BIO Web of Conferences **1**, 00013 (2011)
DOI: 10.1051/bioconf/20110100013

Challenges in adapting imitation and reinforcement learning to compliant robots

Sylvain Calinon

Department of Advanced Robotics (ADVR), Istituto Italiano di Tecnologia (IIT)
sylvain.calinon@iit.it

Abstract

There is an exponential increase of the range of tasks that robots are forecasted to accomplish. (Re)programming these robots becomes a critical issue for their commercialization and for their applications to real-world scenarios in which users without expertise in robotics wish to adapt the robot to their needs. This paper addresses the problem of designing user-friendly human-robot interfaces to transfer skills in a fast and efficient manner. This paper presents recent work conducted at the Learning and Interaction group at ADVR-IIT, ranging from skill acquisition through kinesthetic teaching to self-refinement strategies initiated from demonstrations. Our group started to explore the use of imitation and exploration strategies that can take advantage of the compliant capabilities of recent robot hardware and control architectures.

1. Introduction

While accuracy and speed have for a long time been top of the agenda for robot design and control, the development of new actuators and control architectures is now bringing a new focus on passive and active compliance, energy optimization, human-robot collaboration, easy-to-use interfaces and safety.

The machine learning tools that have been developed for precise reproduction of reference trajectories need to be re-thought and adapted to these new challenges. For planning, storing, controlling, predicting or re-using motion data, the encoding of a robot skill goes beyond its representation as a single reference trajectory that needs to be tracked or set of points that needs to be reached. Instead, other sources of information need to be considered, such as the local variation and correlation in the movement. Also, most of the machine learning tools developed so far are decomposed into an offline model estimation phase and a retrieval/regression phase. Instead, learning in compliant robots should view demonstration and reproduction as an interlaced process that can combine both imitation and reinforcement learning strategies to incrementally refine the task.

The development of compliant robots brings new challenges in machine learning and physical human-robot interaction, by extending the skill transfer problem towards tasks involving force information, and towards systems capable of learning how to cope with various sources of perturbation introduced by the user and the task. We take the perspective that both the redundancy of the robot architecture AND the task can be exploited to adapt a learned movement to new situations, while at the same time improving safety and energy consumption. Through these new physical guidance capabilities, the robot becomes a tangible interface that can exploit the natural teaching tendency of the user (scaffolding, kinesthetic teaching, exaggeration of movements to highlight the relevant features, etc.).

The long-term view is to develop flexible learning tools that will anticipate the ongoing raise of compliant actuators technologies. In particular, we would like to ensure a smooth transition to passive compliant actuators and manipulators that can be safely used in the proximity of users, by considering physical contact and collaborative interaction as key elements in the transfer of skills.

2. Importance of task representation

Learning from interaction in compliant robots is prone to various sources of continuous perturbations. Planning and control require to be considered altogether to allow the robot to make fast decision during the course of the movement. For this reason, we view task movements as an adaptive flow field that the robot follows (instead of a trajectory to track). We take the perspective that the task can be represented as a weighted

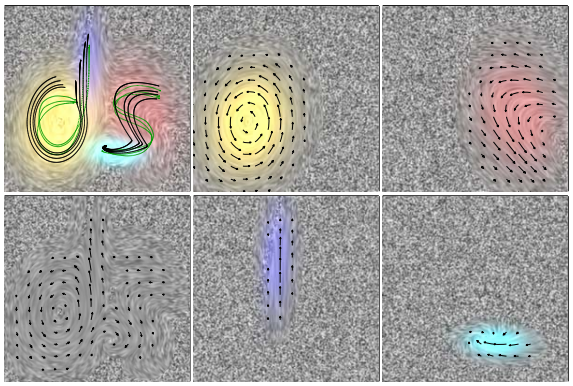

Figure 1: The movement when drawing the letters 'ds' is represented as a superposition of four subsystems, each defined by a matrix $\boldsymbol{A}_i$ and offset vector $\boldsymbol{b}_i$. The superposition rule is driven here by a Gaussian Mixture Model (GMM), in which the likelihood defines the region where each subsystem is active. We see that even in 2D, $\boldsymbol{A}_i$ and $\boldsymbol{b}_i$ can define various types of dynamics (circular fields, sinks, etc.).

combination of linear systems

$$\dot{\boldsymbol{x}} = \sum_i \overbrace{h_i(\boldsymbol{x},t)}^{\substack{\text{scalar weight}\\ \text{(Sec. 4)}}} \overbrace{(\boldsymbol{A}_i\boldsymbol{x}+\boldsymbol{b}_i)}^{\substack{\text{linear subsystem}\\ \text{(Sec. 3)}}}, \qquad (1)$$

where $\boldsymbol{A}_i$ and $\boldsymbol{b}_i$ are respectively full matrices and offset vectors. Fig. 1 presents an example of using such representation to encode a movement.

Several methods can be formulated in this generic manner, and differ in the representation of $\boldsymbol{x}$, in the way $\boldsymbol{A}_i$ and $\boldsymbol{b}_i$ are estimated and constrained, and in the mechanism that combines the different subsystems through scalar weights h_i. We respectively present in Sections 3 and 4 our work on determining appropriate linear subsystems and on weighting those efficiently to transfer a skill.

3. Constraints on linear systems

3.1. Gaussian Mixture Regression (GMR)

Gaussian mixture regression (GMR) has been proposed as a generic approach to handle encoding, recognition, prediction and reproduction in robot programming by demonstration [2].

The parameters of a *Gaussian mixture model* (GMM) of K states are defined by $\{\pi_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^K$, where π_i is the prior probability; $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ represent the center and the covariance matrix of the i-th state.

GMR relies on the learned joint distribution $\mathcal{P}(\boldsymbol{x}^{\mathcal{I}}, \boldsymbol{x}^{\mathcal{O}})$ of the data observed during demonstrations, where the conditional probability $\mathcal{P}(\boldsymbol{x}^{\mathcal{O}}|\boldsymbol{x}^{\mathcal{I}})$ is estimated as another output Gaussian. The retrieved trajectory is continuously differentiable and encapsulates variation and correlation information.

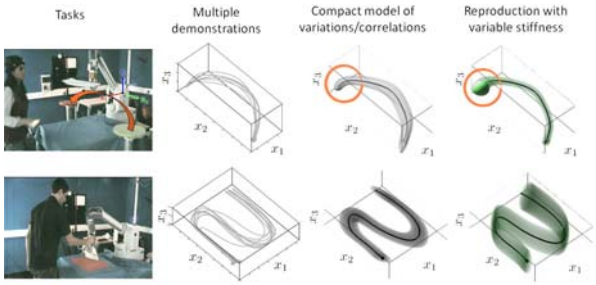

Figure 2: Experiments where the local variations and correlations of the task are used to determine an appropriate stiffness for the automatic reproduction of the task. In the first task, the robot learns to move a tray. Since the demonstrations were more consistent at the end of the movement (when positioning the tray), the stiffness gains progressively increased at the end of the movement. In the second task, an ironing skill is demonstrated to the robot by following an approximate S-shape. Here, the robot extracts from the demonstrations that it is important to be in contact with the table and that a precise tracking of the S-shape is less relevant (more variation in the plane of the table than the vertical direction). The estimated stiffness ellipsoids are then elongated in the vertical direction. See [4] for details.

GMR has mostly been used in two ways: 1) with time as an explicit variable, by learning $\mathcal{P}(t, \boldsymbol{x})$ and retrieving $\mathcal{P}(\boldsymbol{x}|t)$ during reproduction; and 2) as an autonomous system, by learning $\mathcal{P}(\boldsymbol{x}, \dot{\boldsymbol{x}})$ and retrieving $\mathcal{P}(\dot{\boldsymbol{x}}|\boldsymbol{x})$ during reproduction. GMR can be reformulated with the notation of (1) as

$$\dot{\boldsymbol{x}} = \sum_i h_i \left(\overbrace{\boldsymbol{\Sigma}_i^{\dot{\mathcal{X}}\mathcal{X}}(\boldsymbol{\Sigma}_i^{\mathcal{X}})^{-1}}^{\boldsymbol{A}_i} \boldsymbol{x} + \overbrace{\boldsymbol{\mu}_i^{\dot{\mathcal{X}}} - \boldsymbol{\Sigma}_i^{\dot{\mathcal{X}}\mathcal{X}}(\boldsymbol{\Sigma}_i^{\mathcal{X}})^{-1}\boldsymbol{\mu}_i^{\mathcal{X}}}^{\boldsymbol{b}_i} \right). \qquad (2)$$

3.2. Dynamic Movement Primitives (DMP)

The core mechanism of DMP can be formulated as a weighted sum of linear systems, where the linear systems are constrained to act as virtual spring-damper systems (see [1] for more details). Namely, a second order system can be defined by considering the state $\boldsymbol{X} = \left(\begin{smallmatrix}\boldsymbol{x}\\ \dot{\boldsymbol{x}}\end{smallmatrix}\right)$ in which $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ are positions and velocities, and by recursively defining the motion of the system as

$$\overbrace{\begin{bmatrix}\dot{\boldsymbol{x}}\\ \ddot{\boldsymbol{x}}\end{bmatrix}}^{\dot{\boldsymbol{X}}} = \sum_i h_i \left(\overbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \begin{bmatrix}-\kappa^{\mathcal{P}} & 0\\ 0 & \ddots\end{bmatrix} & \begin{bmatrix}-\kappa^{\mathcal{V}} & 0\\ 0 & \ddots\end{bmatrix}\end{bmatrix}}^{\boldsymbol{A}_i} \overbrace{\begin{bmatrix}\boldsymbol{x}\\ \dot{\boldsymbol{x}}\end{bmatrix}}^{\boldsymbol{X}} + \overbrace{\begin{bmatrix}\boldsymbol{0}\\ \kappa^{\mathcal{P}}\boldsymbol{\mu}_i^{\mathcal{X}}\end{bmatrix}}^{\boldsymbol{b}_i} \right). \qquad (3)$$

The two implementation examples given in (2) and (3) show that the representation proposed in (1) can be used to generalize several approaches, and make it possible to introduce new perspectives for the extension and analysis of these different methods.

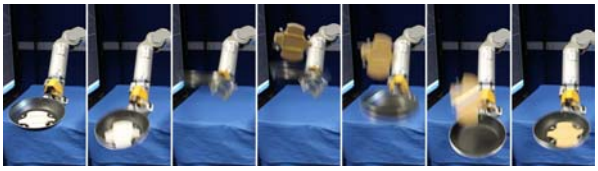

Figure 3: Some skills are difficult to acquire by imitation. Consecutive demonstrations from the user of flipping a pancake result in a training set that does not contain enough consistency to extract the important characteristics of the task. We tested in this experiment if the proposed encoding scheme could also be used with EM-based Reinforcement Learning to let the robot refine the skill on its own. After about 50 trials, the robot learns a movement to flip the pancake with appropriate stiffness matrices. At the beginning of the task, it learns that it needs to be stiff to throw the pancake in the air. At the end of the movement, it learns that it should keep this stiffness in the horizontal plane (to follow the pancake), and compliant in the vertical direction to smoothly catch the pancake. See [5] for details.

In particular, the formulation above allowed us to cope with one of the limitation of DMP by considering in (3) full stiffness matrices $\boldsymbol{K}_i^{\mathcal{P}}$ instead of the scalar gains $\kappa^{\mathcal{P}}$. In the original DMP formulation, each variable acts as a separated system (synchronized by shared weights h_i). This simple structural modification allowed us to consider the local correlations across the different variables. The robot is then driven by

$$\ddot{\boldsymbol{x}} = \sum_{i=1}^{K} h_i \Big[\boldsymbol{K}_i^{\mathcal{P}} (\boldsymbol{\mu}_i^{\mathcal{X}} - \boldsymbol{x}) - \kappa^{\mathcal{V}} \dot{\boldsymbol{x}} \Big], \qquad (4)$$

where $\boldsymbol{K}_i^{\mathcal{P}}$ are virtual full stiffness matrices.

We showed in [4] that the variability and correlation observed across several demonstrations of the same task could be used to estimate the stiffness matrices, see Fig. 2. We also showed in [5] that such compact encoding could be used in Reinforcement Learning to learn a skill requiring different levels of compliance along the task, see Fig. 3. Videos of the experiment are available on `http://programming-by-demonstration.org`.

4. Weighting mechanisms

In [4] and [5], a basic weighting mechanism based on time was used to make smooth transitions between the different subsystems. This section presents our work towards designing more efficient weighting mechanisms in (1) to combine the different subsystems presented in Sec. 3.

Most of the proposed weighting mechanisms are based on the likelihood of Gaussian distributions that need to be rescaled to satisfy the properties of a mixture of experts. If α_i represents the likelihood of state i (e.g., from a GMM or HMM), it needs to be divided by the sum of the other α_i to form a weight h_i satisfying

Figure 4: Bilateral human-robot interaction in which the robot is used by the user as a tangible interface to start/resume tasks and switch between tasks. The robot is gravity-compensated and can be moved without effort. When the user brings the robot in a known region in which a task has previously been demonstrated, the robot recognizes this part of the movement and continues the task actively until the user decides to stop it by physically moving it away from the task, or possibly bringing it towards another task region. See [6] for details.

$\sum_i^K h_i = 1$. We modified this property in [6] to define weights that are independent from each other during reproduction, namely

$$h_i = \frac{\mathcal{N}(\boldsymbol{x};\ \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_k^K \mathcal{N}(\boldsymbol{x};\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \Rightarrow h_i = \frac{\mathcal{N}(\boldsymbol{x};\ \boldsymbol{\mu}_i, c_i\boldsymbol{\Sigma}_i)}{\mathcal{N}(\boldsymbol{\mu}_i;\ \boldsymbol{\mu}_i, c_i\boldsymbol{\Sigma}_i)}, \qquad (5)$$

where c_i is a scaling factor to stay close to $\sum_i^K h_i = 1$ during the whole movement.

The activation functions become semi-independent: they are efficiently organized through the *Expectation-Maximization* (EM) process with (5)-*left*, and they are then considered as independent in the reproduction phase with (5)-*right*. Fig. 4 presents the advantage of this reformulation in the context of bilateral human-robot interaction.

In order to handle various time and space constraints, different weighting mechanisms have been proposed to switch between linear subsystems during the task. On the one hand, a GMM looks at the current position (or state) of the system to determine which subsystem is active (first row of Fig. 5). On the other hand, a representation in time instead activates sequentially the different subsystems independently of where the system is (second row of Fig. 5).

One solution could be to select one or the other model depending on the task requirements, but this would have disadvantages in the case where both time and space are relevant, or when the time-space requirements are changing during the skill. For this reason, we are looking for weighting schemes capable of representing both time and space constraints. HMM provides a solution in-between by augmenting the GMM representation with transition probabilities (third row of Fig. 5). The use of HMM however has a drawback: the formulation gives more importance to the position than to the transition information. This shortcoming can be leveraged by the use of *Hidden Semi-Markov Model* (HSMM) to better model the state duration. Self-transition probabilities are replaced by a statistical model of the duration that the robot stays in each

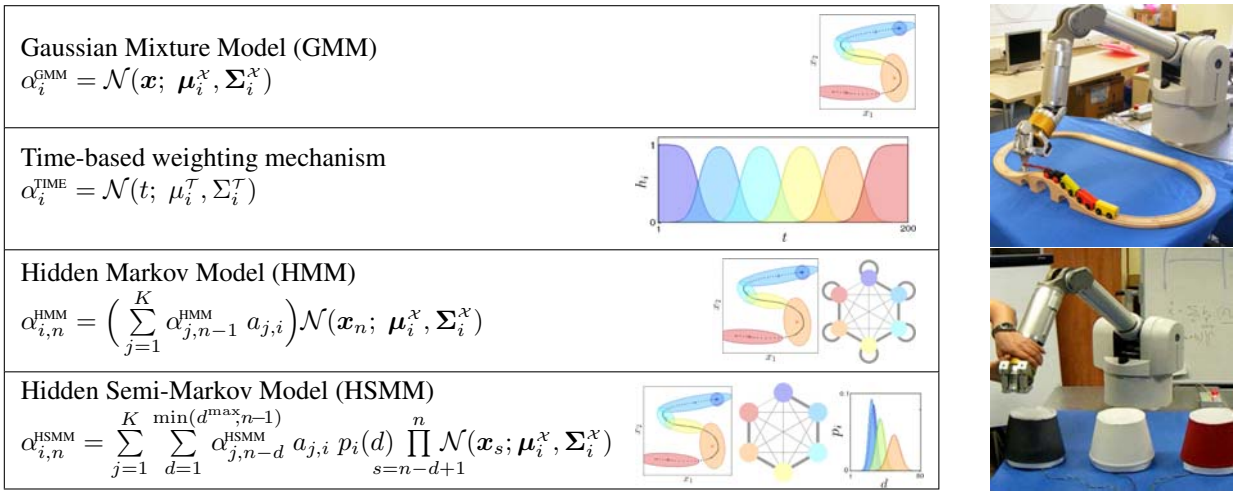

Figure 5: *Left:* Proposed weighting schemes *Right:* Examples of tasks requiring different behaviors to transit between the different subsystems when the robot is faced with continuous sources of perturbation. In the first task, the robot is taught to pull a model train on its track. If the user perturbs the motion (e.g. by holding the robot's end-effector during a few iterations), the robot should take here only spatial information into account to follow the task (the important aspect is to stay on the path). In the second task, the robot is taught to play a melody by pressing three big keys. Here, if the user perturbs the movement, the robot should take into consideration timing constraints to recover from the time delay introduced by the perturbation, by possibly skipping irrelevant parts of the movement (time should be here the component that drives the system). See [3] for details.

state (last row of Fig. 5). Similarly to standard HMM, the model can be trained by EM.

We proposed in [3] to use this HSMM formulation to encapsulate duration and position information in a robust manner, with parameterization on the involvement of time and space constraints, see Fig. 5. It allowed us to cover a wide range of weighting mechanisms in a parameterizable way. Compared to time-based weights, it is now possible to learn where to place the activation functions in time, and to learn the smoothness of the transition to transit to the next subsystems. It can also handle periodic and discrete movements without modifying the parameterization.

5. Conclusion

This paper presented several directions of research from the Learning and Interaction group at ADVR-IIT. These different research aspects are linked to the perspective that a superposition of linear subsystems can represent the realm of the continuous world, and that such representation can be manipulated/visualized by users and help at generalizing the skills to new situations.

6. Acknowledgments

The paper contains material from [3–6], authored or co-authored by Petar Kormushev, Antonio Pistillo, Irene Sardellitti and Darwin G. Caldwell.

References

[1] S. Calinon, F. D'halluin, D. G. Caldwell, and A. G. Billard. Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework. In *Proc. IEEE-RAS Intl Conf. on Humanoid Robots (Humanoids)*, pages 582–588, Paris, France, 2009.

[2] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard. Learning and reproduction of gestures by imitation: An approach based on hidden Markov model and Gaussian mixture regression. *IEEE Robotics and Automation Magazine*, 17(2):44–54, June 2010.

[3] S. Calinon, A. Pistillo, and D. G. Caldwell. Encoding the time and space constraints of a task in explicit-duration hidden Markov model. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA, September 2011.

[4] S. Calinon, I. Sardellitti, and D. G. Caldwell. Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 249–254, Taipei, Taiwan, October 2010.

[5] P. Kormushev, S. Calinon, and D. G. Caldwell. Robot motor skill coordination with EM-based reinforcement learning. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 3232–3237, Taipei, Taiwan, October 2010.

[6] A. Pistillo, S. Calinon, and D. G. Caldwell. Bilateral physical interaction with a robot manipulator through a weighted combination of flow fields. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA, September 2011.