The Chief Cook Learning Robot

Sylvain Calinon, Eric Sauser, Marino Alge, Florent Guenter and Aude Billard

***Abstract*— This video presents a Chief Cook Robot learning to cook an omelet by whipping eggs, cutting ham and grating cheese. Through the use of a probabilistic model using Gaussian Mixture Model (GMM) and Gaussian Mixture Regression (GMR), the robot progressively learns to generalize a skill to various situations by being robust to dynamic perturbations.**

I. INTRODUCTION

Robot Programming by Demonstration (RbD) covers methods by which a robot learns new skills through human guidance. We developed a framework to teach gestures to a HOAP-3 humanoid robot by providing a set of demonstrations performed in slightly different situations. Through the use of *Gaussian Mixture Regression* (GMR), the robot can extract autonomously the essential characteristics of a set of trajectories captured through the demonstrations [1], [2]. The approach can be used generically for different robot architectures and allows to simultaneously handle constraints on multiple objects in task space and in joint space [3].

Several regression techniques based on a probabilistic representation of the dataset such as *Locally Weighted Regression* (LWR) or *Gaussian Process Regression* (GPR) were proposed in robotics to generalize over a set of demonstrations. Our approach follows a similar strategy by using *Gaussian Mixture Model* (GMM) and *Gaussian Mixture Regression* (GMR) to respectively encode a set of trajectories and retrieve a smooth generalized version of these trajectories with associated variabilities, where the dataset is encoded in a compact form learned through the efficient *Expectation-Maximization* (EM) algorithm. For the applications that we consider, the principal advantages of this method are: (1) it allows to deal with recognition and reproduction issues in a common probabilistic framework; and (2) the learning process is distinct from the retrieval process, where a simple and fast learning process is first used to model the demonstrated skill during the phases of the interaction that do not require real-time computation (i.e. after the demonstrations), and where a faster regression process is then used for controlling the robot in an online manner during the reproduction phases. For an exhaustive review and comparisons of our approach with the different methods proposed above, the interested reader can refer to [4].

This work was supported by the European Commission as part of the Robot@CWE project (`http://www.robot-at-cwe.eu`) under contract FP6-2005-IST-5, and as part of the FEELIX GROWING project (`http://www.feelix-growing.org`) under contract FP6 IST-045169.

The authors are with the Learning Algorithms and Systems Laboratory (LASA), Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland. Contact: `sylvain.calinon@epfl.ch`

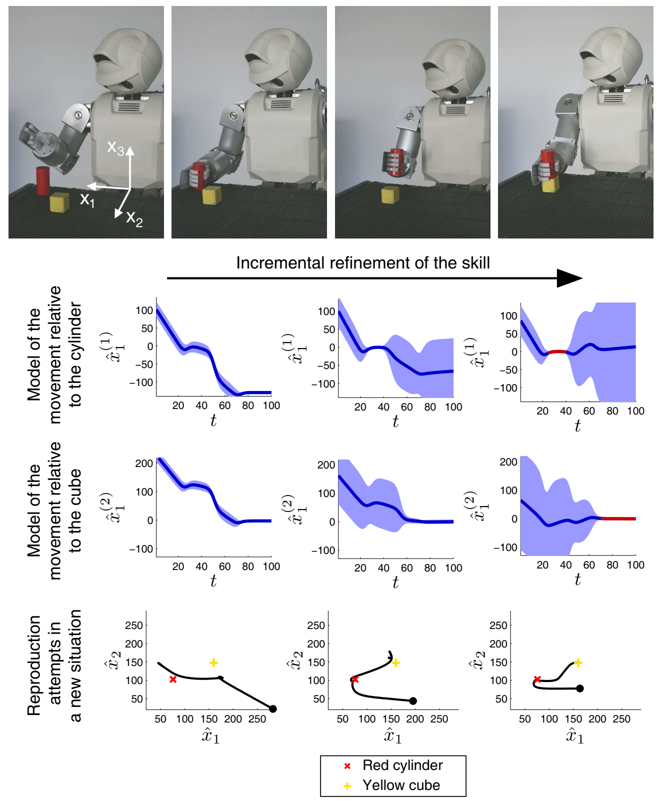

Fig. 1. Incremental refinement of a stacking task that consists of grasping a first object (a cylinder) and putting it on a second object (a cube). The robot learns generalized trajectories coded in frames of reference located on the objects that are manipulated.

II. EXTRACTING THE TASK CONSTRAINTS THROUGH A STATISTICAL APPROACH

To extract the task constraints from a set of trajectories characterized by a set of temporal values t and spatial values x, the joint probability $P(t,x)$ is first encoded in a *Gaussian Mixture Model*, trained incrementally through *Expectation-Maximization* (EM) algorithm. A generalized version of the trajectories can then be estimated by computing the conditional Gaussian distribution $P(x|t)$. The constraints of the motion at each time step t is then given by a mean value $\hat{x}$ and associated covariance matrix $\hat{\Sigma}$. If multiple constraints are considered (e.g., by considering relative motions $x^{(1)}$ and $x^{(2)}$ with respect to two different objects), the resulting constraint is estimated by first retrieving the two Gaussian distributions $P(x^{(1)}|t)$ and $P(x^{(2)}|t)$ and then computing the product of the two Gaussian distributions to retrieve a controller $\hat{x}$ by automatically computing an optimal tradeoff with respect to the constraints extracted for the two objects. Thus, through the use of the compact GMM representation, the robot can extract autonomously the essential characteristics of a set of trajectories captured through multiple demonstrations. The regression process is then used to retrieve a controller satisfying several constraints simultaneously (either in joint space or in task space).

Fig. 1 illustrates the generalization and reproduction methods with a task involving manipulation and displacement of objects. In this experiment, the skill is represented as constraints in task space by considering the right hand path relative to two objects tracked by the robot in its environment through a stereoscopic vision system. The three columns of the graph correspond respectively to a representation of the task constraints after 1, 3 and 6 demonstrations. The first two rows of the graph show the refinement of the GMR model representing the constraints for the cylinder (*first row*) and for the cube (*second row*) along the movement. After a few demonstrations, the trajectories relative to the two objects are highly constrained for particular subparts of the task, namely when reaching for the cylinder (thin envelope around time step 30) and when placing it on the cube (thin envelope around time step 100). The last row shows the robot's reproduction attempts (after 1, 3 and 6 demonstrations) for a new situation that has not been demonstrated. After 6 demonstrations, the robot correctly reproduces the essential characteristics of the skill, namely reaching for the cylinder and dropping it on the cube. We see that by encapsulating the task constraints through GMR, the robot can gradually refine the model of the skill and reproduce it in new situations (new initial positions of objects).

III. Handling of Perturbation Through a Dynamical System

The probabilistic approach is then extended to the use of a dynamical system as in [5] to depart from time in the reproduction process and to allow the robot to deal with external perturbations. An illustration of the approach is presented in Fig. 2. This approach also allows to consider a more flexible scaffolding process where the user can provide partial demonstrations to the robot and where the robot can deal efficiently with perturbations. An example of application is presented in Fig. 3.

References

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

[2] S. Calinon and A. Billard, "What is the teacher's role in robot programming by demonstration? - Toward benchmarks for improved learning," *Interaction Studies*, vol. 8, no. 3, pp. 441–464, 2007.

[3] ——, "A probabilistic programming by demonstration framework handling constraints in joint space and task space," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, September 2008.

[4] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 1371–1394.

[5] M. Hersch, F. Guenter, S. Calinon, and A. Billard, "Dynamical system modulation for robot learning via kinesthetic demonstrations," *IEEE Trans. on Robotics*, 2008, in press.

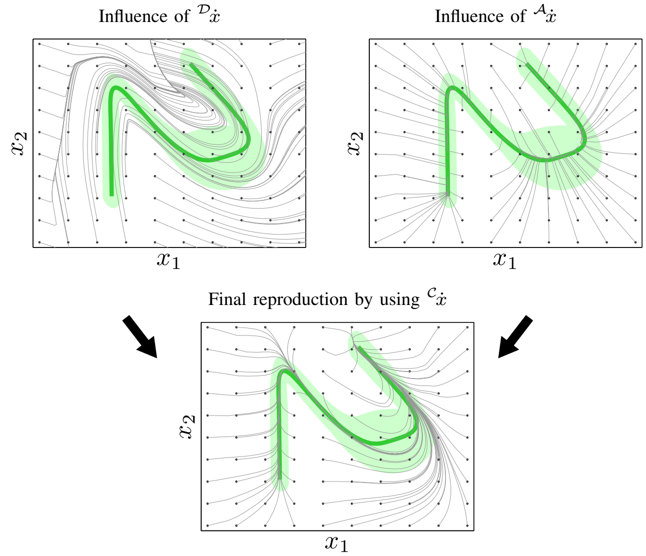

Fig. 2. Illustrative example for the reproduction of a motion where the dataset $\{t, x, \dot{x}\}$ has first been encoded in a Gaussian Mixture Model (GMM), and where Gaussian Mixture Regression (GMR) has been used to retrieve $P(x, \dot{x}|t)$, which allows to define a dynamics component $^{\mathcal{D}}\dot{x}$ estimating the velocity command required at each iteration to follow the dynamics learned by the system (with respect to the current position), and a trajectory component $^{\mathcal{A}}\dot{x}$ used by the system to come back to a known position in task space (i.e. the learned trajectory is used here as an attractor). *First row:* Influence of the two velocity commands (when used separately) when starting from several initial positions equally distributed in the workspace. On the one hand, the dynamics component follows the learned motion but tends to become unstable after a few iterations or when starting from an unexplored position. On the other hand, the trajectory component acts as an attractor to the closest point of the generalized trajectory. *Second row:* Reproduction behaviour by considering simultaneously at each iteration the influence of the two velocity components.

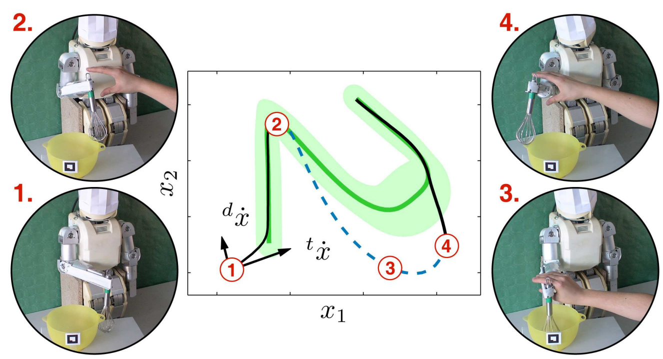

Fig. 3. Illustration of the scaffolding process used to teach new manipulation skills to a humanoid robot. The learned motion is represented in thick line with an associated surrounding envelope representing the extracted constraints in a continuous form. *1.* The robot starts reproducing the learned skill from a new initial position. *2.* At some point during the reproduction, the user holds the robot's arm and provides support for the reproduction of the skill. *3.* The robot lets the user move manually the selected motors (kinesthetic teaching) and records proprioceptive information about its own body motion, while trying to follow the demonstrated motion with the remaining motors that are not controlled by the user. *4.* By releasing the robot's arm, the user then lets the robot pursue the remaining part of the motion on its own. We see here that the robot smoothly comes back to the learned motion. This new demonstration is then used by the robot to refine its model of the skill.