

# On Learning, Representing and Generalizing a Task in a Humanoid Robot

Sylvain Calinon, Florent Guenter and Aude Billard

*LASA Laboratory - Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland*

**Abstract**—We present a Programming by Demonstration (PbD) framework for generically extracting the relevant features of a given task and for addressing the problem of generalizing the acquired knowledge to different contexts. We validate the architecture through a series of experiments in which a human demonstrator teaches a humanoid robot simple manipulatory tasks. A probability based estimation of the relevance is suggested, by first projecting the motion data onto a generic latent space using Principal Component Analysis (PCA). The resulting signals were then encoded using a mixture of Gaussian/Bernoulli distributions (GMM/BMM). This provides a measure of the spatio-temporal correlations across the different modalities collected from the robot which can be used to determine a metric of the imitation performance. The trajectories are then generalized using Gaussian Mixture Regression (GMR). Finally, we analytically compute the trajectory which optimizes the imitation metric and use this to generalize the skill to different contexts.

**Index Terms**—Programming by Demonstration (PbD), Learning by Imitation, Human-Robot Interaction (HRI), human motion subspace, Gaussian Mixture Model (GMM), metric of imitation.

## I. INTRODUCTION

RECENT advances in robot *Programming by Demonstration* (PbD), also referred to as *Learning by Imitation*, have identified a number of key issues for ensuring a generic approach to the transfer of skills across various agents and situations. These have been formulated into a set of generic questions, namely *what-to-imitate*, *how-to-imitate*, *when-to-imitate* and *who-to-imitate* [1]. These questions were formulated in response to the large body of work on PbD which emphasized ad-hoc solutions to sequencing and decomposing complex tasks into known sets of actions, performable by both the demonstrator and the imitator [2], [3]. In contrast to these other works the above four questions and their solutions aim at being generic by making little or no assumption as to the type of skills which may be transmitted. Recent work on PbD addresses these questions at different levels [4]. One approach aims at extracting and encoding low-level features, e.g. primitives of motion in joint space [5]–[8], and makes only weak assumptions as to the form of the primitives or kernels used to encode the motion. By contrast, another body of work stresses the need to introduce prior knowledge as to the way information is encoded in order to achieve fast and reusable learning in the imitation of higher-level features, such as complete actions, tasks, and behaviors [9], [10]. In our work we draw on aspects from both approaches. Different demonstrations of the same task are performed and a probabilistically based estimation of relevance is used

to extract the important aspects of the task. This method provides a continuous representation of the constraints, given by a time-dependent covariance matrix, which can be used to decompose, generalize and reconstruct gestures. We then go on to formally demonstrate how such a statistical representation of motion can be combined with classical solutions to the inverse kinematics problem, in order to find a controller which optimally satisfies the constraints of the tasks and which is also adaptive to various contexts. As humanoid robots are endowed with a large number of sensors, the information contained within the dataset collected by the robot is often redundant and correlated. Through the use of linear decomposition and mixture models, our system finds a suitable representation of the data for both continuous and binary data.

Similar work has previously attempted to find optimal controllers which will reproduce a set of high-level constraints [11]. However, in this previous work, the constraint is unique during each portion of the task and is selected from a set of predefined constraints (e.g. absolute/relative constraints on position/orientation). In our work we use a similar paradigm, but introduce a more generic framework which allows for the extraction of a time-dependent continuous representation of the constraints. To illustrate the advantage of our approach, let us, for example, consider a basketball task. In this task the ball is grasped and subsequently dropped into a basket. The basket is fixed, however the position of the ball can vary from one demonstration to the next. In measuring the absolute position of the hand and its relative position to the ball, we see that a *relative position constraint* is required to grasp the ball, and that an *absolute position constraint* is required to drop the ball into the basket. In this case, reproducing either a relative or absolute constraint would not fulfill the purpose of the task. In contrast however, our model extracts a continuous representation of the constraints with local information on variations and correlations across the variables. It thus provides a localized, efficient, and generic description of the important aspects of the task.

## II. THE EXPERIMENTAL SCENARIO

We present an architecture to generically solve the problem of extracting the relevant features of a given task (*what-to-imitate* issue), the problem of evaluating how the task should be reproduced (*metric of imitation*), and the problem of finding the optimum controller with which to generalize the acquired knowledge to various contexts (part of the *how-to-imitate* issue) [1].

TABLE I  
NOTATIONS USED THROUGHOUT THIS WORK

$n$	Number of demonstrations
$T$	Number of time steps in a demonstration
$N$	Number of data points in a training set
$X$	Training set in the original data space
$\xi$	Training set in the latent space
$(d - 1)$	Spatial dimensionality in the original data space
$(D - 1)$	Spatial dimensionality in the latent space
$K$	Number of Gaussian components

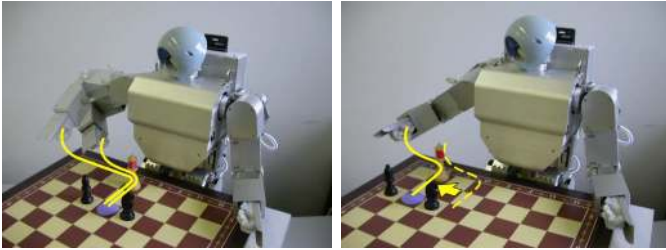


Fig. 1. *Left*: Illustration of the *what-to-imitate* issue. *Right*: Illustration of the *how-to-imitate* issue.

Fig. 1 illustrates these issues in a *Chess Task*. The task consists of grabbing the White Queen and moving it two squares forward. The picture on the left shows the path followed by the robot's hand during training when starting from two different initial locations. In order to extract the relevance of each feature of 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 will reveal weak correlations at the beginning of the motion, as there are a large set of possible paths to reach for the Queen depending on the hand's initial position. However, the analysis will measure a strong spatio-temporal correlation for grabbing the piece and pushing it towards the desired location without hitting the other pieces on the chessboard.

Fig. 1 right illustrates the *how-to-imitate* issue. Once trained to perform a task in a particular context, the robot must be able to generalize and reproduce the same task in a different context. In this example the robot must be able to grab and move the White Queen two squares forward wherever it may be on the chess board. Since the demonstrated joint angles and hand path can be mutually exclusive in the imitator space it is not possible to fulfill both constraints at the same time. Depending on the situation, the robot may have to find a very different joint angle configuration than the one demonstrated. In order to do this, the robot computes the trajectory which gives the optimal trade-off between satisfying the constraints of the task (spatio-temporal correlations across the variables) and its own body constraints.

### III. SYSTEM ARCHITECTURE

Fig. 2 gives an overview of the input-output flow through the complete model. The model is composed of the following modules: ***What-to-imitate***: The signals are encoded in a three-stage process. First, we determine the *latent space* of the motion by linearly projecting the data onto a subspace of lower dimensionality using *Principal Component Analysis* (PCA).

Second, we temporally align the signals using a *Dynamic Time Warping* (DTW) approach. Third, we determine a probabilistic representation of the data in the *latent space* by estimating the optimal *Gaussian Mixture Model* (GMM) and *Bernoulli Mixture Model* (BMM) with which to encode the motion. ***Metric of imitation***: A time-dependent similarity measure is defined by taking into account the relative importance of each variable and the dependencies across the variables using the probabilistic representation of the data. This measure evaluates the reproduction performance of a task. ***How-to-imitate***: We then compute the trajectory which optimizes the metric for a certain context, given the robot's body constraints (encapsulated in a *Jacobian* matrix), and the position of the object(s) in the scene.

Next, we describe the computations carried out in each of these modules. Section IV presents the acquisition of the data, the reduction of dimensionality, the constraints on the dataset and the temporal alignment of the signals. Section V presents the probabilistic encoding in mixture models, the criterion used to select the number of parameters and the regression process. Section VI discusses the evaluation of imitation performance and its derivation to find an optimal controller to reproduce the task.

## IV. DATA REPRESENTATION

### A. Data acquisition

The experiments were conducted using a Fujitsu HOAP-2 humanoid robot with 25 degrees of freedom (DOFs), of which only the 11 DOFs of the arms and torso were required in the experiments. The remaining DOFs of the legs were set to a constant position, so as to support the robot in an upright posture facing a table (see Fig. 1). The robot was taught through kinesthetics, i.e. by the demonstrator moving its two arms through each of the task's steps. To achieve this the robot motors were set in a passive mode, whereby each limb could be moved by the human demonstrator. The kinematics of each joint motion were recorded at a rate of 1000Hz during the demonstrations and were then re-sampled to a fixed number of points. The robot is provided with motor encoders for every DOF, except for the hands and the head actuators. Standing behind the robot and moving simultaneously its two arms is an efficient method to demonstrate a task to the robot using its own body, see Fig. 3. We will use the term *kinesthetic learning* throughout this paper to describe this data acquisition process. The robot is not provided with force sensors. However, by moving its limbs, the robot "senses" its own motion by registering the joint angle data provided by the motor encoders. The interaction when embodying the robot is more playful than using a graphical simulation and it presents the advantage for the user to implicitly feel the robot's limitations in its real-world environment.

The initial position of the different objects are registered by helping the robot to grasp and release the objects. It provides an intuitive and user-friendly means of controlling the robot without any need of specific sensory hardware to control simultaneously multiple DOFs. In other work, we have described the use of motion sensors attached to the body of

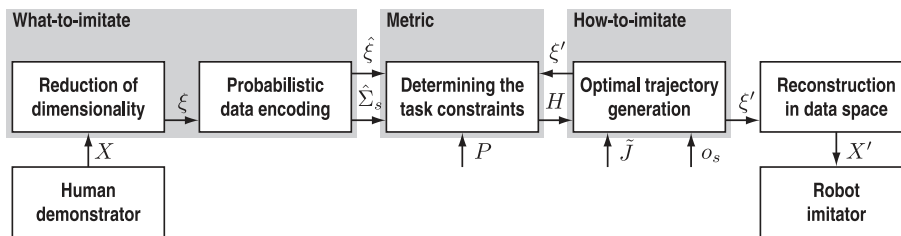


Fig. 2. Information flow across the complete system. The original demonstrated signals  $X$  are projected onto a *latent space*, with  $\xi$  the projected signals.  $\hat{\xi}$  is the generalized version of the signals with associated time-dependent covariance matrix  $\hat{\Sigma}_s$ .  $P$  is the optional time-dependent priors matrix.  $J$  is the *Jacobian* matrix defined by the architecture of the robot.  $o_s$  is the initial position of the object.  $H$  is the imitation metric.  $X'$  and  $\xi'$  are the reproduced signals in the original *data space* and *latent space*.

the demonstrator and the use of a stereoscopic vision system to track the Cartesian position of objects based on color segmentation [12]. The advantage of kinesthetic learning over these systems is that it provides a fast and accurate way of acquiring data, because it does not require to wear motion sensors/colored patches and it does not require a calibration phase. Finally, it simplifies the correspondence problem. The main disadvantage is that it remains difficult to generalize to whole body motion, because it is not possible to control simultaneously more degrees of freedom (e.g. moving the two arms and two legs of the robots simultaneously). Another disadvantage is that the demonstrated arm motions can appear less human than if they were directly produced by the user's body.

We make the assumption that the important sensory information are coming from: 1) The posture of the robot, defined by the joint angles provided by the motor encoders actuating the upper-body part. 2) The absolute positions of the hands of the robot in a Cartesian space, calculated by direct kinematics using the joint angles. 3) The relative positions between the hands and the initial position of the object, calculated from the absolute initial position of the object and the absolute positions of the hands in a Cartesian space. 4) The open/close status of the two hands of the robot.

The hand-object directional vectors and the nonlinear combination of the joint angles used to retrieve the position of the hands are hard-coded instead of being extracted autonomously by the system, which creates additional redundant information in the dataset. As we are considering manipulation tasks, we decided to include this information at hand in our system, since the importance of this information (i.e. position of the hands in a fixed system of reference) is quite straightforward and meaningful, and remains general for a broad range of tasks. This preprocessing reduces the number of examples needed to extract the important aspects of the task, since it alleviates the need of learning the direct kinematics of the robot. Note that choosing initially the adequate task-dependent variables may not be trivial for more complex paradigms. In the experiments reported here,  $n$  demonstrations of a task are produced. For each variable, the trajectory length is rescaled to a fixed value  $T$ . The total number of observations is thus  $N = n \times T$ .

Table I provides the notations used throughout this work. The training set consists of  $N$  observations of the set of variables  $\{\theta, x, y, h\}$ . Each variable consists of a 1-dimensional temporal value (time elapsed from the beginning of the

demonstration) and  $(d - 1)$ -dimensional spatial values, i.e.  $\theta = \{\theta_t, \theta_s\}$ ,  $x = \{x_t, x_s\}$ ,  $y = \{y_t, y_s\}$  and  $h = \{h_t, h_s\}$ .  $\theta_s \in \mathbb{R}^{N \times 9}$  are the joint angles of the two arms and the torso,  $x_s \in \mathbb{R}^{N \times 6}$  are the Cartesian positions of the two hands,  $y_s \in \mathbb{R}^{N \times 6}$  represent the hands-object relationships and  $h_s \in \mathbb{R}^{N \times 2}$  represent the activities of the two hands, i.e. binary signals defining the open/close status of the two hands. For each of the  $n$  demonstrations,  $y_s$  is defined by the distance vector between the initial position of the object  $o_s \in \mathbb{R}^6$  (the Cartesian position in  $\mathbb{R}^3$  is defined two times for the two hands) and the position of the hands  $x_s$ , i.e.:

$$y_{s,i,j} = x_{s,i,j} - o_{s,i} \quad \left| \begin{array}{l} \forall i \in \{1, \dots, n\} \\ \forall j \in \{1, \dots, T\} \end{array} \right. \quad (1)$$

The initial position of objects is registered by teaching kinesthetically their position. The user grabs and releases the different objects to register their position in a preliminary phase.

### B. Reduction of dimensionality

The collected data present redundancies for the majority of the tasks. However, the degree and the type of redundancies can differ from one task to another. We are looking for a latent space onto which we project the original dataset to find an optimal representation for the given task. For example, an optimal latent space for a writing task is typically represented as a projection of the 3-dimensional original Cartesian position of the hand onto a 2-dimensional latent space defined by the writing surface, while a waving motion is typically represented as a combination of a single 1-dimensional cyclic pattern. Due to the small number of examples provided, we are only considering a latent space extracted through a linear combination of the data in the original data space. It is already a difficult problem that can be resolved using various constraints.

Linear decomposition of the data can be formulated as a *Blind Source Separation* (BSS) problem.  $X_s$  is assumed to be composed of statistically independent signals. By observing  $X_s$ , the goal is to estimate the mixing matrix  $A$  and recover the original signals  $\xi_s$ . This goal can not be achieved in practice, due to the lack of general measure of statistical independence. However, other related criteria can be used to approximate the decomposition. In [8], we compared the use of *Principal Component Analysis* (PCA) and *Independent Component Analysis* (ICA) to reduce the dimensionality of human motion data. We found that ICA presented very few

advantage over PCA, i.e. decorrelation was a sufficient pre-processing step. The uncorrelatedness assumption of PCA is weaker than statistical independence, but it is still a sufficient constraint to decompose our human motion dataset. Thus, we are using PCA that finds analytically a mixing matrix  $A$  projecting the dataset  $X_s$  onto uncorrelated components  $\xi_s$ , with the criterion of preserving as much variance as possible. PCA thus assumes that important information is contained in the energy of the signals. We consider a linear transformation from the original centered *data space*  $X_s \in \{\theta_s, x_s, y_s\}$ , consisting of  $N$  data points of dimensionality  $(d-1)$ , to a *latent space*  $\xi_s \in \{\xi_s^\theta, \xi_s^x, \xi_s^y\}$ , consisting of  $N$  data points and of dimensionality  $(D-1)$ :

$$X_s - \bar{X}_s = A \xi_s \quad (2)$$

where  $\bar{X}_s \in \mathbb{R}^{N \times (d-1)}$  is a matrix containing the means of the training set  $X_s$  for each dimension, and  $A \in \{A^\theta, A^x, A^y\}$  is the transformation matrix.

We apply PCA separately to the set of variables  $X_s \in \{\theta_s, x_s, y_s\}$ , in order to identify an underlying uncorrelated representation in each dataset. Using the covariance matrix of the dataset  $\Sigma = E(X_s X_s^T) + \bar{X}_s$ , *eigenvectors*  $v_i$  and associated *eigenvalues*  $\lambda_i$  are computed, given  $\Sigma v_i = \lambda_i v_i$ ,  $\forall i \in \{1, \dots, (d-1)\}$ . By keeping the first  $(D-1)$  *eigencomponents*, we project the dataset onto their respective basis of *eigenvectors*, and obtain  $\xi_s \in \mathbb{R}^{N \times D-1}$ . The mixing matrix is then defined as  $A = \{v_1, v_2, \dots, v_{D-1}\}$ , where  $(D-1)$  is the minimal number of *eigenvectors* needed to obtain a "satisfying" representation of the original dataset, i.e. such that the projection of the data onto the reduced set of eigenvectors covers at least 98% of the data's spread:  $\sum_{i=1}^{(D-1)} \lambda_i > 0.98$ , see Fig. 5.

For each demonstration, the velocities  $\dot{X}_s$  in the *data space* are estimated as:

$$\dot{X}_{s,i,j} = X_{s,i,j} - X_{s,i,j-1} \quad \left| \begin{array}{l} \forall i \in \{1, \dots, n\} \\ \forall j \in \{2, \dots, T\} \end{array} \right. \quad (3)$$

The corresponding velocities  $\dot{\xi}_s$  in the *latent space* are estimated as:

$$\dot{\xi}_{s,i,j} = \xi_{s,i,j} - \xi_{s,i,j-1} \quad \left| \begin{array}{l} \forall i \in \{1, \dots, n\} \\ \forall j \in \{2, \dots, T\} \end{array} \right. \quad (4)$$

Using (3) and (4), we can rewrite (2) for the velocities:

$$\dot{X}_s = A \dot{\xi}_s \quad (5)$$

### C. Dataset constraints

External constraints are given to the system (see Fig. 2). The *Jacobian*  $\tilde{J}$  describes the architecture of the robot used to reproduce the task, i.e. body constraints. The initial position of the object  $o_s$  defines environmental constraints, when reproducing the task in a new situation. Relations between the different variables can be expressed with respect to these external constraints.

1) *Body constraints*:  $\dot{\theta}_s$  and  $\dot{x}_s$  are kinematically constrained, see e.g. [13]. We consider an iterative, locally linear solution to the inverse kinematics problem. Using (5), inverse kinematics can be expressed in the *latent space*:

$$\begin{aligned} \dot{x}_s &= \tilde{J}(\theta) \dot{\theta}_s \\ \Leftrightarrow A^x \dot{\xi}_s^x &= \tilde{J}(A^\theta \xi_s^\theta + \bar{\theta}_s) A^\theta \dot{\xi}_s^\theta \\ \Leftrightarrow \dot{\xi}_s^x &= J(\xi_s^\theta) \dot{\xi}_s^\theta \\ \text{with } J(\xi_s^\theta) &= (A^x)^{-1} \tilde{J}(A^\theta \xi_s^\theta + \bar{\theta}_s) A^\theta \end{aligned} \quad (6)$$

where  $(A^x)^{-1}$  is the pseudo-inverse of  $A^x$ ,  $\tilde{J}$  is the Jacobian in the original *data space*, and  $J$  is the Jacobian in the *latent space*.

2) *Environmental constraints*: Relation (1) can be expressed in terms of velocities, and re-written in the *latent space*, providing a constraint between  $\dot{\xi}_s^x$  and  $\dot{\xi}_s^y$ :

$$\begin{aligned} \dot{y}_s &= \dot{x}_s - \dot{o}_s \\ \Leftrightarrow A^y \dot{\xi}_s^y &= A^x \dot{\xi}_s^x - \dot{o}_s \\ \Leftrightarrow \dot{\xi}_s^y &= A^z \dot{\xi}_s^x - (A^y)^{-1} \dot{o}_s \\ \text{with } A^z &= (A^y)^{-1} A^x \end{aligned} \quad (7)$$

where  $\dot{o}_s$  is the initial velocity of objects (null in our experiments) and  $(A^y)^{-1}$  is the pseudo-inverse of  $A^y$ .

### D. Temporal alignment of the signals

In our previous work, *Hidden Markov Models* (HMMs) were used to encapsulate the temporal variations of the signals, previously encoded in *Gaussian Mixture Models* (GMMs) [14]. The aim of this method was to combine the temporal alignment properties of HMM with the regression capabilities of a static GMM encoding. However, the temporal information encoded in the two models was redundant, modelled as transition probabilities in the HMM, and as an additional variable in the GMM. In HMM, multivariate Gaussians are modelling local portions of the signals. By probabilistically encoding the transitions between these Gaussians, HMMs can deal with non-homogenous temporal deformation of the signals. They so act as a method to temporally align different signals, represented spatially by the Gaussians. HMM can be seen a double stochastic process, described by transition probabilities and output probabilities. Thus, retrieving a smooth trajectory from the model (generalized version of the demonstrated trajectories) is not an easy task. Previous work suggested an "averaging" approach to retrieve human motion sequences from HMM, see e.g. [15]. 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. By encoding the temporal signals directly in a mixture of Gaussians, i.e. considering the temporal component as an additional dimension, it is possible to retrieve analytically a smooth signal through regression. To do so, each time-step is considered as an input query point, and an estimation of

the output for each dimension is found by regression. The clear advantage of *Gaussian Mixture Regression* (GMR) over a stochastic retrieval process is that it provides a fast and analytic way to reconstruct the "best" sequence from a Gaussian model.

In the work presented here, temporal alignment of the signals is performed by a pattern-based approach used as a pre-processing step. It provides a more coherent framework, allowing temporal distortion between different examples and providing a simple and unique description of the sequential information contained in the data. *Dynamic Time Warping* (DTW) is used as a template matching pre-processing step to temporally align the signals, see e.g. [16]. Being much simpler, training of the DTW carries a small computational cost benefit. The drawback is that recognition under this system requires a large number of distance computations.

Regression using GMMs offers a way of extracting, from a data set, a single generalized signal made up from the set of signals used to train the model. The signal is not part of the dataset, but instead encapsulates all of its essential features. *Gaussian Mixture Regression* (GMR) treats the temporal information and spatial components of the signal indifferently in the fitting of multivariate Gaussians. This method can then perform poorly when the signal has strong non-homogeneous time distortions. We thus explore the use of DTW as a pre-processing step to improve the quality of the regression, after encoding the aligned signals with GMM. DTW is sometimes considered as a weaker method than HMM, but it does have the advantage of being simple and robust, and can be used conjointly with GMR. DTW finds a non-linear alignment which minimizes the error between the signals and reference signal. A distance table is first built and a DTW path is searched by a dynamic programming approach through the table, with slope limits to prevent degenerate warps. Here, to improve the computational efficiency, the alignment is performed in the *latent space*, which is usually of lower dimensionality than the original *data space*.

Let us consider two multivariate signals  $\xi_s^A$  and  $\xi_s^B$  of length  $T$ . We define the distance measure between two datapoints of temporal index  $k_1$  and  $k_2$  by  $h(k_1, k_2) = \|\xi_{s, k_1}^A - \xi_{s, k_2}^B\|$ . A warping path  $S = \{s_l\}_{l=1}^L$  is defined by  $L$  elements  $s_l = \{k_1, k_2\}$ . The warping path is subject to several constraints. *Boundary conditions* are given by  $s_1 = \{1, 1\}$  and  $s_K = \{T, T\}$ . If  $s_k = \{a, b\}$  and  $s_{k-1} = \{a', b'\}$ , *monotonicity* is given by  $a \geq a'$  and  $b \geq b'$ , while *continuity* is defined by  $a - a' \leq 1$  and  $b - b' \leq 1$ . Dynamic programming is used to minimize  $\sum_{l=1}^L s_l$ , by evaluating iteratively:

$$\gamma(k_1, k_2) = h(k_1, k_2) + \min\{\gamma(k_1-1, k_2-1), \gamma(k_1-1, k_2), \gamma(k_1, k_2-1)\}$$

Global and local constraints are defined so as to reduce the computational cost of the algorithm, and to limit the permissible warping paths. Here, we experimentally fixed an *adjustment window condition* and a *slope constraint condition*, defining the maximum amount of warping allowed, as in [17].

## V. MIXTURE MODELS

A probabilistic representation of the projected, temporally aligned data  $\xi_j = \{\xi_{t,j}, \xi_{s,j}\}$  with  $\xi_{t,j} = X_{t,j}$  is used to estimate the variations and correlations across the variables, allowing a localized characterization of the different parts of the gesture. Mixture modelling is a popular approach for density approximation of continuous or binary data, see e.g. [18]. It allows for flexibility by looking at an appropriate trade-off between model complexity and variations of the available training data. A mixture model of  $K$  components is defined by a probability density function:

$$p(\xi_j) = \sum_{k=1}^K p(k) p(\xi_j|k) \quad (8)$$

where  $\xi_j$  is a datapoint,  $p(k)$  is the prior and  $p(\xi_j|k)$  the conditional probability density function.

### A. Gaussian Mixture Model (GMM)

Let us consider a dataset in the *latent space*  $\xi_j = \{\xi_{t,j}, \xi_{s,j}\}_{j=1}^N$ . The dataset consists of  $N$  datapoints of dimensionality  $D$ , taking into account the temporal information. The dataset can be either joint angles, hand paths, or hands-object distance vectors. The dataset is modelled by a mixture of  $K$  Gaussians of dimensionality  $D$ . The parameters in (8) become:

$$\begin{aligned} p(k) &= \pi_k \\ p(\xi_j|k) &= \mathcal{N}(\xi_j; \mu_k, \Sigma_k) \\ &= \frac{1}{\sqrt{(2\pi)^D |\Sigma_k|}} e^{-\frac{1}{2}((\xi_j - \mu_k)^T \Sigma_k^{-1} (\xi_j - \mu_k))} \end{aligned} \quad (9)$$

where  $\{\pi_k, \mu_k, \Sigma_k\}$  are the parameters of the Gaussian component  $k$ , defining respectively the prior, mean and covariance matrix. *Maximum Likelihood Estimation* of the mixture parameters is performed iteratively using the standard *Expectation-Maximization* (EM) algorithm [19]. EM is a simple local search technique that guarantees monotone increase of the likelihood of the training set during optimization. The algorithm requires an initial estimate, and to avoid getting trapped into a poor local minima a rough *k-means* clustering technique is first applied to the data. The Gaussian parameters are then derived from the clusters found by *k-means*.

### B. Bernoulli Mixture Model (BMM)

To encode a set of  $(D-1)$ -dimensional binary datapoint  $h_{s,j} = \{h_{s,j,i}\}_{i=1}^{D-1}$  in a probabilistic framework, a *mixture of multivariate Bernoulli distributions* is used in a similar fashion to the use of a *mixture of multivariate Gaussian distributions* for continuous data, see e.g. [20]. The dependencies across the binary data are captured thanks to the contribution of the different components of the mixture.

A mixture of  $(D-1)$ -dimensional Bernoulli density function component  $k$ , of parameters or *prototype*  $p_k = \{p_{k,i}\}_{i=1}^{D-1}$ , with  $p_{k,i} \in [0, 1]$ , is defined by (8) and:

$$\begin{aligned} p(k) &= \pi_k \\ p(h_{s,j}|k) &= \mathcal{B}(h_{s,j}; p_k) = \prod_{i=1}^{D-1} (p_{k,i})^{h_{s,j,i}} (1 - p_{k,i})^{1-h_{s,j,i}} \end{aligned}$$

Similarly to GMM, parameters  $\{\pi_k, p_k\}$  are estimated using the EM-algorithm.

### C. Model selection

A drawback of EM is that the optimal number of components  $K$  in a model may not be known beforehand. A common method consists of estimating multiple models with increasing number of components, and selecting an optimum based on some model selection criterion, see e.g. [21].

We therefore need to arrive at a trade-off between optimizing the model's likelihood (a measure of how well the model fits the data) and minimizing the number of parameters needed to encode the data. Different criteria have been proposed: Cross validation, *Akaike Information Criterion* (AIC), *Bayesian Information Criterion* (BIC) or *Minimum Description Length* (MDL) are commonly found in the literature. Cross validation has the disadvantage to require additional demonstrations to form a test set. We selected the *Bayesian Information Criterion* (BIC) [22] that provided the most satisfying results with our dataset. Multiple GMMs/BMMs are then estimated and the BIC score is used to select the optimal number of GMM/BMM components  $K$ :

$$S_{\text{BIC}} = -\mathcal{L} + \frac{n_p}{2} \log(N)$$

where  $\mathcal{L} = \sum_{j=1}^N \log(p(\xi_j))$  is the log-likelihood of the model using the demonstrations as testing set,  $n_p$  is the number of free parameters required for a mixture of  $K$  components, i.e.  $n_p = (K - 1) + KD$  for a Bernoulli Mixture Model and  $n_p = (K - 1) + K(D + \frac{1}{2}D(D + 1))$  for a Gaussian Mixture Model with full covariance matrix.  $N$  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 which aims to minimize the number of parameters. In our experiments we compute a set of candidate GMMs/BMMs with up to 10 components and keep the model with the minimum score, see Fig. 6. Several approaches exist to optimize the model selection paradigm and the computational efficiency of the algorithms, but they are not discussed here.

### D. Gaussian Mixture Regression (GMR)

To reconstruct a general form for the signals we apply *Gaussian Mixture Regression* (GMR) [23]. Consecutive temporal values  $\xi_t$  are used as query points and the corresponding spatial values  $\hat{\xi}_s$  are estimated through regression. For each GMM, the temporal and spatial components (input and output parameters) are separated, i.e. the mean and covariance matrix of the Gaussian component  $k$  are defined by:

$$\mu_k = \{\mu_{t,k}, \mu_{s,k}\} \quad , \quad \Sigma_k = \begin{pmatrix} \Sigma_{t,k} & \Sigma_{ts,k} \\ \Sigma_{st,k} & \Sigma_{s,k} \end{pmatrix}$$

For each Gaussian component  $k$ , the conditional expectation of  $\xi_{s,k}$  given  $\xi_t$ , and the estimated conditional covariance of  $\xi_{s,k}$  given  $\xi_t$  are:

$$\begin{aligned} \hat{\xi}_{s,k} &= \mu_{s,k} + \Sigma_{st,k}(\Sigma_{t,k})^{-1}(\xi_t - \mu_{t,k}) \\ \hat{\Sigma}_{s,k} &= \Sigma_{s,k} - \Sigma_{st,k}(\Sigma_{t,k})^{-1}\Sigma_{ts,k} \end{aligned} \quad (10)$$

$\hat{\xi}_{s,k}$  and  $\hat{\Sigma}_{s,k}$  are mixed according to the probability that the Gaussian component  $k \in \{1, \dots, K\}$  has of being responsible for  $\xi_t$ :

$$\beta_k = \frac{p(\xi_t|k)}{\sum_{i=1}^K p(\xi_t|i)} \quad (11)$$

Using (10) and (11), for a mixture of  $K$  components, the condition expectation of  $\xi_s$  given  $\xi_t$ , and the conditional covariance of  $\xi_s$  given  $\xi_t$  are:

$$\hat{\xi}_s = \sum_{k=1}^K \beta_k \hat{\xi}_{s,k} \quad , \quad \hat{\Sigma}_s = \sum_{k=1}^K \beta_k^2 \hat{\Sigma}_{s,k}$$

Thus, by evaluating  $\{\hat{\xi}_s, \hat{\Sigma}_s\}$  at different time steps  $\xi_t$ , a generalized form of the motions  $\hat{\xi} = \{\xi_t, \hat{\xi}_s\}$  and associated covariance matrix are produced. The temporal interval between two time steps can be directly related to the controller requirements of the robot. Note that it is not equivalent to taking the mean and variance of the data at each time step, which would produce jerky trajectories and increase dramatically the amount of parameters (the mean and variance values would be kept in memory for each time step). With a probabilistic model, only the means and covariance matrices of the Gaussians are kept in memory.

## VI. EVALUATION OF IMITATION PERFORMANCE

### A. Weighted similarity measure

To measure the similarity between a candidate position  $\xi_s$  and a desired position  $\hat{\xi}_s$ , both of dimensionality  $(D - 1)$ , a weighted Euclidean distance measure can be defined as:

$$\sum_{i=1}^{D-1} w_i (\xi_{s,i} - \hat{\xi}_{s,i})^2 = (\xi_s - \hat{\xi}_s)^T W (\xi_s - \hat{\xi}_s)$$

where  $W$  is a  $(D - 1) \times (D - 1)$  time-dependent matrix.  $W$  can be used as a diagonal matrix, with weights  $w_i$  in the diagonal defining the importance of the different variables. In the most general case, a full covariance matrix is used to account for the correlations across the different variables.

### B. Metric of imitation

We proposed in [24] a general formalism for evaluating the reproduction of a task. The generic similarity measure  $H$  takes into account the variations of constraints and the dependencies across the variables which have been learned over time. The metric is continuous, positive, and can be estimated at any point along the trajectory.

In the latent space, let  $\{\hat{\xi}_s^\theta, \hat{\xi}_s^x, \hat{\xi}_s^y\}$  be, respectively, the generalized joint angle trajectories, the generalized hand path and the generalized hands-object distance vectors extracted from the demonstrations. Let  $\{\xi_s^\theta, \xi_s^x, \xi_s^y\}$  be the candidate trajectories for reproducing the motion. The metric of imitation performance (i.e. cost function for the task)  $H$  is given by:

$$\begin{aligned} H &= (\xi_s^\theta - \hat{\xi}_s^\theta)^T W^\theta (\xi_s^\theta - \hat{\xi}_s^\theta) \\ &+ (\xi_s^x - \hat{\xi}_s^x)^T W^x (\xi_s^x - \hat{\xi}_s^x) \\ &+ (\xi_s^y - \hat{\xi}_s^y)^T W^y (\xi_s^y - \hat{\xi}_s^y) \end{aligned} \quad (12)$$

We consider the additional variables  $c_1$ ,  $c_2$  and  $c_3$  defined by:

$$\begin{aligned} c_{1,i,j} &= \hat{\xi}_{s,i,j}^\theta - \xi_{s,i,j-1}^\theta & \forall i \in \{1, \dots, n\} \\ c_{2,i,j} &= \hat{\xi}_{s,i,j}^x - \xi_{s,i,j-1}^x & \forall j \in \{2, \dots, T\} \\ c_{3,i,j} &= \hat{\xi}_{s,i,j}^y - \xi_{s,i,j-1}^y \end{aligned}$$

and rewrite (12) as:

$$\begin{aligned} H &= (\dot{\xi}_s^\theta - c_1)^T W^\theta (\dot{\xi}_s^\theta - c_1) \\ &+ (\dot{\xi}_s^x - c_2)^T W^x (\dot{\xi}_s^x - c_2) \\ &+ (\dot{\xi}_s^y - c_3)^T W^y (\dot{\xi}_s^y - c_3) \end{aligned} \quad (13)$$

### C. Finding an optimal controller

The problem is now reduced to finding a minimum of (13), when subjected to the body constraint (6) and environmental constraint (7). Since  $H$  is a quadratic function, the problem can be solved analytically by *Lagrange* optimization. We define the *Lagrangian* as:

$$\begin{aligned} L(\dot{\xi}_s^\theta, \dot{\xi}_s^x, \dot{\xi}_s^y, \lambda_1, \lambda_2) &= H + \lambda_1^T (\dot{\xi}_s^x - J \dot{\xi}_s^\theta) \\ &+ \lambda_2^T (\dot{\xi}_s^y - A^z \dot{\xi}_s^x + (A^y)^{-1} \dot{o}_s) \end{aligned}$$

where  $\lambda_1$  and  $\lambda_2$  are the vectors of associated *Lagrange* multipliers. To solve  $\nabla L = 0$ , we consider symmetric matrix  $W^T = W$  and derive respectively  $\frac{\partial L}{\partial \xi_s^\theta} = 0$ ,  $\frac{\partial L}{\partial \xi_s^x} = 0$  and  $\frac{\partial L}{\partial \xi_s^y} = 0$ :

$$-2 W^\theta (\dot{\xi}_s^\theta - c_1) - J^T \lambda_1 = 0 \quad (14)$$

$$-2 W^x (\dot{\xi}_s^x - c_2) + \lambda_1 - (A^z)^T \lambda_2 = 0 \quad (15)$$

$$-2 W^y (\dot{\xi}_s^y - c_3) + \lambda_2 = 0 \quad (16)$$

Using (15) and (16), we find:

$$\lambda_1 = 2 W^x (\dot{\xi}_s^x - c_2) + (A^z)^T 2 W^y (\dot{\xi}_s^y - c_3) \quad (17)$$

Using (17) and (14), we find:

$$\begin{aligned} W^\theta (\dot{\xi}_s^\theta - c_1) + J^T W^x (\dot{\xi}_s^x - c_2) + \\ J^T (A^z)^T W^y (\dot{\xi}_s^y - c_3) = 0 \end{aligned}$$

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

$$\begin{aligned} \dot{\xi}_s^\theta &= (W^\theta + J^T W^x J + (A^z J)^T W^y (A^z J))^{-1} \\ &\times (W^\theta c_1 + J^T W^x c_2 + (A^z J)^T W^y c_4) \end{aligned}$$

$$\text{with } c_4 = (A^y)^{-1} \dot{o}_s + c_3$$

We can then compute iteratively the joint angle trajectories with:

$$\xi_{s,i,j}^\theta = \xi_{s,i,j-1}^\theta + \dot{\xi}_{s,i,j}^\theta \quad \left| \begin{array}{l} \forall i \in \{1, \dots, n\} \\ \forall j \in \{2, \dots, T\} \end{array} \right.$$

The joint angle trajectories are finally found using the relation  $\theta_s = A^\theta \xi_s^\theta + \hat{\theta}_s$ .

In our model,  $W \in \{W^\theta, W^x, W^y\}$  are weighting matrices which represent the time-varying constraints during the task. The statistical variations and relations across the different variables  $\hat{\Sigma}_s \in \{\hat{\Sigma}_s^\theta, \hat{\Sigma}_s^x, \hat{\Sigma}_s^y\}$  serve as a basis to represent the constraints, i.e.:

$$W = (\hat{\Sigma}_s)^{-1}$$

Note that the current framework can be extended to other loss functions, as long as these can be continuously differentiable along the state variables.

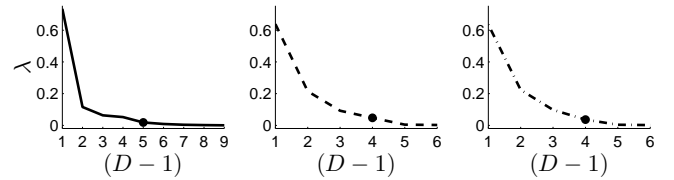


Fig. 5. Estimation of the number of components required to reduce the dimensionality of the data space for the *Bucket Task*, using *eigenvalues* (solid line for the joint angles dataset, dashed line for the hands paths dataset and dash-dotted line for the hands-object distance vectors dataset). The point corresponds to the number of dimensions retained to represent at least 98% of the data variance.

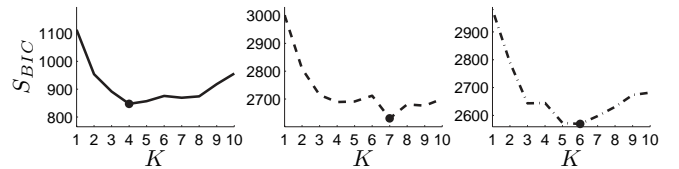


Fig. 6. Estimation of the number of Gaussian components required to model the trajectories in the latent space for the *Bucket Task* (solid line for the joint angles dataset, dashed line for the hands paths dataset and dash-dotted line for the hands-object relationships dataset). Adding more Gaussian components increases the log-likelihood, but also increases the number of parameters. The BIC criterion defines a trade-off to select an optimal number of parameters.

## VII. EXPERIMENTAL RESULTS

We conducted three experiments to demonstrate the validity of our model for teaching a humanoid robot simple manipulatory tasks, see Fig. 3. Control affected only the eight DOFs of the arms, the one DOF of the torso, and the two binary commands to open and close the robot's hands. The robot was shown the task 4 to 7 times by an expert user. Note that the number of examples required for an efficient reproduction of the task depends on the teaching efficiency of the user: an expert teacher produces demonstrations that are exploring as much as possible the variations allowed by the task, while a naive user can demonstrate the task several times in the same manner without fully exploiting the constraints required by the task. Once trained, the robot was required to reproduce each task under different constraints, by placing the object 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.

Fig. 4 shows an encoding example for the *Bucket Task*. By observing the continuous description of the variations along the trajectories, we see that the object-hands distance vectors are highly constrained at time steps 30-50, when grabbing and holding the bucket (i.e. relative constraint). The hands paths are also highly constrained at the end of the motion. The essential features of the task have thus been extracted

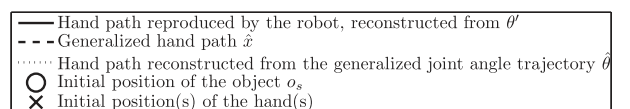


Fig. 7. Legend for Fig. 8-13.

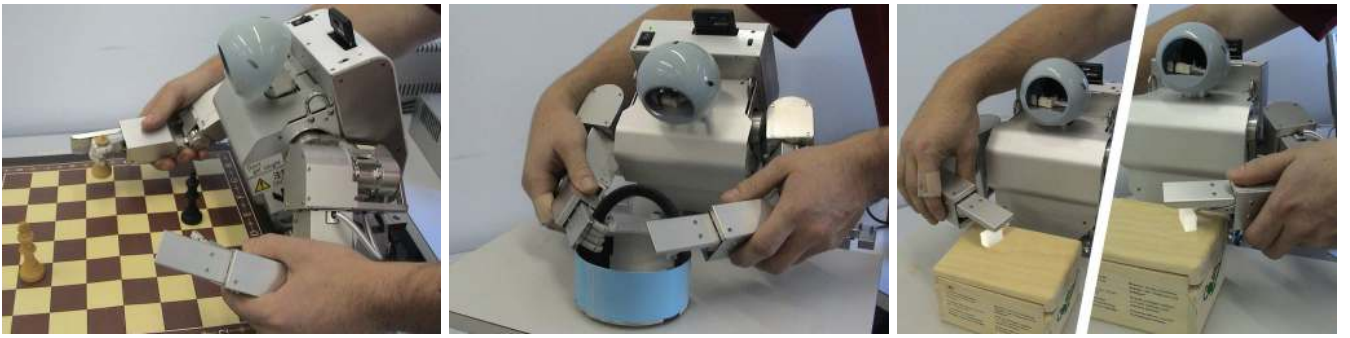


Fig. 3. Teaching through kinesthetics for the 3 experiments conducted. *Chess Task*: Grabbing and moving a chess piece two squares forward. *Bucket Task*: Grabbing and bringing a bucket to a specific position. *Sugar Task*: Grabbing a piece of sugar and bringing it to the mouth, using either the right or left hand.

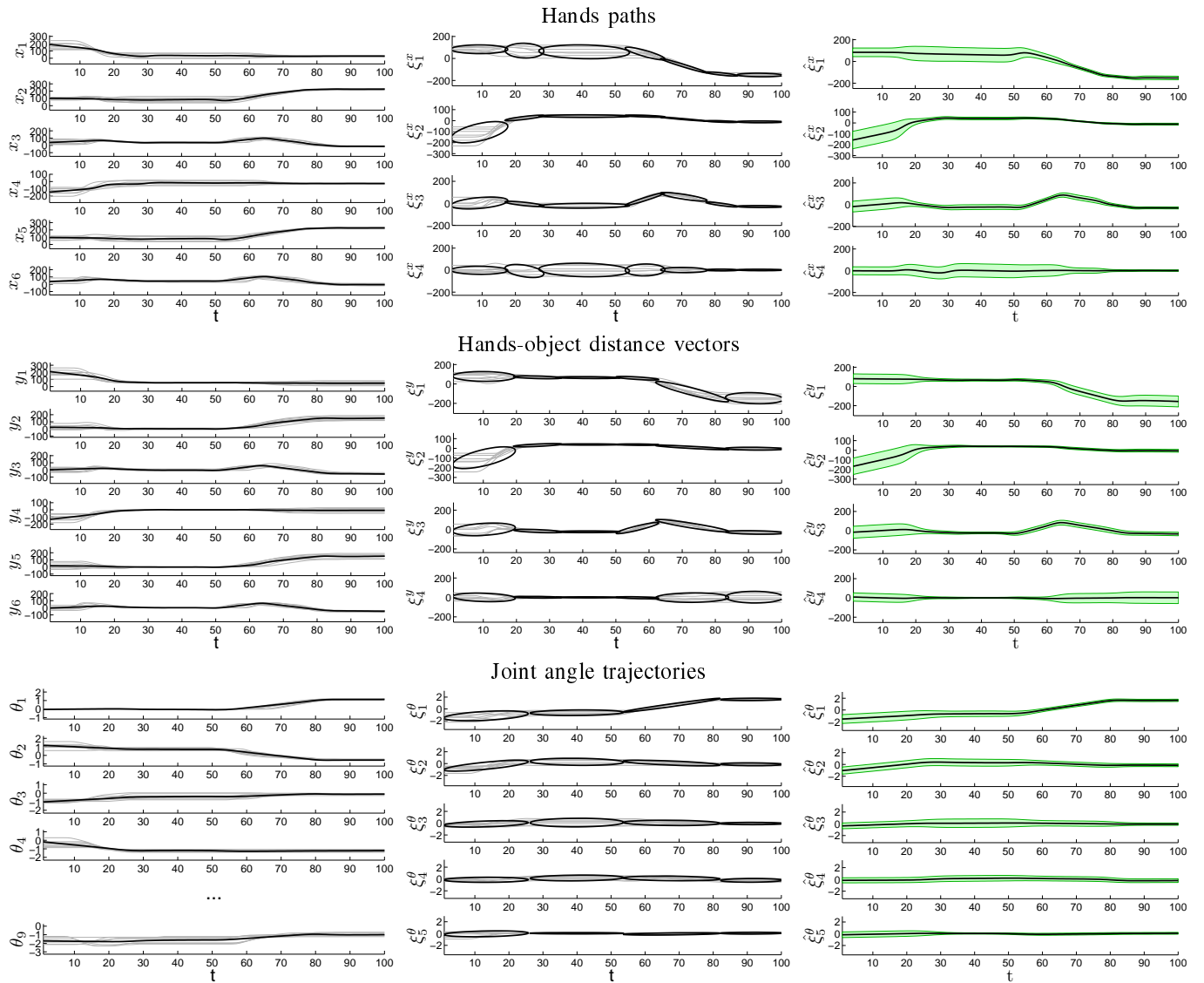


Fig. 4. Probabilistic encoding for the *Bucket Task*. *Left column*: Generalization of the hands paths  $x$  ( $\{x_1, x_2, x_3\}$  and  $\{x_4, x_5, x_6\}$  represent respectively the right/left hand paths), hands-object relationships  $y$  ( $\{y_1, y_2, y_3\}$  and  $\{y_4, y_5, y_6\}$  represent respectively the relationships for the right/left hand) and joint angle trajectories  $\theta$  ( $\theta_1$  represents the torso joint angle,  $\{\theta_2, \dots, \theta_5\}$  and  $\{\theta_6, \dots, \theta_9\}$  represent respectively the right/left arm joint angles). The demonstrations are represented in grey lines and the generalized signal reconstructed from the latent space is represented in bold line. *Middle column*: Reduction of dimensionality and temporal alignment. The signals  $X$  are projected onto a latent space of lower dimensionality, and processed by DTW. The resulting signals  $\xi$  are encoded in GMM, whose covariance matrix is represented by ellipses. *Right column*: Extraction of the constraints. The generalized version of the signals in the latent space  $\hat{\xi}$  is represented in bold line, with the corresponding covariance information  $\hat{\Sigma}_s$  represented as an envelope around  $\hat{\xi}$ . The first clear observation is that the hands-object relationships are highly constrained when the user is grabbing the object at time steps 30-50, i.e. the generalized signal presents a narrow envelope for each dimension. The second observation is that the generalized hands paths are highly constrained at the end of the motion, since the ending-positions vary very little across the demonstrations (the bucket is always placed at a specific location after being grabbed).



TABLE II  
NUMBER OF PARAMETERS FOUND AUTOMATICALLY BY THE SYSTEM (SEE TABLE I FOR THE NOTATIONS).

		Data space			Latent space	
		$n$	$T$	$(d-1)$	$(D-1)$	$K$
<i>Chess Task</i>	$\theta$	7	100	9	4	5
	$x$	7	100	6	4	5
	$y$	7	100	6	4	4
	$h$	7	100	2	(2)	1
<i>Bucket Task</i>	$\theta$	7	100	9	5	4
	$x$	7	100	6	4	7
	$y$	7	100	6	4	6
	$h$	7	100	2	(2)	1
<i>Sugar Task - left</i>	$\theta$	4	100	9	2	5
	$x$	4	100	6	3	5
	$y$	4	100	6	3	6
	$h$	4	100	2	(2)	1
<i>Sugar Task - right</i>	$\theta$	4	100	9	2	6
	$x$	4	100	6	3	6
	$y$	4	100	6	3	6
	$h$	4	100	2	(2)	1

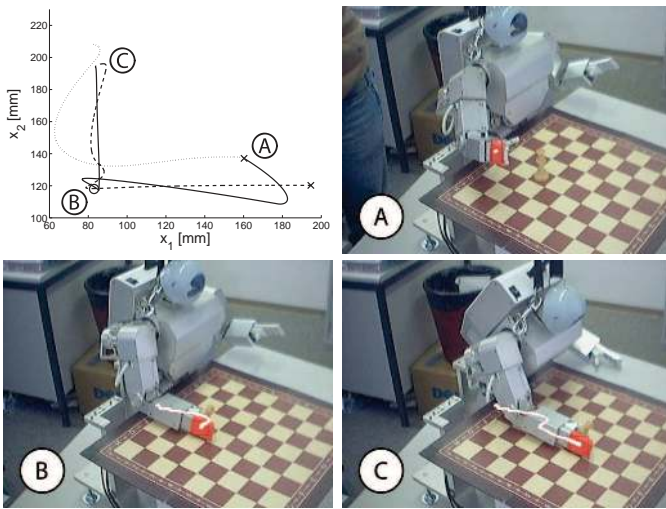


Fig. 8. Decomposition of the *Chess Task*, when reproducing the task with an initial position of the object which is close to the generalized trajectories. The hands paths have been tracked by a stereoscopic vision system.

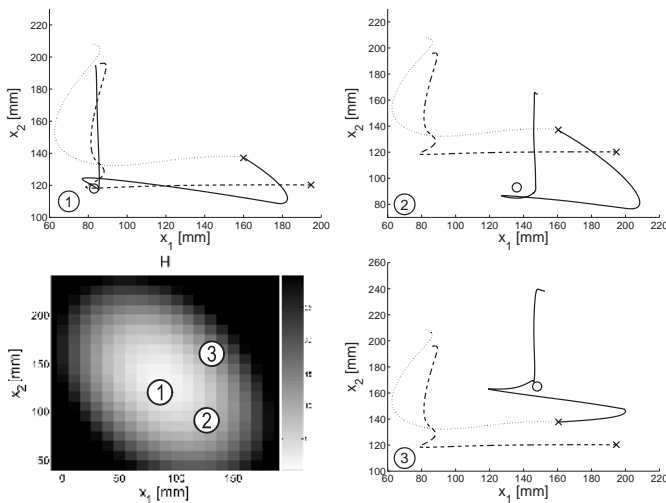


Fig. 9. *Bottom left*: Mean values taken by the cost function  $H$  for the *Chess Task*, with a varying initial position of the chess piece on the board. 1,2,3: Reproduction for the corresponding three locations on the map.

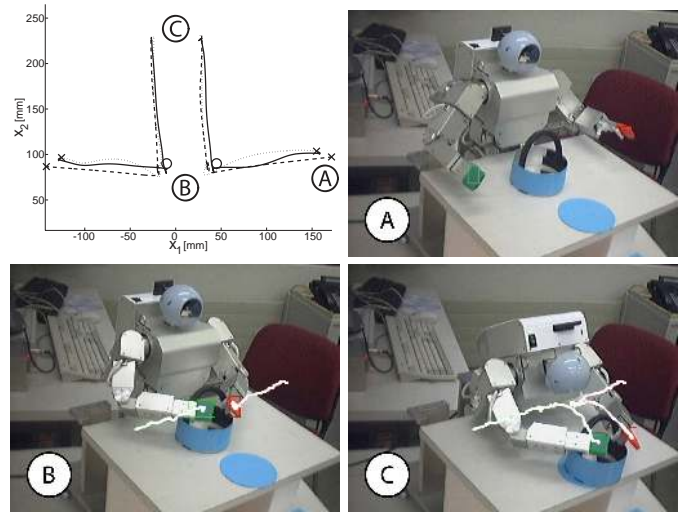


Fig. 10. Decomposition of the *Bucket Task*, when reproducing the task with an initial position of the object close to the generalized trajectories.

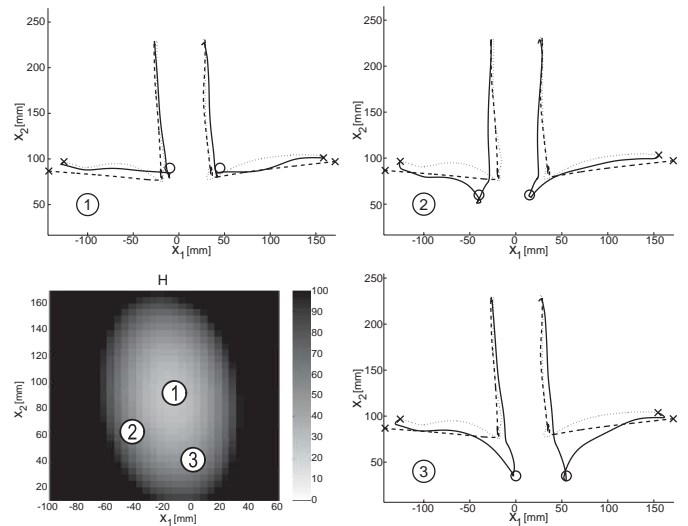


Fig. 11. *Bottom left*: Mean values taken by the cost function  $H$  for the *Bucket Task* with a varying initial position of the bucket on the table. 1,2,3: Reproduction for the corresponding three locations on the map.

successfully in a continuous representation. The joint angle trajectories can present some redundancies with the hands paths but they still provide useful additional information. As the robot's arms permit to position the hands in the space with different joint configurations, an inverse kinematics problem arises when hands paths must be reproduced. Adding the constraint of matching the joint angles solution demonstrated by the user to the inverse kinematics solution produces results that are looking more natural. This process is also highly relevant when considering different embodiments, e.g. different segment lengths. In this condition, it is important to consider what are the features that should be reproduced, i.e. joint angles or hands paths, because they are not consistent for the different embodiments.

Fig. 5 and 6 provide an example for the automatic selection of the number of PCA/GMM components for the *Bucket Task*. Results for the different tasks are provided in Table II. We see

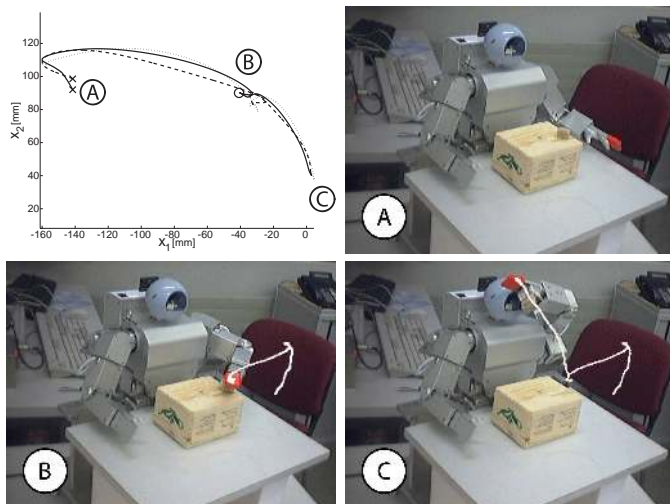


Fig. 12. Decomposition of the *Sugar Task* for the left hand, when reproducing the task with an initial position of the object close to the generalized trajectories.

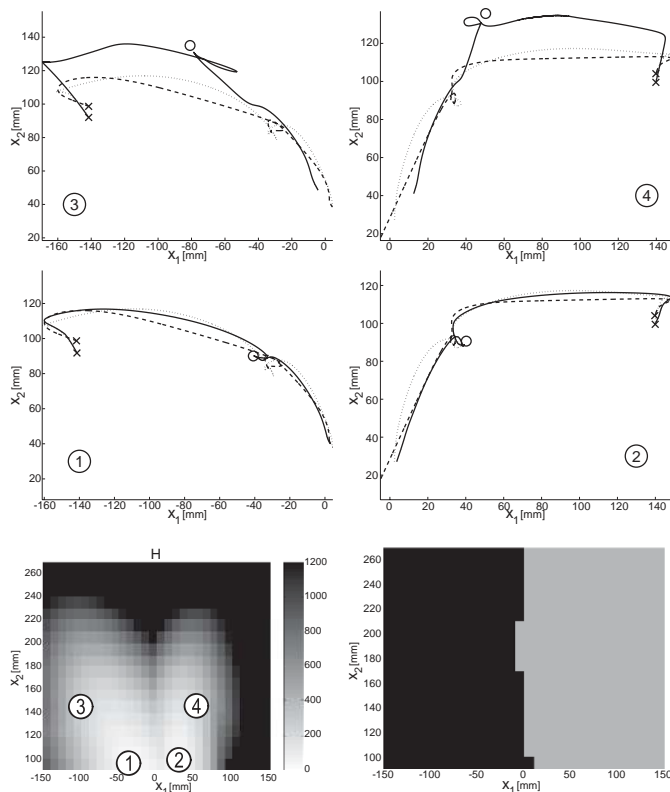


Fig. 13. *Bottom left*: Mean values taken by the cost function  $H$  for the *Sugar Task*, with a variation in the initial position of the piece of sugar on the box. We distinguish two different areas for the left and right arm. The difference in size is due to the different variations used for the left and right part. *Bottom right*: Selection of a left/right arm controller depending on the value of  $H$  (black areas correspond to  $H_{\text{left}} < H_{\text{right}}$ ). 1,2,3,4: Reproduction for the corresponding four locations on the map.

that the dimensionality of the latent space for the *Chess Task* and *Bucket Task* are higher than for the *Sugar Task*, probably due to the facts that both hands are used simultaneously in these cases (for the *Chess Task*, the left hand is used to bend over the table). For the *Sugar Task*, one hand is usually motionless while the other is performing the task. The number of Gaussian components are between 4 and 7. The number of components for the binary signals  $h$  is always 1 (one hand or two hands closed simultaneously).

The total number of parameters depends quadratically on the dimensionality of the latent space and linearly on the number of Gaussian components, i.e. the total number of parameters used to encode the data are  $n_{\text{PCA}} = (D - 1)(d - 1)$  and  $n_{\text{GMM}} = (K - 1) + K(D + \frac{1}{2}D(D + 1))$ .

Fig. 8-13 are using the same legend as that presented in Fig. 7. Fig. 8 and 9 show the reproduced trajectories for the *Chess Task*, depending on the initial position of the chess piece. Knowing that the right shoulder position is  $\{x_1, x_2\} = \{100, -35\}$ , we see on the map of Fig. 9 that the best location to reproduce the motion is to initially place the chess piece in front of the right arm, see inset (1). In inset (2) and (3), the chess piece is placed initially at different positions unobserved during the demonstrations.

Fig. 10 and 11 show the reproduced trajectories for the *Bucket Task*, depending on the initial position of the bucket. The optimal trajectory which satisfies the learned constraints follows globally the demonstrated hands paths, still using the demonstrated object-hands trajectories when approaching the bucket.

Fig. 12 and 13 show the reproduced trajectories for the *Sugar Task*, depending on the initial position of the piece of sugar. In this task, a box is centered in front of the robot and two different gestures are taught to the robot. Firstly the robot is taught how to grasp with its right hand a piece of sugar located at the far right on the top of the box. Then, it is taught how to grasp with its left hand a piece of sugar located at the far left on the top of the box. We compute an optimal controller for both the left and right arms, evaluate each controller with its respective metric, and select the best controller to reproduce the task. We see in the bottom-right inset of Fig. 13 that the limit between the left/right part is clearly situated at  $x_1 = 0$  (i.e. on the symmetry axis of the robot). Insets (3) and (4) of Fig. 13 correspond to an initial position of the piece of sugar which differs from the initial positions used during the demonstrations.

## VIII. DISCUSSION

The imitation metric in our experiments is used to find a solution that tries to match as best as possible the object-hands relationships, the hands paths and the joint angle trajectories used to produce these hands paths. Depending on the task, these variables have different importance, and the different levels of relevance are extracted by observation of the task produced by a human expert. In a goal-directed framework, these three variables have also different levels of relevance. If an object is manipulated, the first variable shows the highest importance. If there is no object in the scene and the hands

paths follow an invariant pattern, the second variable dominates. Reproducing the exact gesture is often less important for manipulation tasks but can become highly relevant for motion such as waving the hand, dancing, or knocking on a door. Bekkering and colleagues have set up experiments to show that imitation is goal-directed, using several gestures to reproduce during an imitation game [25]. They suggested that the hands paths and hands-object relationships have different levels of importance, following a hierarchy of relevance. They also suggested that the use of the different levels mainly depends on the working memory capacities of the infants/adults. While infants focus on a single level, adults use multiple levels simultaneously with preferences for the levels of highest relevance in the hierarchy. In previous work [12], we used a similar paradigm by weighting the different levels of variables, and showed how these priors can speed up the extraction of the task constraints.

The presented system can deal with two types of generalizations: 1) By projecting the original data onto a latent space and by encoding the resulting data in Gaussian Mixture Models and Bernoulli Mixture Models, the system can generalize over the variations in joint angles, hands paths, hands-object relationships, and signals commanding the opening and closing of the hands. 2) By extracting the variation and correlation information and using this information to find a solution to the inverse kinematics, the system is also able to generalize over different situations, i.e. over different initial position of the object. Note that the range of generalization permitted for the initial position depends directly on the dimensionality of the latent space obtained by the system. For example, if the system detects that the hands are constrained to move an object in a plane (2-dimensional hands paths latent space), the system will not be able to generalize over initial position of the object that is not in this plane. For small changes in the initial position of the object between the demonstrations and the reproduction the robot managed to correctly adapt its motions. It reproduced the important qualitative features of each task, namely grabbing and moving the chess piece with a specific relative path, grabbing the bucket with two hands and moving it to a specific location or grabbing the piece of sugar and bringing it to its mouth with either left or right arm. None of these high-level goals were explicitly represented in the robot's control system, nevertheless they were correctly extracted by our probabilistic system.

In the experiments reported here, we implicitly assumed that kinematics information was sufficient to describe the task and that dynamics information was less important. This may not always be true and certainly in some tasks the forces applied to the object would be very important, see e.g [26]. However, it is very likely that these processes are not learned through imitation but rather through more generic motor learning processes. The proposed system is open-loop and is aimed at providing a solution to the reproduction of a task, which is a generalization of the demonstrations produced. It restrains the search space of the possible solutions that the robot can use to achieve a task, and can be used co-jointly with other systems to refine the solution. Our current work investigate the use of this system to provide a solution to the reproduction of

a task, which is then used in a dynamical system achieving stable solutions in case of perturbations [27].

In [28] and [29], we investigated the use of social cues to extract information about the relevant features in the task, as well as to guide the teaching scenario through communicative gesture or speech. These social cues are aimed at understanding the intent of the user and can be used and combined differently, depending on the demonstrator's and imitator's personalities. For humans, the combination of these cues involves complex emotional and cultural interaction aspects. Some of these hints are subtle and can involve misunderstanding (e.g. facial expression), while others are more explicit (e.g. speech). Depending on who they are interacting with, humans naturally select the appropriate means by which to transfer their intention (e.g. use of gestures with deaf people). A humanoid robot also has its own particular sensory and working memory capabilities. Actual humanoid robot sensors are still not able to capture subtle social cues in the way humans do. Thus, we suggest that one of the most robust and appropriate ways of transferring information to the robot is by using statistics and associated machine learning algorithms. Learning of relationships and invariance across a set of sensory data can be carried out quite efficiently by the robot. Compared to humans, keeping track of a large amount of information is not a bottleneck for the robot, due to its high working memory capacity.

## IX. CONCLUSION

We presented a method to: 1) extract the important features of a task, where the important features consisted of spatio-temporal correlations across a multivariate dataset, 2) to determine a generic metric to evaluate the robot's imitative performance, and, finally, 3) to optimize the robot's reproduction of the task, according to the metric of imitation performance and when placed in a new context.

The method was validated in three experiments in which a humanoid robot was taught simple manipulation tasks through kinesthetics. Various types of regularities were extracted from the demonstrated motions, defining time-varying constraints that drive the reproduction of the motions. We showed that the important features of the task were successfully reproduced by the robot, and, this for different initial conditions.

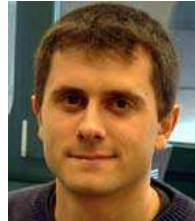
## ACKNOWLEDGMENT

The work described in this paper was supported in part by the Secretariat d'Etat a 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] M. Skubic and R. Volz, "Acquiring robust, force-based assembly skills from human demonstration," in *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, 2000, pp. 772–781.
- [3] M. Yeasin and S. Chaudhuri, "Toward automatic robot programming: learning human skill from visual data," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 30, no. 1, pp. 180–185, 2000.
- [4] A. Billard and R. Siegwart, "Robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 65–67, 2004.
- [5] A. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 15, 2002, pp. 1547–1554.
- [6] A. Ude, C. Atkeson, and M. Riley, "Programming full-body movements for humanoid robots by observation," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 93–108, 2004.
- [7] A. Shon, K. Grochow, and R. Rao, "Robotic imitation from human motion capture using gaussian processes," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2005.
- [8] 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.
- [9] R. Zöllner, 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.
- [10] J. J. Steil, F. Röthling, R. Haschke, and H. Ritter, "Situating robot learning for multi-modal instruction and imitation of grasping," in *Robotics and Autonomous Systems*, 2004, vol. 47, no. 2-3, pp. 129–141.
- [11] 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.
- [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] G. Tevatia and S. Schaal, "Inverse kinematics for humanoid robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2000.
- [14] S. Calinon, F. Guenter, and A. Billard, "On learning the statistical representation of a task and generalizing it to various contexts," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2006.
- [15] 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.
- [16] C. Chiu, S. Chao, M. Wu, and S. Yang, "Content-based retrieval for human motion data," *Visual Communication and Image Representation*, vol. 15, pp. 446–466, 2004.
- [17] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.
- [18] G. McLachlan and D. Peel, *Finite Mixture Models*. Wiley, 2000.
- [19] A. Dempster and N. L. D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society B*, vol. 39, no. 1, pp. 1–38, 1977.
- [20] M. Carreira-Perpiñán, "Continuous latent variable models for dimensionality reduction and sequential data reconstruction," Ph.D. dissertation, Dept. of Computer Science, University of Sheffield, UK, 2002.
- [21] N. Vlassis and A. Likas, "A greedy EM algorithm for Gaussian mixture learning," *Neural Processing Letters*, vol. 15, no. 1, pp. 77–87, 2002.
- [22] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, pp. 461–464, 1978.
- [23] D. Cohn, Z. Ghahramani, and M. Jordan, "Active learning with statistical models," *Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.
- [24] A. Billard, S. Calinon, and F. Guenter, "Discriminative and adaptive imitation in uni-manual and bi-manual tasks," *Robotics and Autonomous Systems*, vol. 54, no. 5, 2006.
- [25] H. Bekkering, A. Wohlschläger, and M. Gattis, "Imitation of gestures in children is goal-directed," *Quarterly Journal of Experimental Psychology*, vol. 53A, no. 1, pp. 153–164, 2000.
- [26] K. Yamane and Y. Nakamura, "Dynamics filter - concept and implementation of online motion generator for human figures," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 3, pp. 421–432, 2003.
- [27] M. Hersch and A. Billard, "A biologically-inspired model of reaching movements," in *Proceedings of the 2006 IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2006.
- [28] S. Calinon and A. Billard, "Teaching a humanoid robot to recognize and reproduce social cues," in *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2006.
- [29] S. Calinon, J. Epiney, and A. Billard, "A humanoid robot drawing human portraits," in *IEEE-RAS International Conference on Humanoid Robots*, 2005.



**Sylvain Calinon** was born in 1980 in Switzerland. He received his BSc/MSc in MicroEngineering with specialization in Robotics from the Ecole Polytechnique Fédérale de Lausanne (EPFL) in March 2003. He then joined the Learning Algorithms and Systems Laboratory (LASA) at EPFL, where he is currently finishing his Ph.D. thesis. His research interests cover Robot Programming by Demonstration, Machine Learning and Human-Robot Interaction.



**Florent Guenter** was born in 1979 in Sion, Switzerland. He obtained his BSc and MSc in MicroEngineering with specialization in Robotics from EPFL (Ecole Polytechnique Fédérale de Lausanne) in 2004. He currently pursues his studies at the Learning Algorithms and Systems Laboratory (LASA) at EPFL to obtain a Ph.D degree. His research interests include Robot control and Programming by Demonstration.



**Aude Billard** is Associate Professor and head of the Learning Algorithms and Systems Laboratory at the school of Engineering at EPFL (Ecole Polytechnique Fédérale de Lausanne), which she joined in 2002. She received her BSc (1994) and MSc (1995) in Physics from EPFL, with specialization in Particle Physics at the European Center for Nuclear Research (CERN). She received her MSc in Knowledge Based System (1996) and her PhD in Artificial Intelligence (1998) from the department of Artificial Intelligence at the University of Edinburgh. She, then, was a Post-doctoral Fellow at IDSIA and LAMI (EPFL, 1998-99). Aude Billard was a Research Associate (1999-2000), Research Assistant Professor (2000-2002) and remained an adjunct faculty since then at the Computer Science department at the University of Southern California (USC). Dr. Billard's research interests cover the fields of Artificial Neural Networks, Robotics, Neural Modeling, Computational Neuroscience and more generally Machine Learning. Her work tackles special topics, such as Programming Through Demonstration, Imitation Learning and Language Acquisition.