the table, i.e. F_z, and the corresponding torques, T_x and T_y in the computations. We also did not consider the effect of T_z to focus only on the 2D forces and left the torque feedback analysis for future work with DMP quaternions. Therefore, in the Eq. 4, we set the the first two diagonal elements of $\mathbf{K_1}$ corresponding to x and y directions equal to 80 and other elements to zero. $\mathbf{K_2}$ and Jacobian are set to identity matrices since we worked on task space. In experiments we used three different conditions for evaluating our method: Misplaced-No-Force, Misplaced-Memory-Force and Misplaced-PHMM-Force (Table I). Misplaced-No-Force condition is the base condition, whereas Misplaced-Memory-Force condition uses a memorized force feedback trajectory that is obtained from the most similar environment conditions, i.e. object masses. Misplaced-PHMM-Force is the condition where our proposed model-based method is tested.

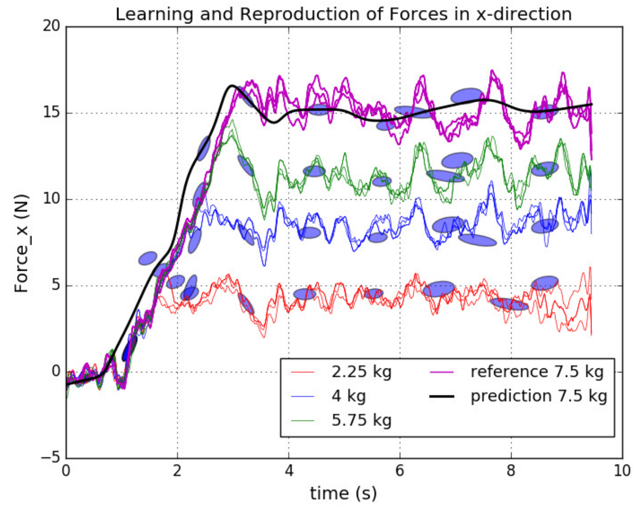


Fig. 2: PHMM model with GMR prediction and executions with different mass objects. Blue, red, green and magenta lines correspond respectively to the executions with 2.25 kg, 4 kg, 5.75 kg and 7.5 kg battery packs. Ellipses are the PHMM model learned from the red, blue and green executions. And the black line is the prediction obtained from this model with the novel environment parameter 7.5 kg.

E. Results

We made a systematic evaluation that compares the performance of push actions with PHMM model based force feedback terms against push actions that use memorized force feedback trajectories, i.e. no model. The object is placed to 5 different positions for this purpose. For each initial object position, three push actions are executed with and without PHMM models. After each push action, the distance of the final object position to the goal position is measured. Fig. 3 shows the results. Both with and without PHMM, higher error was observed when the object is placed further away from the position where push action was demonstrated. More importantly, less error was observed with PHMM-based push executions in all configurations compared to the executions that do not use any model.

Fig. 4 provides final positions of the object that were initially placed 3.75cm away from the demonstrated position. In Fig. 3, we did not provide the result of push action that do not exploit force feedback term, i.e. misplaced-no-force, as those push actions failed in all cases as expected and visible from even one snapshot, Fig. 4(a).

We have chosen randomly a specific misplacing distance of 3.75cm to show the changing forces felt by the robot during the execution and based on different experimental conditions explained in Table I. The experiment snapshots and corresponding force and positions plots (only in x and y directions) are shown respectively by Fig. 4 and Fig. 5 and an accompanying video is attached to the paper. In Fig. 5, since each DoF of force feedback affects its position, x and y plots should be interpreted separately. Seeing the

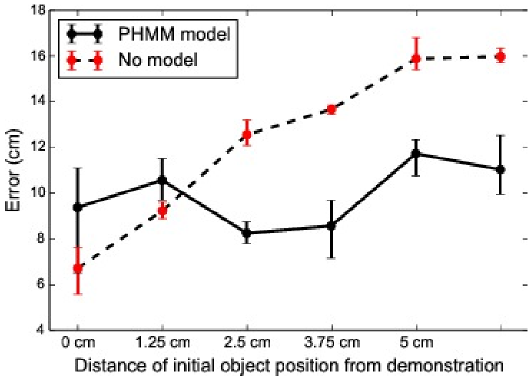


Fig. 3: The errors, i.e. distances to the goal position, observed at the end of push actions when object is placed to different initial positions.The bars show maximum, mean and minimum errors obtained from 3 executions in the same configurations.

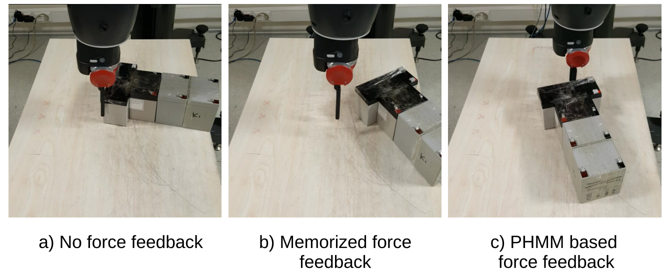


Fig. 4: Final positions of the objects at the end of push actions. Object is placed 3.75cm away from the position that demonstrations were collected and its mass is different from the objects in demonstrations. As shown, while push action that did not use force feedback term completely failed, action that exploited PHMM-based force feedback model generalized best to the different configuration. The difference between (b) and (c) is not that high, but still significant, as observed from the object orientations in Fig. 3. The attached video shows the push action of the robot and its interaction with the object during action execution.

execution video and verifying by looking at the plots for the *Misplaced-PHMM-Force* condition,although the desired values are not reached, as shown, the predicted force feedbacks are followed better compared to other and resulting in more successful push actions. The success lies in the fact that while x position follows slowly while y position searches for the right forces.

IV. DISCUSSION

Since our robot was barely pushing 7.5kg battery pack, it was not possible to try to push heavier objects, neither was interesting to push lighter ones since the force/torque feedback received was around zero, to analyze the limitations of the generalization capability of the PHMM model. However we argue that such linear model is sufficiently generalizable in simple tasks like pushing or pouring [18] with robots that have more payloads. Therefore, we leave the analysis of the restrictions of the model as future work.

By assuming diagonal gain matrices $\mathbf{K}$ for the sensory feedback coupling term, we underestimate the possible effects of some degrees of freedom on the others. For example, we assume that only x-position is affected when there is an error between the desired and actual generalized forces in x direction. However, we should learn a $\mathbf{K}$ for specific tasks rather than setting a constant value, by investigating the synergies between the degrees of freedom.

Since we worked only on the x and y directions of the robot end-effector frame, without any rotation, this did not cause any problems. However, using quaternion DMPs to analyze the generalization and adaptation capabilities of our model is set as future work.

Instead of force feedback, visual feedback can be used by keeping track of the object position. However, this is challenging because of the high dimensional visual space and difficulty in detecting the relevant features.

V. CONCLUSION

In this paper, we learned and exploited sensory event models to correct ongoing movements that are affected from noisy perception and to generalize to novel environments. Our system successfully exploited the learned force feedback models in order to adapt to noisy situations in a object pushing task with non-linear trajectory. We also showed that the desired force/torque profile for the pushing task in a novel situation can be predicted using PHMM models.

REFERENCES

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, “A survey of robot learning from demonstration,” *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[2] S. Calinon, P. Evrard, E. Gribovskaya, A. Billard, and A. Kheddar, “Learning collaborative manipulation tasks by demonstration using a haptic interface,” in *Advanced Robotics, 2009. ICAR 2009. International Conference on*. IEEE, 2009, pp. 1–6.

[3] T. Asfour, P. Azad, F. Gyarfas, and R. Dillmann, “Imitation learning of dual-arm manipulation tasks in humanoid robots,” *International Journal of Humanoid Robotics*, vol. 5, no. 02, pp. 183–202, 2008.

[4] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal, “Online movement adaptation based on previous sensory experiences,” in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 365–371.

[5] H. Ben Amor, O. Kroemer, U. Hillenbrand, G. Neumann, and J. Peters, “Generalization of human grasping for multi-fingered robot hands,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 2043–2050.

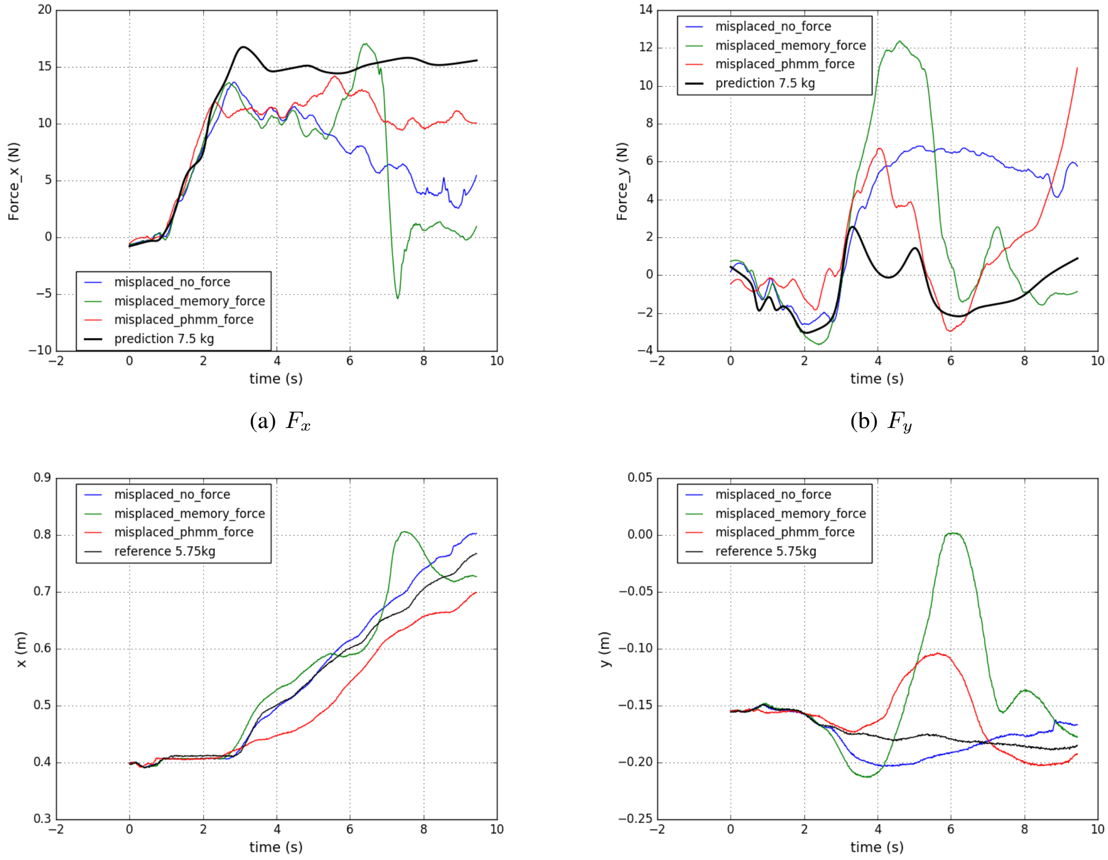
[6] M. Mühlig, M. Gienger, and J. J. Steil, “Interactive imitation learning of object movement skills,” *Autonomous Robots*, vol. 32, no. 2, pp. 97–114, 2012.

[7] S. Schaal, “Dynamic movement primitives-a framework for motor control in humans and humanoid robotics,” in *Adaptive Motion of Animals and Machines*. Springer, 2006, pp. 261–280.

[8] C. G. Atkeson, A. W. Moore, and S. Schaal, “Locally weighted learning for control,” in *Lazy learning*. Springer, 1997, pp. 75–113.

[9] S. Vijayakumar and S. Schaal, “Locally weighted projection regression: Incremental real time learning in high dimensional space,” in *Proceedings of the Seventeenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., 2000, pp. 1079–1086.

[10] S. Calinon, F. Guenter, and A. Billard, “On learning, representing, and generalizing a task in a humanoid robot,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 2, pp. 286–298, 2007.



(a) F_x (b) F_y

(c) x (d) y

Fig. 5: Actual forces (a,b) felt by the robot along the (c) x and (d) y directions in the execution of the trajectory from the trial of experimental conditions to push a 7.5kg battery back. The black lines correspond to the prediction forces or the reference trajectories, while the red, green and blue lines correspond respectively to *Misplaced-PHMM-Force*, *Misplaced-Memory-Force* and *Misplaced-No-Force*.

[11] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, p. 0278364913495721, 2013.

[12] P. Pastor, M. Kalakrishnan, L. Righetti, and S. Schaal, "Towards associative skill memories," in *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*. IEEE, 2012, pp. 309–315.

[13] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 763–768.

[14] V. Chu, I. McMahon, L. Riano, C. G. McDonald, Q. He, J. Martinez Perez-Tejada, M. Arrigo, N. Fitter, J. C. Nappo, T. Darrell *et al.*, "Using robotic exploratory procedures to learn the meaning of haptic adjectives," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3048–3055.

[15] T. Araki, T. Nakamura, T. Nagai, K. Funakoshi, M. Nakano, and N. Iwahashi, "Autonomous acquisition of multimodal information for online object concept formation by robots," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 1540–1547.

[16] A. Droniou, S. Ivaldi, and O. Sigaud, "Deep unsupervised network for multimodal perception, representation and classification," *Robotics and Autonomous Systems*, vol. 71, pp. 83–98, 2015.

[17] A. Kramberger, A. Gams, B. Nemec, D. Chrysostomou, O. Madsen, and A. Ude, "Generalization of orientation trajectories and force-torque profiles for robotic assembly," *Robot. Auton. Syst.*, vol. 98, no. C, pp. 333–346, Dec. 2017. [Online]. Available: https://doi.org/10.1016/j.robot.2017.09.019

[18] L. Rozo, P. Jiménez, and C. Torras, "Force-based robot learning of pouring skills using parametric hidden markov models," in *Robot Motion and Control (RoMoCo), 2013 9th Workshop on*. IEEE, 2013, pp. 227–232.

[19] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Comput.*, vol. 25, no. 2, pp. 328–373, Feb. 2013. [Online]. Available: http://dx.doi.org/10.1162/NECO_a_00393

[20] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb 1989.

[21] A. D. Wilson and A. F. Bobick, "Parametric hidden markov models for gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 884–900, Sep 1999.

[22] Z. Ghahramani and M. I. Jordan, "Supervised learning from incomplete data via an em approach," in *Advances in Neural Information Processing Systems 6*. Morgan Kaufmann, 1994, pp. 120–127.