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

Sylvain Calinon, Florent Guenter, and Aude Billard, *Member, IEEE*

***Abstract*—We present a programming-by-demonstration 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. The resulting signals are encoded using a mixture of Gaussian/Bernoulli distributions (Gaussian mixture model/Bernoulli mixture model). 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. Finally, we analytically compute the trajectory which optimizes the imitation metric and use this to generalize the skill to different contexts.**

***Index Terms*—Gaussian mixture model (GMM), human motion subspace, human–robot interaction (HRI), learning by imitation, metric of imitation, programming by demonstration (PbD).**

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 in 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 this paper, 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 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 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 this paper, 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. Experimental Scenario

We present an architecture that will generically solve the problem of extracting the relevant features of a given task

Manuscript received December 15, 2005; revised March 20, 2006. This work was partially conducted within the EU Integrated Project COGNIRON ("The Cognitive Companion") and supported in part by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020 and in part by the Swiss National Science Foundation under Grant 620-066127 of the SNF Professorships program. This paper was recommended by Guest Editor Y. Demiris.

The authors are with the Learning Algorithms and Systems Laboratory, Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2006.886952

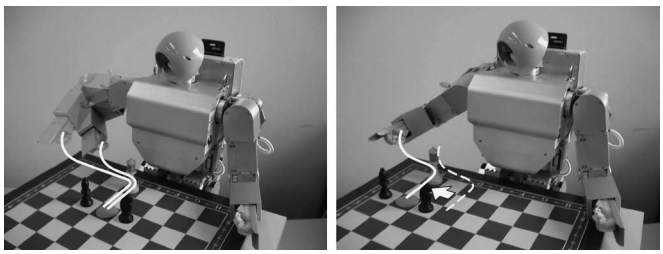

Fig. 1. (Left) Illustration of the what-to-imitate issue. (Right) Illustration of the how-to-imitate issue.

(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].

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 is 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 toward 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 chessboard. 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 angles configuration than the one demonstrated. In order to do this, the robot computes the trajectory which gives the optimal tradeoff 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.

1) *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.
2) *Metric of imitation:* A time-dependent similarity measure is defined by taking into account the relative importance of each variable and the dependences across the variables using the probabilistic representation of the data. This measure evaluates the reproduction performance of a task.
3) *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, reduction of dimensionality, constraints on the dataset, and temporal alignment of the signals. Section V presents the probabilistic encoding in mixture models, the criteria 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 1000 Hz during the demonstrations and were then resampled 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 in demonstrating a task to the robot using its own body (see Fig. 3). We 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 in grasping and releasing 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 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

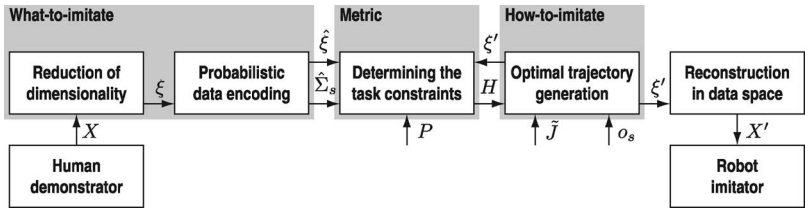

Fig. 2. Information flow across the complete system. The original demonstrated signals X are projected onto a latent space, with ξ as the projected signals. $\hat{\xi}$ is the generalized version of the signals with associated time-dependent covariance matrix $\hat{\Sigma}_s$. P is the optional time-dependent priors matrix. $\tilde{J}$ 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 ξ' are the reproduced signals in the original data space and latent space.

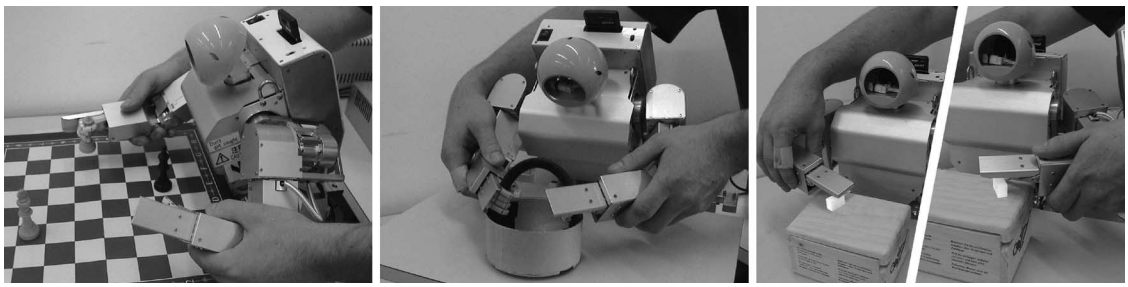

Fig. 3. Teaching through kinesthetics for the three experiments conducted. Chess task: Grabbing and moving a chess piece two squares forward. Bucket task: Grabbing and bringing a bucket to a specific position. Sugar task: Grabbing a piece of sugar and bringing it to the mouth, using either the right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

systems is that it provides a fast and accurate way of acquiring data because it does not require wearing 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 the whole body motion because it is not possible to control simultaneously more DOFs (e.g., moving the two arms and two legs of the robot 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 is 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; and 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 frame 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

TABLE I
NOTATIONS USED THROUGHOUT THIS PAPER

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
ξ	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

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 paper. The training set consists of N observations of the set of variables $\{\theta, x, y, h\}$. Each variable consists of a 1-D 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.,

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

The initial position of the 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 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-D original Cartesian position of the hand onto a 2-D latent space defined by the writing surface, while a waving motion is typically represented as a combination of a single 1-D 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 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 ξ_s. This goal cannot 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 PCA and independent component analysis (ICA) in reducing the dimensionality of human motion data. We found that ICA presented very few advantages over PCA, i.e., decorrelation was a sufficient preprocessing step. The uncorrelatedness assumption of PCA is weaker than statistical independence, but it is still a sufficient constraint in decomposing our human motion dataset. Thus, we are using a PCA that finds analytically a mixing matrix A, projecting the dataset X_s onto uncorrelated components ξ_s, with the criteria 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 $(D-1)$

$$X_s - \bar{X}_s = A\xi_s \tag{2}$$

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 λ_i are computed, given $\Sigma v_i = \lambda_i v_i$, $\forall i \in \{1, \ldots, (d-1)\}$. 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 $A = \{v_1, v_2, \ldots, v_{D-1}\}$, 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

$$\dot{X}_{s,i,j} = X_{s,i,j} - X_{s,i,j-1} \quad \left| \begin{matrix} \forall i \in \{1, \ldots, n\} \\ \forall j \in \{2, \ldots, T\}. \end{matrix} \right. \tag{3}$$

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

$$\dot{\xi}_{s,i,j} = \xi_{s,i,j} - \xi_{s,i,j-1} \quad \left| \begin{matrix} \forall i \in \{1, \ldots, n\} \\ \forall j \in \{2, \ldots, T\}. \end{matrix} \right. \tag{4}$$

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

$$\dot{X}_s = A\dot{\xi}_s. \tag{5}$$

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

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

where $(A^x)^{-1}$ is the pseudoinverse 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 rewritten in the latent space, providing a constraint between $\dot{\xi}_s^x$ and $\dot{\xi}_s^y$

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

where $\dot{o}_s$ is the initial velocity of objects (null in our experiments) and $(A^y)^{-1}$ is the pseudoinverse 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 GMMs [14]. The aim of this method was to combine the temporal alignment properties of HMMs with

the regression capabilities of a static GMM encoding. However, the temporal information encoded in the two models was redundant, modeled as transition probabilities in the HMM and as an additional variable in the GMM. In HMM, multivariate Gaussians are modeling local portions of the signals. By probabilistically encoding the transitions between these Gaussians, HMMs can deal with nonhomogenous temporal deformation of the signals. They act as a method in temporally aligning different signals, represented spatially by the Gaussians. HMM can be seen as 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 in retrieving 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 that can be essential in reproducing 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 in reconstructing the "best" sequence from a Gaussian model.

In this paper, temporal alignment of the signals is performed by a pattern-based approach used as a preprocessing 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. DTW is used as a template matching preprocessing 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 dataset, 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. 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 nonhomogenous time distortions. We thus explore the use of DTW as a preprocessing step in improving 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 nonlinear alignment of the signals which minimizes the error with respect to a 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 ξ_s^A and ξ_s^B of length T. We define the distance measured between two datapoints of temporal index k_1 and k_2 by $h(k_1, k_2) = \|\xi^A_{s,k_1} - \xi^B_{s,k_2}\|$. 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 $s_L = \{T, T\}$. If $s_l = \{a, b\}$ and $s_{l-1} = \{a', b'\}$, 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 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 modeling 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 tradeoff between model complexity and variations of the available training data. A mixture model of K components is defined by a probability density function

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

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

A. Gaussian Mixture Model

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, hands paths, or hands–object distance vectors. The dataset is modeled by a mixture of K Gaussians of dimensionality D. The parameters in (8) become

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

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 monotonic 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

To encode a set of $(D-1)$-dimensional binary datapoints $h_{s,j} = \{h_{s,j,i}\}_{i=1}^{D-1}$ 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 dependences across the binary data are captured due to the contribution of the different components of the mixture.

A mixture of $(D-1)$-dimensional Bernoulli density function 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

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

Similar 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 criteria (see, e.g., [21]).

We therefore need to arrive at a tradeoff 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 criteria, Bayesian information criteria (BIC), or minimum description length are commonly found in the literature. Cross validation has the disadvantage of requiring additional demonstrations to form a test set. We selected the 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

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

where $\mathcal{L} = \sum_{j=1}^{N} \log(p(\xi_j))$ is the log-likelihood of the model using the demonstrations as testing set and n_p is the number of free parameters required for a mixture of K components, i.e., $n_p = (K-1) + KD$ for a BMM and $n_p = (K-1) + K(D + (1/2)D(D+1))$ for a GMM 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 ten 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

To reconstruct a general form for the signals, we apply GMR [23]. Consecutive temporal values ξ_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

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

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

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

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

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

Using (10) and (11), for a mixture of K components, the condition expectation of ξ_s, given ξ_t, and the conditional covariance of ξ_s, given ξ_t, are

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

Thus, by evaluating $\{\hat{\xi}_s, \hat{\Sigma}_s\}$ at different time steps ξ_t, a generalized form of the motions $\hat{\xi} = \{\xi_t, \hat{\xi}_s\}$ and associated covariance matrices 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 our 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 ξ_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 dependences 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 hands paths, 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

$$H = \left(\xi_s^\theta-\hat{\xi}_s^\theta\right)^T W^\theta\left(\xi_s^\theta-\hat{\xi}_s^\theta\right) + \left(\xi_s^x-\hat{\xi}_s^x\right)^T W^x\left(\xi_s^x-\hat{\xi}_s^x\right) + \left(\xi_s^y-\hat{\xi}_s^y\right)^T W^y\left(\xi_s^y-\hat{\xi}_s^y\right). \quad (12)$$

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

$$\begin{aligned} c_{1,i,j} &= \hat{\xi}_{s,i,j}^\theta - \xi_{s,i,j-1}^\theta \\ c_{2,i,j} &= \hat{\xi}_{s,i,j}^x - \xi_{s,i,j-1}^x \\ c_{3,i,j} &= \hat{\xi}_{s,i,j}^y - \xi_{s,i,j-1}^y \end{aligned} \quad \left| \begin{array}{l} \forall i \in \{1,\ldots,n\} \\ \forall j \in \{2,\ldots,T\} \end{array} \right.$$

and rewrite (12) as

$$H = \left(\dot{\xi}_s^\theta-c_1\right)^T W^\theta\left(\dot{\xi}_s^\theta-c_1\right) + \left(\dot{\xi}_s^x-c_2\right)^T W^x\left(\dot{\xi}_s^x-c_2\right) + \left(\dot{\xi}_s^y-c_3\right)^T W^y\left(\dot{\xi}_s^y-c_3\right). \quad (13)$$

C. Finding an Optimal Controller

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

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

where λ_1 and λ_2 are the vectors of associated Lagrange multipliers. To solve $\nabla L = 0$, we consider symmetric matrices $W^T = W$ and derive, respectively, $\partial L/\partial\dot{\xi}_s^\theta = 0$, $\partial L/\partial\dot{\xi}_s^x = 0$, and $\partial L/\partial\dot{\xi}_s^y = 0$

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

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

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

Using (15) and (16), we find

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

Using (17) and (14), we find

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

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

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

with

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

We can then compute iteratively the joint angle trajectories with

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

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

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.

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

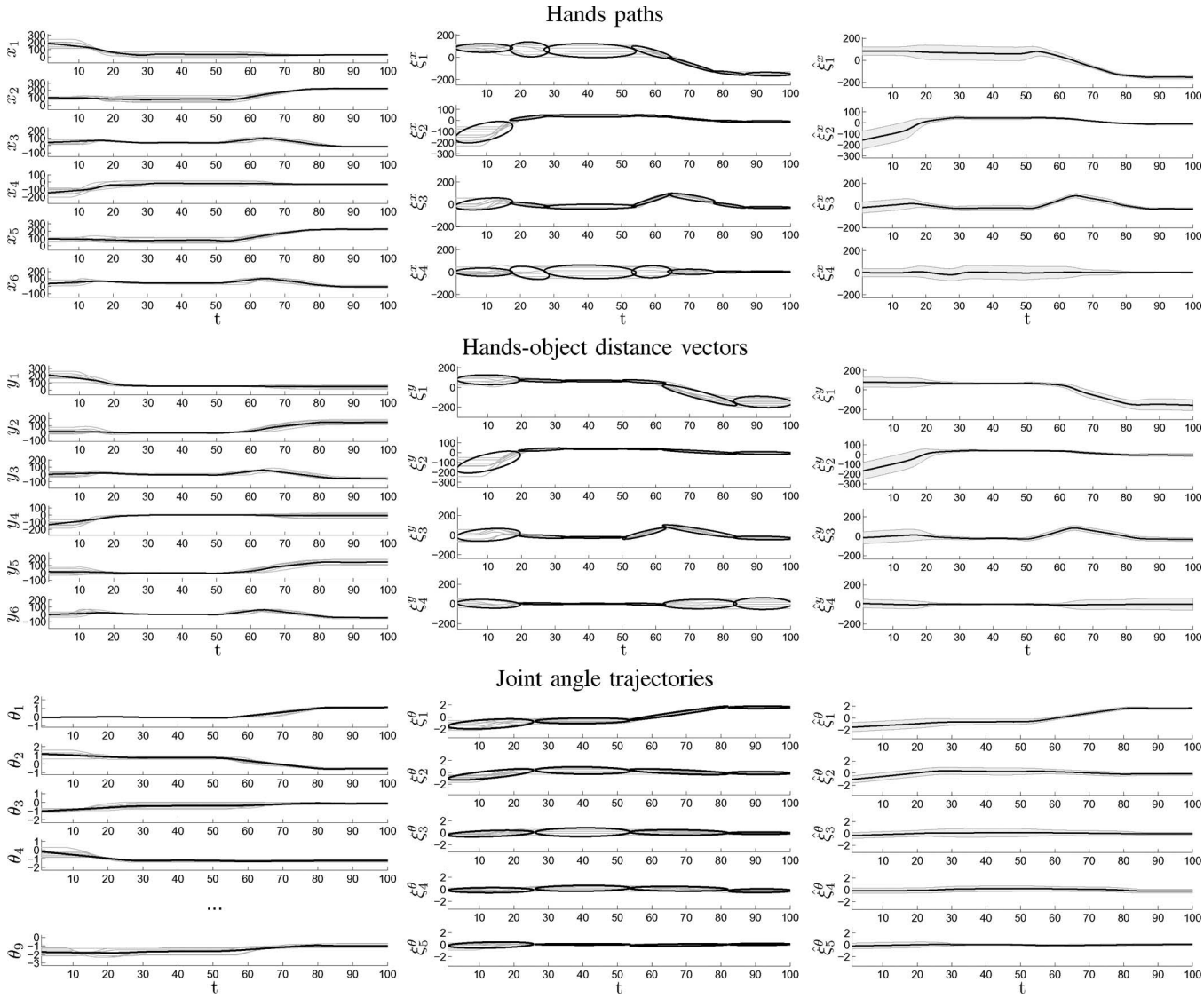

Fig. 4. Probabilistic encoding for the bucket task. (Left column) Generalization of the hands' paths x ($\{x_1, x_2, x_3\}$ and $\{x_4, x_5, x_6\}$ represent, respectively, the right/left hand paths), hands–object relationships y ($\{y_1, y_2, y_3\}$ and $\{y_4, y_5, y_6\}$ represent, respectively, the relationships for the right/left hand), and joint angle trajectories θ (θ_1 represents the torso joint angle and $\{\theta_2, \ldots, \theta_5\}$ and $\{\theta_6, \ldots, \theta_9\}$ represent, respectively, the right/left arm joint angles). The demonstrations are represented in gray lines, and the generalized signal reconstructed from the latent space is represented in bold lines. (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 ξ are encoded in GMM, in which the covariance matrices are 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 lines, 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).

shown the task four to seven 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 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

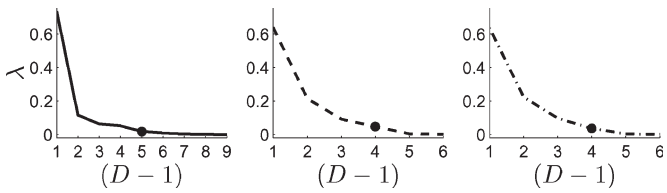

Fig. 5. Estimation of the number of components required to reduce the dimensionality of the data space for the bucket task using eigenvalues (solid line for the joint angles dataset, dashed line for the hands' paths dataset, and dashed–dotted line for the hands–object distance vectors dataset). The point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

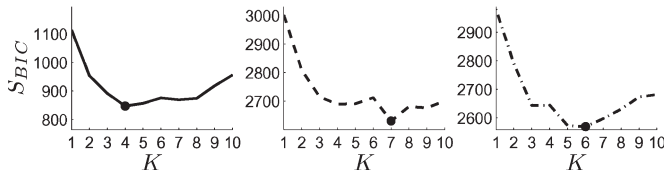

Fig. 6. Estimation of the number of Gaussian components required to model the trajectories in the latent space for the bucket task (solid line for the joint angles dataset, dashed line for the hands' paths dataset, and dashed–dotted line for the hands–object relationships dataset). Adding more Gaussian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

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
Chess Task	θ	7	100	9	4	5
	x	7	100	6	4	5
	y	7	100	6	4	4
	h	7	100	2	(2)	1
Bucket Task	θ	7	100	9	5	4
	x	7	100	6	4	7
	y	7	100	6	4	6
	h	7	100	2	(2)	1
Sugar Task - left	θ	4	100	9	2	5
	x	4	100	6	3	5
	y	4	100	6	3	6
	h	4	100	2	(2)	1
Sugar Task - right	θ	4	100	9	2	6
	x	4	100	6	3	6
	y	4	100	6	3	6
	h	4	100	2	(2)	1

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.

Figs. 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 that the dimensionality of the latent space for the chess task and bucket task is higher than for the sugar task, probably due to the fact 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

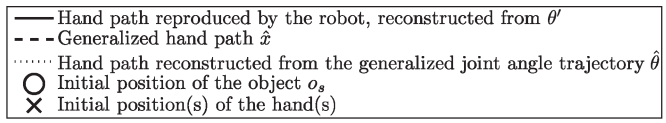

Fig. 7. Legend for Figs. 8–13.

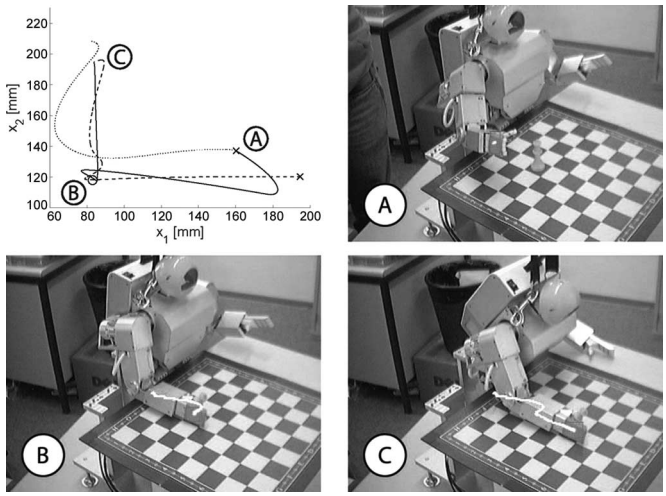

Fig. 8. Decomposition of the chess task when reproducing the task with an initial position of the object which is close to the generalized trajectories. The hands' paths have been tracked by a stereoscopic vision system.

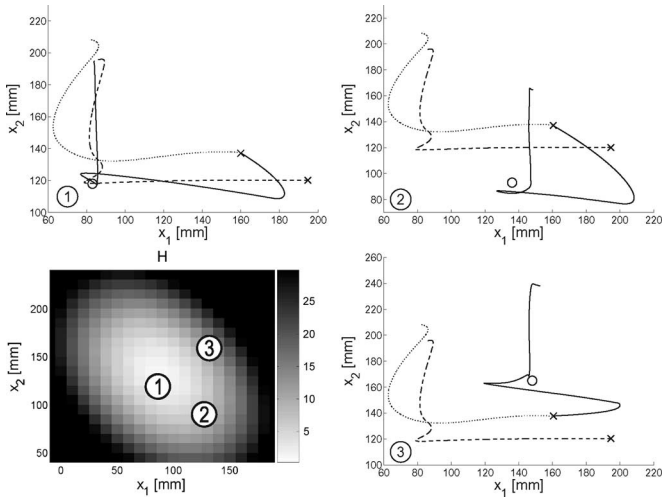

Fig. 9. (Bottom left) Mean values taken by the cost function H for the chess task, with a varying initial position of the chess piece on the board. 1, 2, and 3: Reproduction for the corresponding three locations on the map.

is performing the task. The number of Gaussian components is between four and seven. The number of components for the binary signals h is always one (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 $n_{\mathrm{PCA}} = (D-1)(d-1)$ and $n_{\mathrm{GMM}} = (K-1) + K(D + (1/2)D(D+1))$.

Figs. 8–13 are using the same legend as that presented in Fig. 7. Figs. 8 and 9 show the reproduced trajectories for the chess task, depending on the initial position of the chess

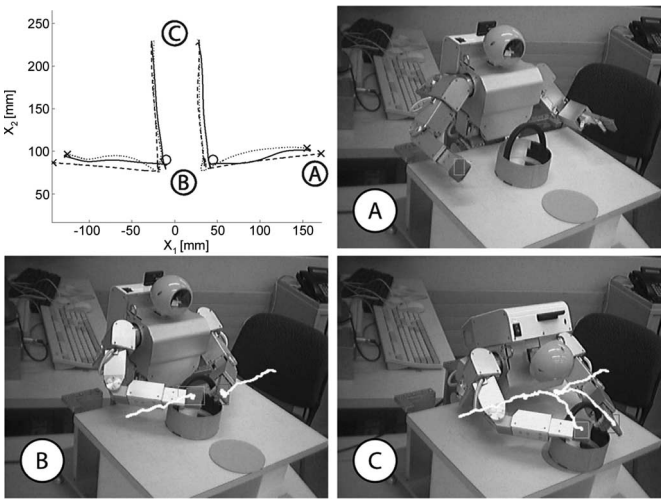

Fig. 10. Decomposition of the bucket task, when reproducing the task with an initial position of the object close to the generalized trajectories.

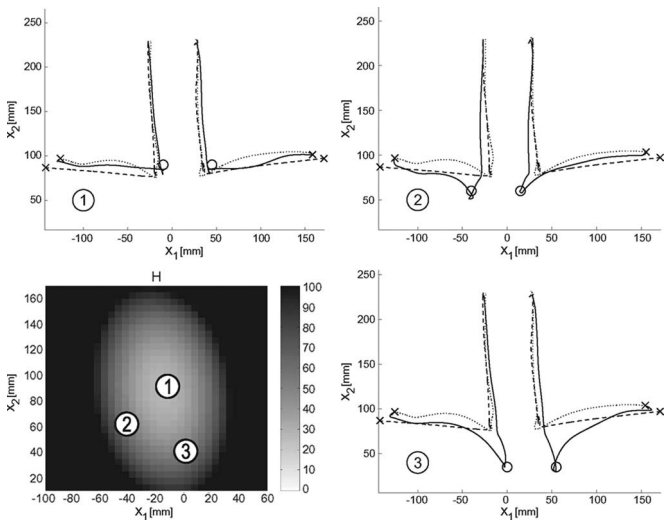

Fig. 11. (Bottom left) Mean values taken by the cost function H for the bucket task with a varying initial position of the bucket on the table. 1, 2, and 3: Reproduction for the corresponding three locations on the map.

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 in reproducing the motion is to initially place the chess piece in front of the right arm [see Fig. 9, inset (1)]. In inset (2) and (3) of Fig. 9, the chess piece is placed initially at different positions unobserved during the demonstrations.

Figs. 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.

Figs. 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. First, the robot is taught how to grasp with its right hand a piece of sugar located at the

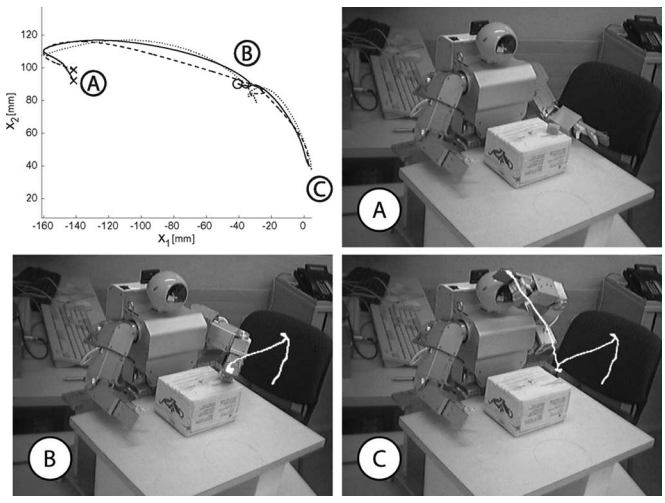

Fig. 12. Decomposition of the sugar task for the left hand when reproducing the task with an initial position of the object close to the generalized trajectories.

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 *et al.* [25] have set up experiments to show that imitation is goal-directed, using several gestures to reproduce during an imitation game. 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 a 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.

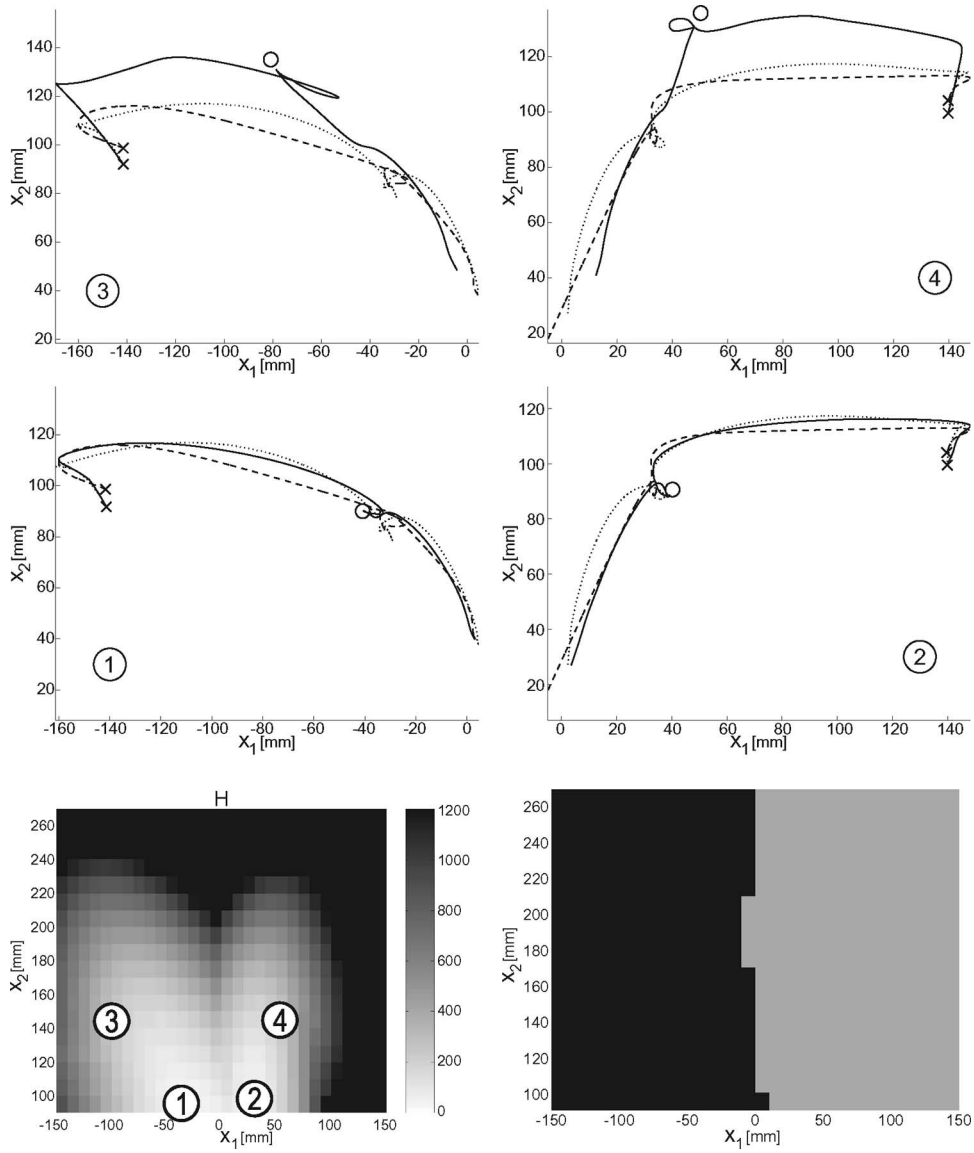

Fig. 13. (Bottom left) Mean values taken by the cost function H for the sugar task, with a variation in the initial position of the piece of sugar on the box. We distinguish two different areas for the left and right arm. The difference in size is due to the different variations used for the left and 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, and 4: Reproduction for the corresponding four locations on the map.

The presented system can deal with two types of generalizations: 1) by projecting the original data onto a latent space and encoding the resulting data in GMMs and BMMs, 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; and 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-D 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 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 was 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 in describing 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 cojointly with other systems to refine the solution. Our current work investigates the use of this system in providing 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 whom 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) determine a generic metric to evaluate the robot's imitative performance; and, finally, 3) 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 is for different initial conditions.

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*, vol. 24, J. Demiris and A. Birk, Eds. Singapore: World Scientific, 2000, pp. 136–161.

[2] M. Skubic and R. Volz, "Acquiring robust, force-based assembly skills from human demonstration," *IEEE Trans. Robot. Autom.*, vol. 16, no. 6, pp. 772–781, Dec. 2000.

[3] M. Yeasin and S. Chaudhuri, "Toward automatic robot programming: Learning human skill from visual data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 30, no. 1, pp. 180–185, Feb. 2000.

[4] A. Billard and R. Siegwart, "Robot learning from demonstration," *Robot. Auton. Syst.*, 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 *Proc. Adv. NIPS*, 2002, vol. 15, pp. 1547–1554.

[6] A. Ude, C. Atkeson, and M. Riley, "Programming full-body movements for humanoid robots by observation," *Robot. Auton. Syst.*, 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 *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2005, pp. 129–134.

[8] S. Calinon and A. Billard, "Recognition and reproduction of gestures using a probabilistic framework combining PCA, ICA and HMM," in *Proc. ICML*, 2005, pp. 105–112.

[9] R. Zöllner, M. Pardowitz, S. Knoop, and R. Dillmann, "Towards cognitive robots: Building hierarchical task representations of manipulations from human demonstration," in *Proc. IEEE ICRA*, 2005, pp. 1535–1540.

[10] J. J. Steil, F. Röthling, R. Haschke, and H. Ritter, "Situated robot learning for multi-modal instruction and imitation of grasping," *Robot. Auton. Syst.*, vol. 47, no. 2/3, pp. 129–141, 2004.

[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 *Proc. IEEE Int. Symp. Comput. Intell. Robot. and Autom.*, 2005, pp. 61–66.

[12] S. Calinon, F. Guenter, and A. Billard, "Goal-directed imitation in a humanoid robot," in *Proc. IEEE ICRA*, 2005, pp. 299–304.

[13] G. Tevatia and S. Schaal, "Inverse kinematics for humanoid robots," in *Proc. IEEE ICRA*, 2000, pp. 294–299.

[14] S. Calinon, F. Guenter, and A. Billard, "On learning the statistical representation of a task and generalizing it to various contexts," in *Proc. IEEE ICRA*, 2006, pp. 2978–2983.

[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 *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2005, pp. 469–474.

[16] C. Chiu, S. Chao, M. Wu, and S. Yang, "Content-based retrieval for human motion data," *J. Vis. Commun. Image Represent.*, vol. 15, no. 3, pp. 446–466, 2004.

[17] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-26, no. 1, pp. 43–49, Feb. 1978.

[18] G. McLachlan and D. Peel, *Finite Mixture Models*. Hoboken, NJ: Wiley, 2000.

[19] A. Dempster and N. L. D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc. 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. Comput. Sci., Univ. Sheffield, Sheffield, U.K., 2002.

[21] N. Vlassis and A. Likas, "A greedy EM algorithm for Gaussian mixture learning," *Neural Process. Lett.*, vol. 15, no. 1, pp. 77–87, 2002.

[22] G. Schwarz, "Estimating the dimension of a model," *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, 1978.

[23] D. Cohn, Z. Ghahramani, and M. Jordan, "Active learning with statistical models," *Artif. Intell. Res.*, 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," *Robot. Auton. Syst.*, vol. 54, no. 5, pp. 370–384, 2006.

[25] H. Bekkering, A. Wohlschläger, and M. Gattis, "Imitation of gestures in children is goal-directed," *Q. J. Exp. Psychol.*, 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 Trans. Robot. Autom.*, vol. 19, no. 3, pp. 421–432, Jun. 2003.

[27] M. Hersch, F. Guenter, S. Calinon, and A. G. Billard, "Learning Dynamical System Modulation for Constrained Reaching Tasks," in *Proc. IEEE–RAS Int. Conf. Humanoid Robots*, 2006, pp. 444–449.

[28] S. Calinon and A. Billard, "Teaching a humanoid robot to recognize and reproduce social cues," in *Proc. IEEE Int. Symp. RO-MAN*, 2006, pp. 346–351.

[29] S. Calinon, J. Epiney, and A. Billard, "A humanoid robot drawing human portraits," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2005, pp. 161–166.

Sylvain Calinon was born in Switzerland, in 1980. He received the B.Sc. and M.Sc. degrees in microengineering, with specialization in robotics, from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 2003, where he is currently working toward the Ph.D. degree.

He is now with the Learning Algorithms and Systems Laboratory (LASA), EPFL. His research interests cover robot programming by demonstration, machine learning, and human–robot interaction.

Florent Guenter was born in Sion, Switzerland, in 1979. He received the B.Sc. and M.Sc. degrees in microengineering, with specialization in robotics, from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 2004, where he is currently working toward the Ph.D. degree.

His research interests include robot control and programming by demonstration.

Aude Billard (M'00) received the B.Sc. and M.Sc. degrees in physics from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 1994 and 1995, respectively, with specialization in particle physics at the European Center for Nuclear Research (CERN), and the M.Sc. degree in knowledge-based systems, in 1996, and the Ph.D. degree in artificial intelligence from the University of Edinburgh, Edinburgh, U.K., in 1998.

She was a Postdoctoral Fellow with the Dalle Molle Institute for Artificial Intelligence (IDSIA) and with the Microprocessor Systems Laboratory (LAMI) (EPFL, 1998–1999). She was then a Research Associate (1999–2000) and Research Assistant Professor (2000–2002) and remained an adjunct faculty since then at the Computer Science Department, University of Southern California. She is currently an Associate Professor and Head of the Learning Algorithms and Systems Laboratory, School of Engineering, EPFL, where she joined in 2002. Her research interests cover the fields of artificial neural networks, robotics, neural modeling, computational neuroscience, and more general machine learning. Her work tackles special topics such as programming through demonstration, imitation learning, and language acquisition.