

# On Learning the Statistical Representation of a Task and Generalizing it to Various Contexts

Sylvain Calinon, Florent Guenter and Aude Billard

LASA Laboratory, School of Engineering (EPFL), CH-1015 Lausanne, Switzerland

{sylvain.calinon, florent.guenter, aude.billard}@epfl.ch

**Abstract**—This paper presents an architecture for solving generically the problem of extracting the constraints of a given task in a programming by demonstration framework and the problem of generalizing the acquired knowledge to various contexts. We validate the architecture in a series of experiments, where a human demonstrator teaches a humanoid robot simple manipulatory tasks. First, the combined joint angles and hand path motions are projected into a generic latent space, composed of a mixture of Gaussians (GMM) spreading across the spatial dimensions of the motion. Second, the temporal variation of the latent representation of the motion is encoded in a Hidden Markov Model (HMM). This two-step probabilistic encoding provides a measure of the spatio-temporal correlations across the different modalities collected by the robot, which determines a metric of imitation performance. A generalization of the demonstrated trajectories is then performed using Gaussian Mixture Regression (GMR). Finally, to generalize skills across contexts, we compute formally the trajectory that optimizes the metric, given the new context and the robot's specific body constraints.

**Index Terms**—Robot Programming by Demonstration, Imitation Learning, Humanoid Robots, Gaussian Mixture Model, Metric of Imitation.

## I. INTRODUCTION

Recent advances in *Robot Programming by Demonstration* (RbD), also referred to as *Learning by Imitation*, have identified a number of key issues that need to be solved for ensuring a generic approach to transferring skills across various agents and situations. These have been formulated as a set of generic questions, namely *what-to-imitate*, *how-to-imitate*, *when-to-imitate* and *who-to-imitate* [1]. This paper presents an architecture for solving generically the problems of extracting the constraints of a given task (*what-to-imitate*) and the problem of generalizing the acquired knowledge to various contexts (part of the *how-to-imitate* issue).

Other approaches have previously attempted to find optimal controllers which would reproduce a set of high-level constraints [2]–[4]. In each of these previous works, the set of constraints were given, and in any portion of the task, only a unique constraint could be satisfied at a time (e.g. absolute/relative constraints on position/orientation). In our work we introduce a generic framework which allows for the extraction of a time-dependent continuous representation of the constraints.

## II. THE EXPERIMENTAL SCENARIO

Fig. 1 presents the setup of our experiments. Fig. 2 illustrates the issues that we address in this paper for a *Chess*

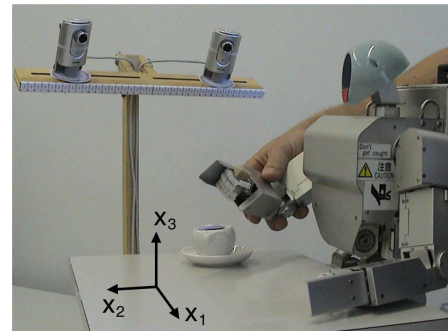


Fig. 1. Experimental setup. The robot is taught a new task through kinesthetics, i.e. by the user moving the robot's arm through the task. A fixed pair of cameras tracks the 3-D position of the target (a color patch), here, the robot's mouth.

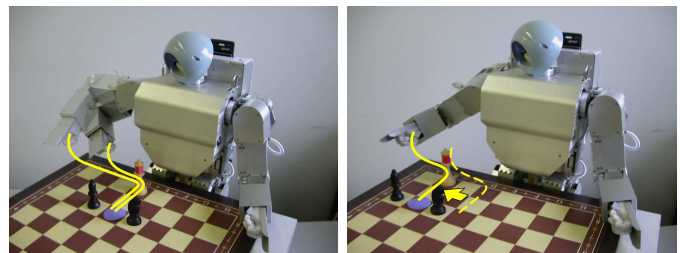


Fig. 2. *Left*: Illustration of the *what-to-imitate* issue. With two demonstrations of a *Chess Task*, we see that the constraints are varying along the motion. To approach the chess piece, a large set of paths are possible, depending on the initial position. Grabbing and pushing the piece requires higher constraints, i.e. the paths do not change much between two consecutive demonstrations. *Right*: Illustration of the *how-to-imitate* issue. To re-use the learned skill in different situation (here, different position of the chess piece), a cost function is used to select a controller that best fulfills the task constraints along the trajectories.

*Task*. Left picture shows the path followed by the hand of the robot during training when starting from two different initial locations. In order to extract the relevance of each variable collected during the demonstration (i.e. to determine *what-to-imitate*), the robot computes the spatio-temporal variations and correlations across the variables. In the *Chess Task*, this analysis reveals weak correlations at the beginning of the motion, as there is a large set of possible paths to reach for the chess piece, depending on the hand's initial position. In contrast, the analysis reveals strong spatio-temporal correlations for, first, grabbing the piece and, then, pushing it toward the desired location without hitting the other pieces lying on the

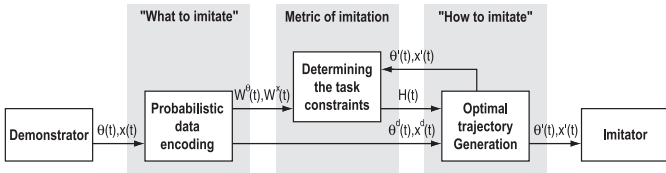


Fig. 3. Information flow across the complete system.

chessboard.

Fig. 2 right illustrates the *how-to-imitate* issue. Once trained to perform a particular task in a particular context, the robot must be able to generalize and to reproduce the same task in a different context. Here, the robot must be able to grab and place the White Queen to a specific position (between the Black Knight and Bishop), wherever these may be located on the chess board. However, joint angles and hand path can be mutually exclusive in the imitator workspace, i.e. both constraints can not be fulfilled at the same time. Depending on the situation, the robot may have to find a very different joint angles configuration than that first demonstrated, in order to avoid breaking its arm. To do so, the robot computes the trajectory that finds a optimal trade-off between satisfying the constraints of the task (spatio-temporal correlations across the variables) and its own body constraints.

### III. THE ARCHITECTURE

Fig. 3 gives an overview of the input-output flow through the complete model. The model is composed of the following processes:

- **Probabilistic data encoding:** The signals are encoded in a two-stage process: First, we determine the latent space of the motions, by estimating the optimal *Gaussian Mixture Model* (GMM) to encode the motions. Second, we encode the dynamics of the motions (i.e. the transition across the states of the GMM), using *Hidden Markov Model* (HMM).
- **Determining the task constraints:** We then compute a time-dependent measure of the relative importance of each variable, and the dependencies across the variables, using the probabilistic description of the task. This measure is used to determine a *metric* of imitation performance for the particular task.
- **Optimal trajectory generation:** We then compute (using Lagrange optimization) the trajectory that optimizes the metric for a given context, given a set of robot's body constraints.

Next, we describe the computation carried out in each of these modules.

#### A. Probabilistic data encoding

To avoid making too many assumptions on the spatio-temporal variability of the dataset, we should, ideally, use a HMM with the most general architecture, such as a fully-connected continuous HMM, with full covariance matrix, describing the output variables distribution. However, using

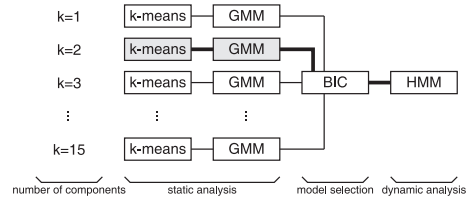


Fig. 4. Schematic of the *probabilistic data encoding* process. The selection of the number of components required to encode a task is performed in the static phase (estimation of several *Gaussian Mixture Models*). The dynamic analysis is then provided by a single estimation of *Hidden Markov Model*.

such a model requires the estimation of a large set of parameters, which can be achieved only with a large dataset. However, to program efficiently a robot by demonstration, the demonstrator should not have to perform more than a few (5 to 10) demonstrations. This means that the set of parameters to learn is quite large, compared to the amount of training data.

The standard *Expectation-Maximization* (EM) algorithm used to estimate the HMM parameters starts from initial estimates, and converges to the nearest local maximum of the likelihood function. Thus, initialization highly affects the model performance. To better estimate the state distribution of the HMM, we perform first a rough clustering of the data using *k-means*, as in [5]. Next, we estimate a *Gaussian Mixture Model* (GMM) by EM, using the *k-means* clusters at initialization. Finally, the dynamics, i.e. transitions across states, are encoded in a HMM, with the GMM state variable distribution (see Fig. 4).

1) **Gaussian Mixture Model (GMM):** A dataset of  $N$  data of dimensionality  $D$ ,  $X = \{\vec{x}(t_1), \vec{x}(t_2), \dots, \vec{x}(t_N)\}$  with  $\vec{x}(t_n) \in \mathbb{R}^D$  is modelled by a multivariate Gaussian mixture of  $K$ -components<sup>1</sup> (see e.g. [6]):

$$p(\vec{x}(t_n)) = \sum_{k=1}^K \pi_k \mathcal{N}(\vec{x}(t_n); \vec{\mu}_k, \Sigma_k)$$

where  $\pi_k$  is the prior probability on the Gaussian component  $k$ , and  $\mathcal{N}(\vec{x}(t_n); \vec{\mu}_k, \Sigma_k)$  is the  $D$ -dimensional Gaussian density of component  $k$ .  $\vec{\mu}_k$  and  $\Sigma_k$  are, respectively, the mean and covariance matrix of the multivariate Gaussian  $k$ .  $\{\pi_k, \vec{\mu}_k, \Sigma_k\}$  are estimated using the *Expectation-Maximization* (EM) algorithm.

2) **Model selection:** The optimal number of components  $K$  in a model may not be known beforehand. We need to determine a trade-off between optimizing the model's likelihood (a measure of how well the model fits the data) and minimizing the parameters (i.e. the number of states used to encode the data). To determine the number of states in a HMM, heuristic methods are often used, sometimes not adequately tuned for HMM. In our approach, model selection is performed in the GMM initialization phase. Multiple GMMs

<sup>1</sup>The process is applied indifferently to joint angles  $\theta$  and hand position  $X$

are estimated, the best model is selected, and a single HMM estimation is performed.

We are using the *Bayesian Information Criterion* (BIC) [7] to select the optimal number of components  $K$ :

$$S_{BIC} = -\mathcal{L} + \frac{n}{2} \log(N) \quad (1)$$

where  $\mathcal{L}$  is the log-likelihood of the model,  $n$  is the number of free parameters required for a mixture of  $K$  components with full covariance matrix, i.e.  $n = (K - 1) + K(D + \frac{1}{2}D(D + 1))$ .  $N$  is the number of  $D$ -dimensional datapoints. The first term of the equation measures how well the model fits the data, while the second term is a penalty factor that aims at keeping the total number of parameters low. In our experiments, we compute a set of candidate GMMs with up to 15 states and keep the model with the minimum score (see Fig. 4).

3) **Hidden Markov Model (HMM)**: Similarly to Gaussian Mixture Models, Hidden Markov Models use a mixture of multivariate Gaussians to describe the distribution of the data. The difference is that HMM also encapsulate the transitions probabilities between the Gaussians. It offers, thus, a way of describing probabilistically the temporal variations of the data<sup>2</sup>.

Let  $\{\mathbf{\Pi}, \mathbf{A}, \mathbf{B}\}$  be, respectively, the initial state distribution, the transition probabilities between the states (or components), and the multivariate output data distribution. In our experiments, we compute only  $\{\mathbf{\Pi}, \mathbf{A}\}$  by *Baum-Welch* algorithm, and set  $\mathbf{B} = \{\bar{\mu}_k, \Sigma_k\}_{k=1}^K$ , where  $\{\bar{\mu}_k, \Sigma_k\}_{k=1}^K$  are the state distributions learned by the GMM.

Once trained, the HMM can be used to recognize gestures. In our experiments, this is used to decide whether a new demonstration belongs or not to the same task, following an approach similar to [9]. In order to measure the similarity between a new gesture and the ones encoded in the model, we run the *forward-algorithm*, an iterative procedure to estimate the likelihood that the observed data could have been generated by the model.

4) **Gaussian Mixture Regression (GMR)**: To reconstruct a signal from the GMM/HMM encoding, after training and generalization over the demonstrations, we apply a *Gaussian Mixture Regression* (GMR), see e.g. [10]. Previous work suggested an "averaging" approach to retrieve human motion sequences from HMM, see e.g. [3]. This approach did not provide satisfying results with our dataset, essentially because it required a very large amount of generated sequences (more than 1000) to retrieve smooth trajectories that can be run on the robot. Another side-effect of the averaging process is that it tends to cut-off and smooth the local minima and maxima of the signals, which can be essential to reproduce human gestures. The clear advantage of GMR over a stochastic retrieval process is that it provides a fast and analytic way to reconstruct the "best" sequence from a Gaussian model.

<sup>2</sup>People unfamiliar with HMM should refer to [8]

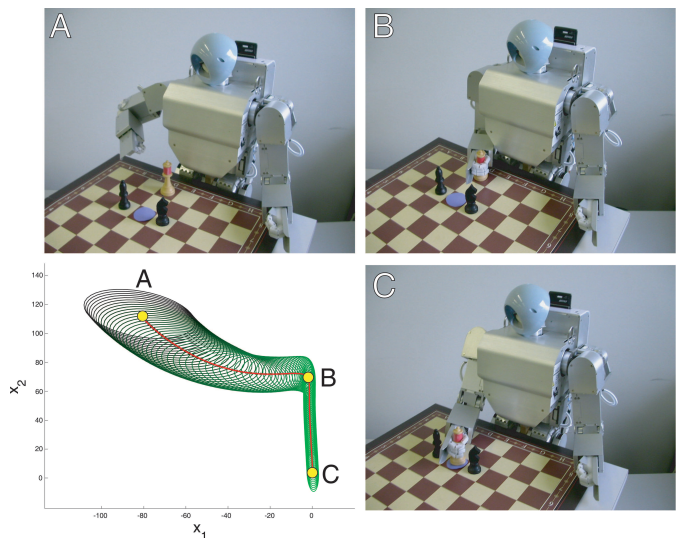


Fig. 5. A, B, C: Snapshots of 3 steps during the reproduction of the *Chess Task*. Bottom left: in solid line, projection along the 2 first dimension of the latent space of the desired trajectory  $\vec{x}^d(t)$  for the robot's hand. The continuous time-dependent weight matrix  $\mathbf{W}^x(t)$  is depicted as a superposition of ellipses, representing the task constraints along the trajectory.

For a  $D$ -dimensional variable  $\vec{x} \in \mathbb{R}^D$ , the means and covariance matrices given by the GMM/HMM representation for component  $k$  are given by  $\bar{\mu}_{kX}^H$  and  $\Sigma_{kX}^H$ . The regression is done along the time index. We compute the means and covariance matrices of the set of observations  $\{t, \vec{x}(t)\}$  with dimension  $(D+1)$ . Note that sole the time-indexed covariances matrices and means are estimated, since the rest of the means and covariance matrices  $\{\bar{\mu}_{kX}^H, \Sigma_{kX}^H\}$  have already been estimated:

$$\begin{aligned} \bar{\mu}_k^R &= \{\mu_{kt}^R, \mu_{kx_1}^H, \mu_{kx_2}^H, \dots, \mu_{kx_D}^H\} \\ \Sigma_k^R &= \begin{pmatrix} \Sigma_{kt}^R & \Sigma_{ktX}^R \\ \Sigma_{kXt}^R & \Sigma_{kX}^H \end{pmatrix} \end{aligned}$$

The *Gaussian Mixture Regression* estimates:

$$\begin{aligned} \vec{x}^d(t) &= \sum_{k=1}^K \beta_k(t) \vec{x}_k^d(t) \\ \beta_k(t) &= \frac{\pi_k \mathcal{N}(t; \mu_{kt}^R, \Sigma_{kt}^R)}{\sum_{i=1}^K \pi_i \mathcal{N}(t; \mu_{it}^R, \Sigma_{it}^R)} \\ \vec{x}_k^d(t) &= \bar{\mu}_{kX}^H + \Sigma_{kXt}^R \Sigma_{kt}^R^{-1} (t - \mu_{kt}^R) \end{aligned}$$

$\vec{x}_k^d(t)$  are the regression output for each associated Gaussian component  $k$ ,  $\beta_k(t)$  the corresponding weight, that measures the relative influence of component  $k$ , and  $\pi_k$  the prior probability.  $\vec{x}^d(t)$  is the *desired trajectory*, a generalized form of the motion learned during training. The time-dependent inverse covariance on  $\vec{x}(t)$  is estimated by:

$$\mathbf{W}^x(t) = \left( \sum_{k=1}^K \beta_k(t) \Sigma_k^R \right)^{-1}$$

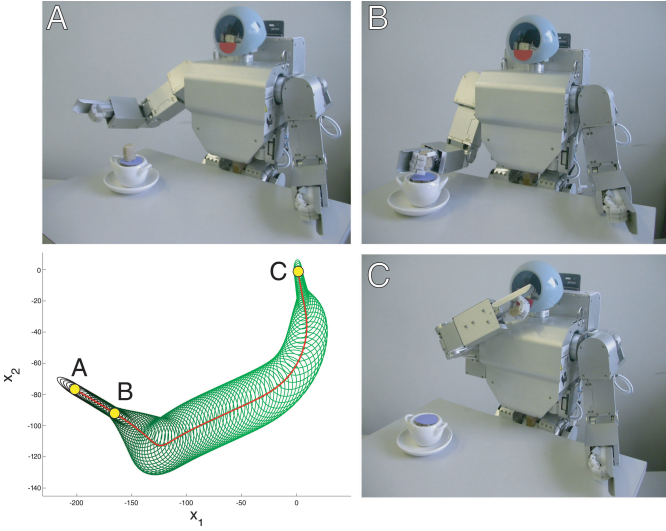


Fig. 6. Probabilistic encoding of the *Sugar Task* (bottom left), by projecting along the first 2 dimensions of  $\vec{x}^d(t)$ . The continuous generalized trajectory  $\vec{x}^d(t)$  is represented in solid line, with associated weight matrix  $\mathbf{W}^x(t)$  depicted as a superposition of ellipses.

### B. Determining the task constraints

In previous work, we proposed a general formalism for determining the cost function of an imitation task [11], [12]. Here, we extend this work and consider a time-dependent version of the original cost function  $H$ . This generic cost function measures the variations of the constraints and of the dependencies across the variables over time. It is continuous, positive definite and can be estimated at any time along the trajectory, by interpolating between the finite set of Gaussians. To have a proper reproduction of the task, the constraints characterizing it are thus defined by: 1) a measure of the variations that we can admit for each signal, and 2) a measure of the correlations that we can admit across the different signals. In our work, the constraints are thus defined statistically as a generalized multivariate signal with associated time-dependent covariance matrix.

Let  $\{\vec{\theta}^d(t), \vec{x}^d(t)\}$  be the desired trajectories for the joints and hand path, generalized forms of the signals gathered during the demonstrations. Let  $\{\vec{\theta}(t), \vec{x}(t)\}$  be the candidate trajectories for reproducing the motions. The metric of imitation performance (cost function for the task)  $H$  is given by:

$$H = \left( \vec{\theta}(t) - \vec{\theta}^d(t) \right)^T \mathbf{W}^\theta(t) \left( \vec{\theta}(t) - \vec{\theta}^d(t) \right) + \left( \vec{x}(t) - \vec{x}^d(t) \right)^T \mathbf{W}^x(t) \left( \vec{x}(t) - \vec{x}^d(t) \right) \quad (2)$$

$H=0$  corresponds to a perfect reproduction.  $\mathbf{W}^\theta(t)$  ( $4 \times 4$  matrix) and  $\mathbf{W}^x(t)$  ( $3 \times 3$  matrix) give a measure of the relative importance of each set of variables, and covariance across the different variables (see section III-A.4).

### C. Optimal trajectory generation

Once the cost function and the relative influence of each constraint have been determined, we generate a trajectory that

is optimal with respect to the cost function  $H$ , by taking into account the body constraints of the robot.

The forward kinematics of the robot is given by  $\vec{x} = f(\vec{\theta})$ . For complex manipulators such as humanoid robot arms, the inverse kinematics (IK) problem is under-constrained, i.e. the computation may have no solutions (degenerate case) or multiple solutions. In this paper, we used the pseudo-inverse with optimization method proposed in [13] and adapt its form to optimize our global cost function  $H$ . We consider an iterative, locally linear, solution to the IK equation, such that<sup>3</sup>:

$$\dot{x}(t) = \mathbf{J} \cdot \dot{\theta}(t) \quad (3)$$

where  $\dot{\theta}(t) = \theta(t) - \theta(t-1)$  and  $\dot{x}(t) = x(t) - x(t-1)$  are the velocity vectors of the joint angles and the hand path.  $\mathbf{J}$  is the Jacobian ( $4 \times 3$  matrix).

The problem is now reduced to finding a minimum of (2) when subjected to (3). Since  $H$  is a quadratic function, the problem can be solved analytically by Lagrange optimization. After substituting  $c_1(t) = \theta^d(t) - \theta(t-1)$  and  $c_2(t) = x^d(t) - x(t-1)$  in (2), we define the *Lagrangian* as:

$$L(\dot{\theta}, \dot{x}, \lambda) = (\dot{\theta} - c_1)^T \mathbf{W}^\theta (\dot{\theta} - c_1) + (\dot{x} - c_2)^T \mathbf{W}^x (\dot{x} - c_2) + \lambda^T (\dot{x} - \mathbf{J} \dot{\theta}) \quad (4)$$

where  $\lambda$  is the vector of associated Lagrange multipliers. By computing  $\nabla L(\dot{\theta}, \dot{x}, \lambda) = 0$ , we find (derivation along  $\lambda, \dot{\theta}, \dot{x}$ ):  $\dot{x} = \mathbf{J} \dot{\theta}$ ,  $-2\mathbf{W}^\theta (\dot{\theta} - c_1) - \mathbf{J}^T \lambda = 0$ ,  $-2\mathbf{W}^x (\dot{x} - c_2) + \lambda = 0$

Solving for  $\dot{\theta}$ , we obtain:

$$\dot{\theta} = (\mathbf{W}^\theta + \mathbf{J}^T \mathbf{W}^x \mathbf{J})^{-1} (\mathbf{W}^\theta c_1 + \mathbf{J}^T \mathbf{W}^x c_2) \quad (5)$$

We can then recompute the joint angle trajectories, using  $\theta(t) = \theta(t-1) + \dot{\theta}(t)$ . The final gesture  $\{\vec{\theta}'(t), \vec{x}'(t)\}$  is computed and evaluated using this method, and reproduced by the robot.

## IV. EXPERIMENTS

We conducted two experiments to demonstrate the validity of our model for teaching a humanoid robot simple manipulatory tasks. The tasks consisted in *moving a chess piece on a chessboard* (see Fig. 2 and 5), and *bringing a piece of sugar to the mouth* (see Fig. 1 and 6). Note that, in these experiments, control affected only the 4 degrees of freedom of the arm and the opening and closing of the robot's hand was hard-coded and activated by the user. In each case, the robot was shown the task 5 times. Hand paths  $\vec{x}(t)$  were encoded in a Cartesian space whose origin is at the goal position (end-position of the chess piece and of the mouth). Once trained, the robot was required to reproduce each task under different constraints, by placing the goals at different locations in the robot's workspace. This procedure aimed at demonstrating the robustness of the system when the constraints were transposed to different locations within the robot's workspace.

<sup>3</sup>For clarity in the notation, we will omit the vectorial notations and the time index for the rest of the developments.

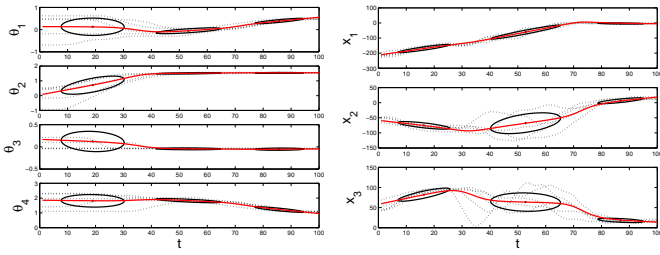


Fig. 7. *Left*: Joint angle trajectories for the *Chess Task*. *Right*: Hand paths for the *Sugar Task*. The dotted lines represent the different demonstrations, while the solid line represent the reconstructed trajectory. The ellipses represent the 3 multivariate Gaussians in the GMM.

### A. Experimental setup

The experiments were conducted with a Fujitsu HOAP-2 humanoid robot with 25 degrees of freedom (DOF), of which only the 4 DOFs of the right arm were required in the experiments. The remaining DOFs of the torso and legs were set to a constant position, so as to support the robot in an upright posture, facing a table, see Fig. 2.

A color-based<sup>4</sup> stereoscopic vision system tracks the 3D-position of pre-defined objects used in the experiments, at a rate of 15Hz and an accuracy of 10mm. The system uses two *Phillips* webcams with a resolution of 320x240 pixels, see Fig. 1. For the *Chess Task*, a marker placed at the desired final position of the chess piece is tracked. For the *Sugar Task*, a marker placed on the robot’s mouth is tracked.

In the experiments reported here, the robot was taught through kinesthetics, i.e. by the demonstrator moving its arm through the task. To achieve this, the robot’s motors were set in a passive mode, whereby each limb could be moved by the human demonstrator. The kinematics of each joint motions was recorded at a rate of 1000Hz during the demonstration and was, then, downrated to 15Hz to match the tracking rate of the cameras.

### B. Experimental results

Fig. 7 left shows the encoding and reconstruction of the joint angle trajectories for the *Chess Task*. 3 states were found to encode optimally the trajectories, according to the BIC criterion, see Section III-A.2. These correspond roughly to the different phases of the motion (approaching the object, adjusting the hand posture and pushing the object). When looking at the width of the ellipses (which represent the variance along the joint angle trajectories across the demonstrations), we observe that the first phase of the motion shows high variability (and, thus, does not require to follow a precise path), whereas the two other phases are much more constrained. This change in the constraints along the task will be encapsulated in the weights  $W^\theta(t)$  of the cost function  $H$ .

Fig. 7 right shows the encoding and reconstruction of the hand path for the *Sugar Task*. 3 states were also found to

<sup>4</sup>The tracking is based on color segmentation in the YCbCr color space. Only Cb and Cr are used, to be robust to changes in luminosity.

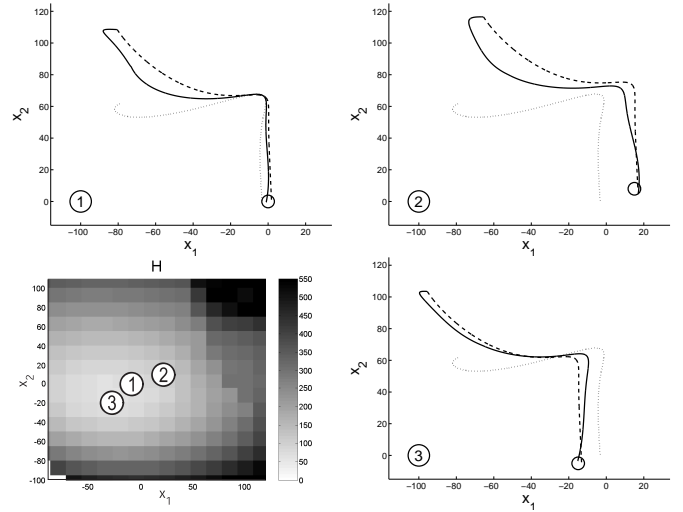


Fig. 8. *Bottom left*: Color map with mean values taken by the cost function  $H$  for the *Chess Task*, for various final locations on the chess board. 1,2,3: Reconstructed hand paths  $\bar{x}^r(t)$  for the corresponding three locations on the map. The solid line represents the hand path  $\bar{x}^r(t)$  reproduced by the robot, the dashed line the desired hand path  $\bar{x}^d(t)$ , with respect to the goal and the dotted line the hand path resulting from the desired joint angles, i.e.  $f(\bar{\theta}^d(t))$ . The circle represents the position of the goal.

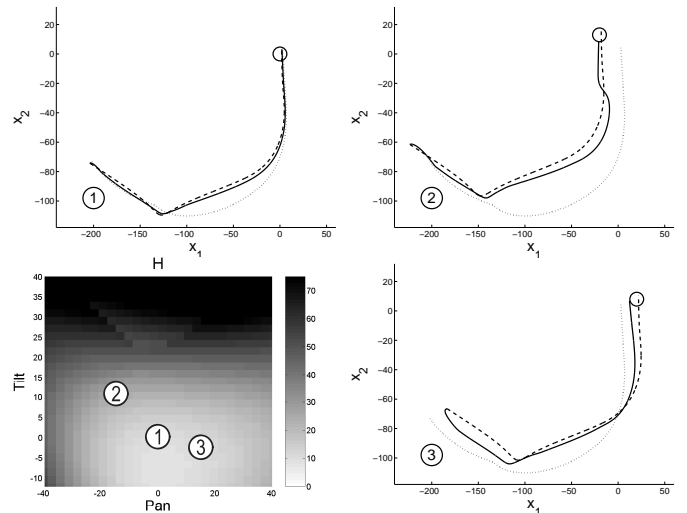


Fig. 9. *Bottom left*: Color map of the mean values taken by the cost function  $H$  for the *Sugar Task*, for various pan/tilt orientation of the head (see Fig. 8 for additional comments).



Fig. 10. Experimental results for the *Chess Task* and *Sugar Task*, when performed under different situations than the ones demonstrated (different final position on the chessboard and different head pan/tilt). The trajectory of the hand was tracked by the vision system, and superimposed to the image.



encode efficiently the task. We see that the states found automatically by the system correspond roughly to the different phases of the task (grabbing the piece of sugar, approaching the mouth, putting the piece of sugar in the mouth). At the beginning and at the end of the trajectory, since the position of the hand with respect to the sugar and to the mouth do not vary across demonstrations, the trajectories are highly constrained, which is reflected by the narrow ellipses at the beginning and at the end of the motion.

Fig. 8 and 9 show the mean values of the cost function  $H$ , when reconstructing the motion for different locations of the target (end-position of the chess piece or robot's mouth). At 3 different locations, the path followed by the robot's hand is shown (see also Fig. 10). In each case, the system finds a correct solution, that reaches correctly the desired locations for the chess piece or the mouth. However, the farther the target is from its original position, the higher the cost function and the less straight the path. This is expected and entirely acceptable, since achieving the high-level goals of the task (reaching for the targets) takes precedence over any other classical control constraint, such as running in a straight trajectory.

## V. DISCUSSION AND CONCLUSION

This paper presented a method to: 1) extract the important characteristics of a task, i.e. the spatio-temporal variations and correlations across the multivariate dataset of the task, 2) to determine a generic metric to evaluate the robot's imitation performance, and, finally, 3) to optimize the robot's reproduction of the task when placed in a new context according to the task metric. The method was validated in two experiments where a robot was taught simple manipulation tasks.

We showed that, in each case, the robot managed to adapt its motions correctly, so as to reproduce the important qualitative constraints of each task, namely grabbing and pushing the chess piece to the correct location, grabbing the piece of sugar and bringing it to its mouth. However, none of these high-level goals were explicitly represented in the robot's control system, but, were nevertheless correctly extracted by our probabilistic system.

The system we presented for solving the *what-to-imitate* and *how-to-imitate* issues is generic, in the sense that it makes no assumption on the robot's configuration (number of degrees of freedom and length of segments). As first stressed out by Nehaniv and colleagues [14], there is a multitude of correspondence problems, when trying to transfer skills across various agents and situations. In the experiments presented in this paper, the robot was being taught through kinesthetics. By showing kinesthetically how to perform a task, the user "embodies" the robot's body. Thus, this way, we simplified the correspondence problem and overlooked the problem of having different embodiments. However, by testing the system in different situations than those taught, we tackled another aspect of the correspondence problem.

Note that, in the experiments reported here, we assumed implicitly that the kinematics of joint angle trajectories and hand path is sufficient to describe the skill, and that dynamics

is of less importance. This may not be true and taking into account the forces applied on the object may certainly be very important in certain task. However, it is very likely that these are not learned through imitation, but, through more generic motor learning processes.

Further work will focus on exploiting dimensionality reduction techniques such as Principal Component Analysis to pre-process the data, and to extend the metric and associated optimization paradigm in this latent space of lower dimensionality.

## ACKNOWLEDGMENT

The work described in this paper was supported in part by the Secretariat d'Etat à l'Education et la Recherche Suisse (SER), under Contract FP6-002020, Integrated Project COGNIRON of the European Commission Division FP6-IST Future and Emerging Technologies, and was supported in part by the Swiss National Science Foundation, through grant 620-066127 of the SNF Professorships program.

## REFERENCES

- [1] C. Nehaniv and K. Dautenhahn, "Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications," in *Interdisciplinary Approaches to Robot Learning*, J. Demiris and A. Birk, Eds. World Scientific Press, 2000, vol. 24, pp. 136–161.
- [2] A. Alissandrakis, C. Nehaniv, K. Dautenhahn, and J. Saunders, "An approach for programming robots by demonstration: Generalization across different initial configurations of manipulated objects," in *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 2005, pp. 61–66.
- [3] T. Inamura, N. Kojo, T. Sonoda, K. Sakamoto, K. Okada, and M. Inaba, "Intent imitation using wearable motion capturing system with on-line teaching of task attention," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2005.
- [4] R. Zoellner, M. Pardowitz, S. Knoop, and R. Dillmann, "Towards cognitive robots: Building hierarchical task representations of manipulations from human demonstration," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005.
- [5] M. Bicego and V. Murino, "Investigating hidden markov models' capabilities in 2d shape classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 281–286, February 2004.
- [6] J. Van der Vaak, "Mixture models for clustering and dimension reduction," Ph.D. dissertation, University of Amsterdam, 2004.
- [7] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, pp. 461–464, 1978.
- [8] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77:2, pp. 257–285, February 1989.
- [9] S. Calinon and A. Billard, "Recognition and reproduction of gestures using a probabilistic framework combining PCA, ICA and HMM," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2005.
- [10] H. Sung, "Gaussian mixture regression and classification," Ph.D. dissertation, Rice University, 2004.
- [11] A. Billard, Y. Epars, S. Calinon, G. Cheng, and S. Schaal, "Discovering optimal imitation strategies," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 69–77, 2004.
- [12] S. Calinon, F. Guenter, and A. Billard, "Goal-directed imitation in a humanoid robot," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005.
- [13] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, 1977.
- [14] C. L. Nehaniv, "Nine billion correspondence problems and some methods for solving them," in *Proceedings of the International Symposium on Imitation in Animals & Artifacts*, 2003, pp. 93–95.