Learning of Gestures by Imitation in a Humanoid Robot

Sylvain Calinon and Aude Billard

Autonomous Systems Lab
Swiss Federal Institute of Technology Lausanne - EPFL
CH-1015 Lausanne, Switzerland
{sylvain.calinon,aude.billard}@epfl.ch
http://asl.epfl.ch/

Keywords: Imitation Learning, PCA, HMM, Humanoid Robots, Programming by Demonstration

1 Introduction

Traditionally, robotics developed highly specific controllers for the robot to perform a specific set of tasks in highly constrained and deterministic environments. This required to embed the controller with an extensive knowledge of the robot's architecture and of its environment. It was soon clear that such an approach would not scale up for controlling robots with multiple degrees of freedom, working in highly variable environments, such as humanoid robots required to interact with humans in their daily environment.

The field has now moved to developing more flexible and adaptive control systems, so that the robot would no longer be dedicated to a single task, and could be re-programmed in a fast and efficient manner, to match the end-user needs.

Robot learning by imitation, also referred to as *robot programming by demonstration*, explores novel means of implicitly teaching a robot new motor skills [5, 10, 20]. This field of research takes inspiration in a large and interdisciplinary body of literature on imitation learning, drawing from studies in Psychology, Ethology and the Neurosciences [9, 4, 1]. To provide a robot with the ability to imitate is advantageous for at least two reasons: it provides a natural, user-friendly means of implicitly programming the robot; it constrains the search space of motor learning by showing possible and/or optimal solutions.

In this chapter, we explore the issue of recognizing, generalizing and reproducing arbitrary gesture [3]. In order to take a general stance toward gesture recognition and reproduction, we address one major and generic issue, namely *how to discover the essence of a gesture*, i.e. how to find a representation of the data that encapsulates only the key aspects of the gesture, and discards the intrinsic variability across people motion.

To illustrate the idea, consider the following examples: when asked to imitate someone writing letters of the alphabet on a board, you will find it sufficient to only track the trajectory followed by the demonstrator's hand on the board. In contrast, when learning to play tennis, you will find it more important to follow

To appear in: Dautenhahn and Nehaniv, editors, Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions. Cambridge University Press, 2007.

the trajectories of the demonstrator's arm joint angles, rather than the position of the hand (the ball's position varying importantly over the trials). Choosing, in advance, the optimal representation of the data (whether hand-path or joint angles) would greatly simplify the analysis and speed up learning.

In the application presented in this chapter, the robot is endowed with numerous sensors enabling it to track faithfully the demonstrator's motions. Some of the data gathered by the sensors are redundant and correlated. The first stage of processing performed on our data consists in applying Principal Component Analysis (PCA) in order to determine a space in which the data are decorrelated, and, consequently, to reduce the dimensionality of the dataset, so as to make the analysis more tractable.

In order for the robot to learn new skills by imitation, it must be endowed with the ability to generalize over multiple demonstrations. To achieve this, the robot must encode multivariate time-dependent datasets in an efficient way. One of the major difficulty in learning, recognizing and reproducing sequential patterns of motion is to deal simultaneously with the variations in the data and with the variations in the sequential structure of these data. The second stage of processing of our model uses Hidden Markov Models (HMMs) to encode the sequential patterns of motion in stochastic finite state automata. The motion is then represented as a sequence of states, where each state has an underlying description of multi-dimensional data (see Figure 4). The system takes inspiration in a recent trend of research that aims at defining a formal mathematical framework for imitation learning [19, 15, 3]. We present an implementation of these approaches in a noisy real-world application.

The remaining of this chapter is divided as follows: section 2 presents the experimental set-up. Section 3 describes in details the model. Results are presented in Section 4, and discussed in Section 5, stressing out the parallels existing between our robotic model and theoretical models of imitation learning in animals.

2 Experimental Set-up

Data used for training the robot have been generated by eight healthy volunteers (students at the EPFL School of Engineering). Subjects have been asked to imitate a set of 6 motions performed by a human demonstrator in a video. The motions consist in:

- Knocking on a door
- Raising a glass, drinking, and putting it back on a table
- Waving goodbye
- Drawing the stylized alphabet letters A, B and C

The subject's gestures have been recorded by 3 x-sens motion sensors, attached to the torso and the right upper- and lower-arm. Each sensor provides the 3D absolute orientation of each segment, by integrating the 3D rate-of-turn, acceleration and earth-magnetic field, at a rate of 100Hz. The joint angle trajectories of the shoulder joint (3 degrees of freedom (DOFs)) and of the elbow

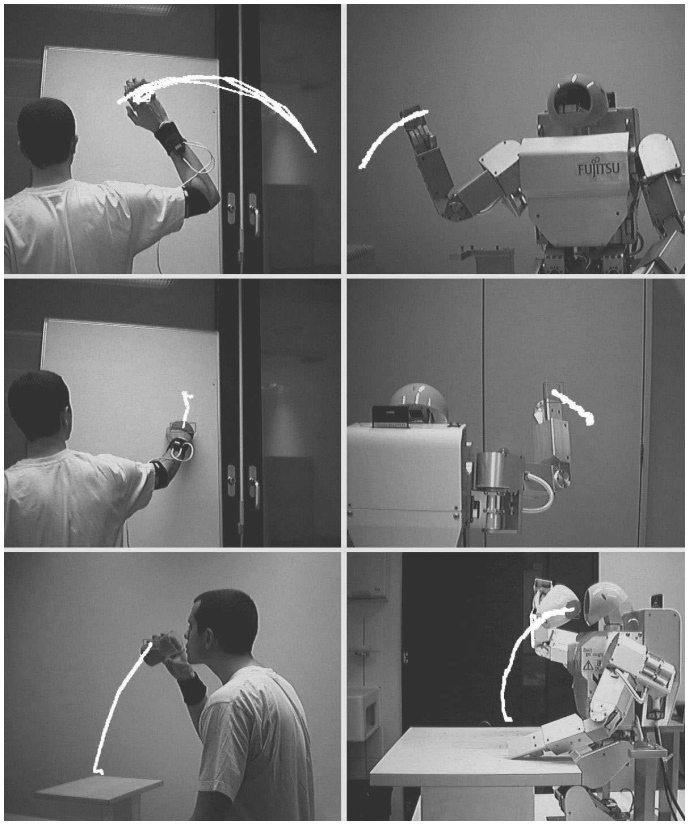

Fig. 1. Demonstration (*left column*) and reproduction (*right column*) of different tasks: waving goodbye (*1st line*), knocking on a door (*2nd line*), and drinking (*3rd line*). Three gyroscopic motion tracking sensors are attached on the upper arm, lower arm and torso of the demonstrator. The trajectories of the demonstrator's hand, reconstructed by the stereoscopic vision system, are superimposed to the image.

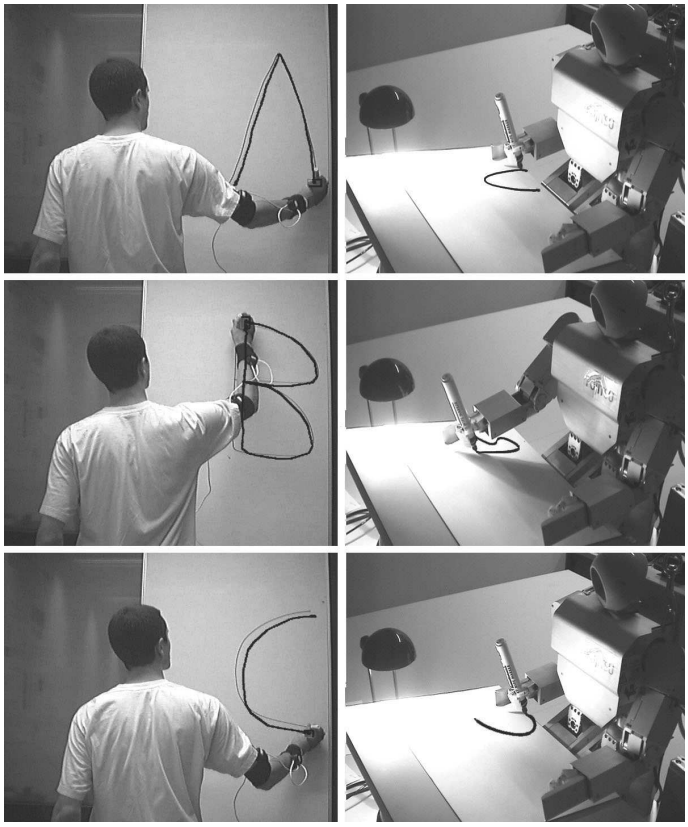

Fig. 2. Demonstration (*left column*) and reproduction (*right column*) of drawing the three stylized alphabet letters A, B and C. The motion reproduced by the robot follows a trajectory generalized across the different demonstrations. As the trajectory is projected by PCA in 2 dimensions, the letters can be written on a different plane.

(1 DOF) are reconstructed with a precision of 1.5 degrees, taking the torso as reference. These sensors provide a motor representation of the gesture, that can be used without major modification to control the robot.

A color-based stereoscopic vision system tracks the 3D-position of a marker placed on the demonstrator's hand, at a rate of 15Hz, with a precision of 10 mm. The system uses 2 Phillips webcams with a resolution of 320x240 pixels. The tracking is based on color segmentation in the YCbCr color space (Y is dismissed to be robust to changes in luminosity).

The robot is a Fujitsu humanoid robot HOAP-2 with 25 DOFs. In the experiments reported here, only the robot's right arm (4 DOFs) is used for the task. The torso and legs are set to a constant and stable position, in order to support the robot's standing-up.

3 Data processing

The complete dataset consists of the trajectories of 4 joint angles and the 3-dimensional trajectory of the hand in Cartesian space. Figure 3 shows a schematic of the sensory-motor flow. The data are first projected onto an uncorrelated, low-dimensional subspace, using PCA. The resulting signals are, then, encoded in a set of Hidden Markov Models. A generalized form of the signals is then reconstructed by interpolating across the time series output by the HMMs and reprojecting onto the original space of the data using the PCA eigenvectors.

For each experiment, the dataset is split equally to training and testing set.

Let $X(t) = \{x_1(t), x_2(t), x_3(t)\}$ be the hand path in Cartesian space, and $\Theta(t) = \{\theta_1(t), \theta_2(t), \theta_3(t), \theta_4(t)\}$ the joint angle trajectories of the right arm, after interpolation, normalization in time (same number of data for each time series), and shifting such as the first data points coincide.

3.1 Preprocessing by Principal Component Analysis (PCA)

PCA is a technique used extensively to discover and reduce the dimensionality of a dataset. In this work, we use it to find a suitable representation of our multivariate dataset [14]. PCA consists in determining the directions (*eigenvectors*) along which the variability of the data is maximal. It assumes that the data are linear and normally distributed.

By projecting the data onto the referential defined by the eigenvectors of the correlation matrix, one obtains a representation of the dataset that minimizes the statistical dependence across the data. Consecutively, dimensionality reduction can be achieved by discarding the dimensions along which the variance of the data is smaller than a criterion. This provides a way to compress the data without losing much information and simplifying the representation.

PCA is applied separately to $\{x_1, x_2, x_3\}$ and $\{\theta_1, \theta_2, \theta_3, \theta_4\}$ to determine if a better representation in each dataset can be used. The means $\{\bar{x}_1, \bar{x}_2, \bar{x}_3\}$ and $\{\bar{\theta}_1, \bar{\theta}_2, \bar{\theta}_3, \bar{\theta}_4\}$ are subtracted for each dimension. Then, the 3 eigenvectors $\{v_1^x, v_2^x, v_3^x\}$ and associated eigenvalues $\{\lambda_1^x, \lambda_2^x, \lambda_3^x\}$ are calculated for the hand

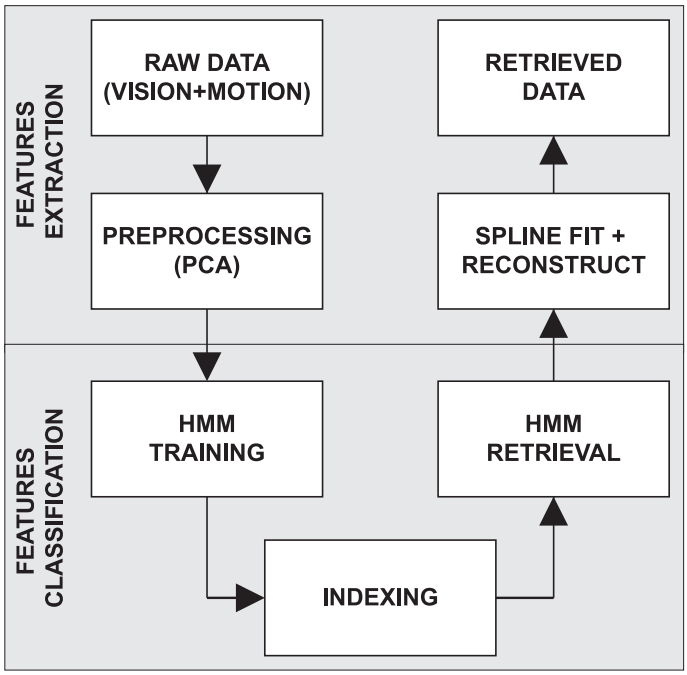

Fig. 3. Schematic of the sensory-motor flow: the data are first projected onto an uncorrelated, low-dimensional subspace, using PCA. The resulting signals are, then, encoded in a set of HMMs. A generalized form of the signals is then reconstructed by interpolating across the time series output by the HMMs and reprojecting onto the original space of the data using the PCA eigenvectors.

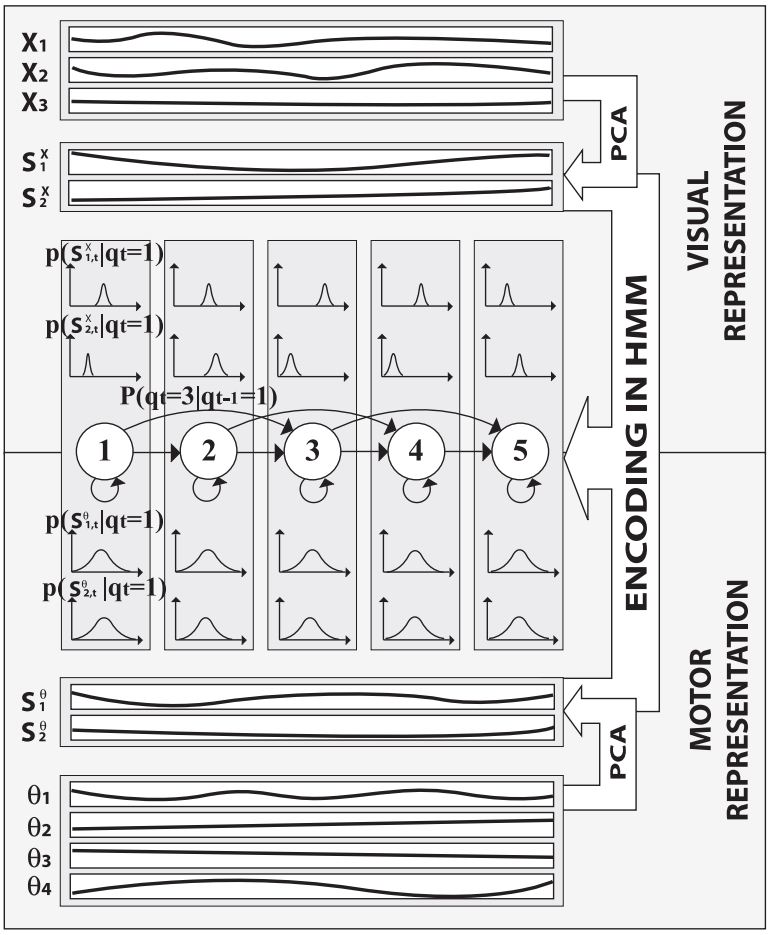

Fig. 4. Encoding of the hand path in Cartesian space $\{x_1, x_2, x_3\}$ and joint angles trajectories $\{\theta_1, \theta_2, \theta_3, \theta_4\}$ in a HMM. The data are pre-processed by PCA, and the resulting signals $\{\xi_1^x, \xi_2^x, \ldots, \xi_I^x\}$ and $\{\xi_1^\theta, \xi_2^\theta, \ldots, \xi_J^\theta\}$ are learned by the HMM. The data are represented as sequences of states, with transition probabilities between the states (not all the transitions are depicted). Each state in the HMM outputs multivariate data, represented by Gaussian functions.

path. The 4 eigenvectors $\{v_1^\theta, v_2^\theta, v_3^\theta, v_4^\theta\}$ and associated eigenvalues $\{\lambda_1^\theta, \lambda_2^\theta, \lambda_3^\theta, \lambda_4^\theta\}$ are calculated for the joint angle dataset. An indication of the relative importance of each direction is given by its eigenvalue. Let I and J be the number of eigenvectors required to obtain a satisfying representation of $\{x_1, x_2, x_3\}$ and $\{\theta_1, \theta_2, \theta_3, \theta_4\}$, such that the information lost by projecting the data onto these eigenvector is small. The selection criterion is to retain the first K components that cover over 80% of data's spread, i.e. $\sum_{i=1}^{K} \lambda_i > 0.8$. By projecting these datasets in the new basis formed by the first K components, the time series become $\{\xi_1^x, \xi_2^x, \ldots, \xi_I^x\}$ with $I \leq 3$ to represent the hand path, and $\{\xi_1^\theta, \xi_2^\theta, \ldots, \xi_J^\theta\}$ with $J \leq 4$ to represent the joint angle trajectories.

Applying PCA before encoding the data in a HMM has the following advantages:

- It helps reducing noise, as the noise is now encapsulated in the lower dimensions (but it also discard the high-frequency information).
- It reduces the dimensionality of the dataset, which reduces the number of parameters in the Hidden Markov Models, and makes the training process faster.
- It produces a parameterizable representation of the dataset that is easier to handle, and that can be used under different conditions, for the reproduction of the task.

For example, while drawing an alphabet letter, the dimensionality of the 3D Cartesian path can be reduced to a 2D trajectory. By projecting the dataset on the drawing plane defined by two eigenvectors, the trajectory is then described by 2 signals (the 3rd eigenvectors is not used to reconstruct the dataset). Similarly, when reaching for an object, the joint angle trajectory of the shoulder is correlated with the joint angle trajectory of the elbow, and, thus, the shoulder and elbow trajectories could be expressed by only one signal.

3.2 Encoding in Hidden Markov Models (HMM)

For each gesture, a set of time series $\{\xi_1^x, \xi_2^x, \ldots, \xi_I^x, \xi_1^\theta, \xi_2^\theta, \ldots, \xi_J^\theta\}$ is used to train a Hidden Markov Model with $I + J$ output variables. The parameters are expressed as a set of parameters $\{\pi, A, \mu^\theta, \mu^x, \sigma^\theta, \sigma^x\}$, representing respectively the initial states distribution, the states transition probabilities, the means of the output variables, and the standard deviations of the output variables [1]. For each state, each output variable is described by a Gaussian, i.e. $p(\xi^{\theta_i}|\mu^{\theta_i}, \sigma^{\theta_i}) = \mathcal{N}(\mu^{\theta_i}, \sigma^{\theta_i}) \forall i = 1 \ldots 4$ and $p(\xi^{x_i}|\mu^{x_i}, \sigma^{x_i}) = \mathcal{N}(\mu^{x_i}, \sigma^{x_i}) \forall i = 1 \ldots 3$.

Continuous HMMs are used to encode the data with a parametric description of the distributions. A single Gaussian is assumed to approximate sufficiently each output variable (see Figure 4). A *mixture of Gaussian* could approximate any shape of distribution. However, it is not useful in our system, since the training is performed with too few training data to generate an accurate model of distribution with more than one Gaussian.

[1] People unfamiliar with HMM should refer to [18]

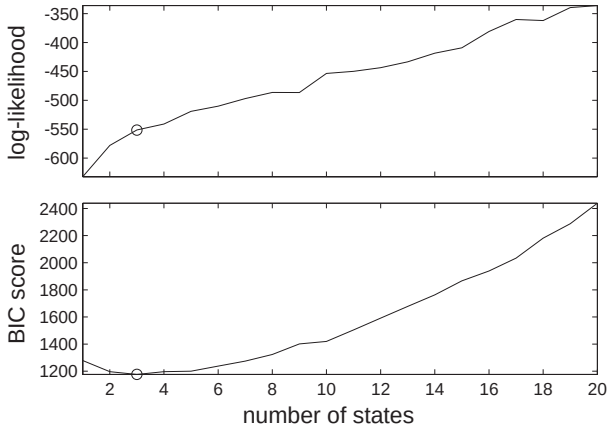

Fig. 5. A BIC criterion is used to determine the optimal number of states of the HMM, required to encode the data. *Top*: log-likelihood of the HMM according to the number of states used to represent the data. *Bottom*: the minimum BIC score gives a criterion to select the minimal number of states required to represent the data. It finds a trade-off between maximizing the likelihood of the model and minimizing the number of parameters used to model the data. Here, the gesture *waving goodbye* is modeled optimally with only 3 states.

The transition probabilities $P(q_t{=}j|q_{t-1}{=}i)$ and the observation distributions $p(\xi_t|q_t{=}i)$ are estimated by Baum-Welch, an *Expectation-Maximization* algorithm, that maximizes the likelihood that the training dataset can be generated by the corresponding model. The optimal number of states in the HMM may not be known beforehand. The number of states can be selected by using a criterion that weights the model fit (i.e. how well the model fits the data) with the economy of parameters (i.e the number of states used to encode the data). In our system, the Bayesian Information Criterion (BIC) [21] is used to select an optimal number of states for the model:

$$BIC = -2\ log(L) + n_p\ log(T) \tag{1}$$

The first term is used for the model fit, with L the likelihood of the fitted model. The second term is a penalty term, with n_p the number of independent parameters in the HMM, and T the number of observation data used in fitting the model. Data are encoded in different HMMs from one state to 20 states, and the model with the minimum score is retained (see Figure 5).

3.3 Recognition

Once trained, the HMM can be used to recognize whether a new gesture is similar to the ones encoded in the model. For each of the HMM, we run the *forward-algorithm* to estimate the likelihood that the new signals could have

been generated by one of the models. A measure of distance across two model's predictions is compared to a model-dependent threshold, to guarantee that the gesture is close to a model, but far enough from the others to be considered as recognized (see [7] for details).

3.4 Data retrieval

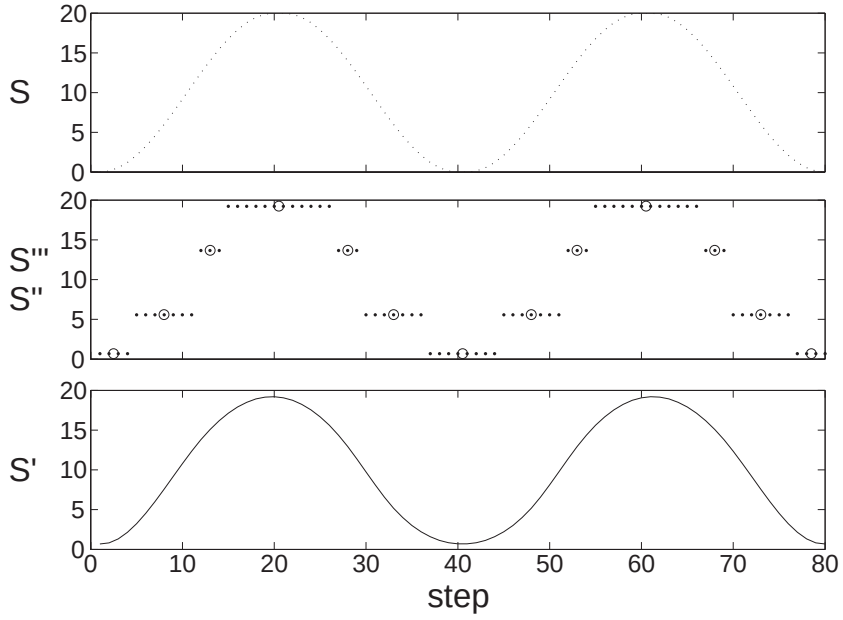

Fig. 6. Example of the retrieval process. The original signal S (*dotted-line*) is encoded in a HMM with 4 states. A sequence of states and corresponding output variables S''' are retrieved by the Viterbi algorithm (*points*). Keypoints S'' are defined from this sequence of output variables (*circles*). The retrieved signal S' (*straight-line*) is then computed by interpolating between the keypoints and normalizing in time.

When a gesture is recognized by a HMM, a generalization of the gesture is reproduced. Given the observation of the gesture and the parameters $\{\pi, A, \mu, \sigma\}$ of the HMM, a sequence of states is reconstructed by the Viterbi algorithm. Given this sequence of states, the output variables $\{\xi_1'''^x, \xi_2'''^x, \ldots, \xi_I'''^x, \xi_1'''^\theta, \xi_2'''^\theta, \ldots, \xi_J'''^\theta\}$ are retrieved, by taking the mean value μ of the Gaussian distribution for each output variable.

Keypoints $\{\xi_1''^x, \xi_2''^x, \ldots, \xi_I''^x, \xi_1''^\theta, \xi_2''^\theta, \ldots, \xi_J''^\theta\}$ are then extracted from these time series. If there is a transition to a state n at time t_1 and if there is a transition to another state at time t_2, a keypoint is created at the mean time $\frac{t_1+t_2}{2}$. By interpolating between these key-points and normalizing in time, the output

variables $\{\xi_1'^x, \xi_2'^x, \ldots, \xi_I'^x, \xi_1'^\theta, \xi_2'^\theta, \ldots, \xi_J'^\theta\}$ are reconstructed (see Figure 6). Finally, by using the eigenvectors found by PCA, the whole hand path $\{x_1', x_2', x_3'\}$ and joint angle trajectories $\{\theta_1', \theta_2', \theta_3', \theta_4'\}$ are reconstructed.

3.5 Determining the task constraints

In [3], [7] and [8], we have developed a general formalism for determining the metric of imitation performance. The metric measures the quality of the reproduction, and, as such, drives the selection of an appropriate controller for the reproduction of the task.

One way to compare the relative importance of each set of variables (i.e. joint angles, hand path) in our experiment is to look at their variability. Here, we take the perspective that the relevant features of the movement, i.e. those to imitate, are the features that appear most frequently, i.e. the invariants in time, and apply the metric to determine the relevance of the Cartesian and joint angle representation to reproduce a gesture.

Following this framework, we model the task's cost function as a weighted linear combination of metrics applied to the joint angle trajectories and the hand path.

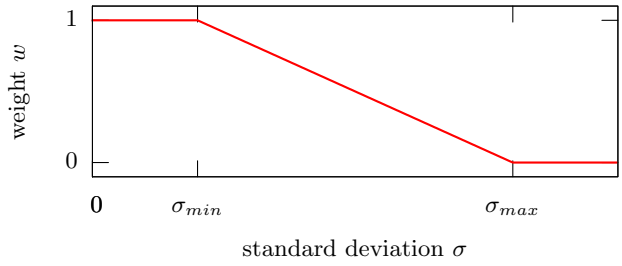

Fig. 7. Function used to transform a standard deviation σ to a weight factor $w \in [0, 1]$. σ_{min} corresponds to the accuracy of the sensors. σ_{max} represents the maximal standard deviation measured during a set of demonstrations generated by moving randomly the arms during one minute.

Unidimensional case: Let $D = \{x_1, x_2, \ldots, x_T\}$ and $D' = \{x_1', x_2', \ldots, x_T'\}$ be the demonstration and the reproduction datasets of a variable x. The cost function J is defined by:

$$J(D, D') = 1 - f(e) \tag{2}$$

$J \in [0, 1]$ calculates an estimation of the quality of the reproduction, using two different metrics. Optimizing the imitation consists of minimizing J

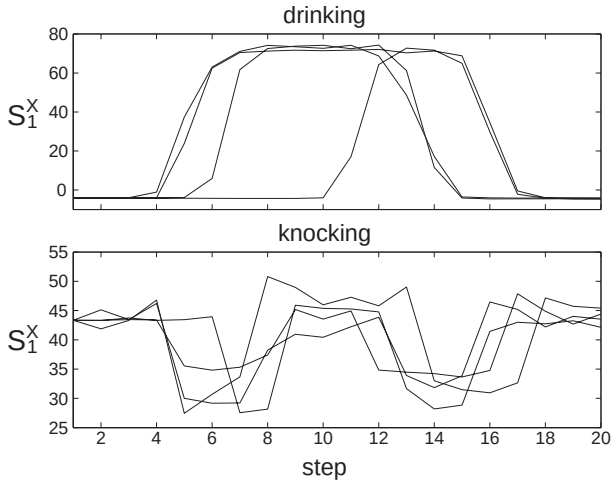

Fig. 8. 4 demonstrations of the *drinking* gesture and the *knocking* gesture (only one variable from the visual information is represented). Even if the trajectories are rescaled in time, data do not overlap, because they present non-homogeneous distortions in time. By encoding the data in HMM, it is still possible to distinguish very well the two datasets.

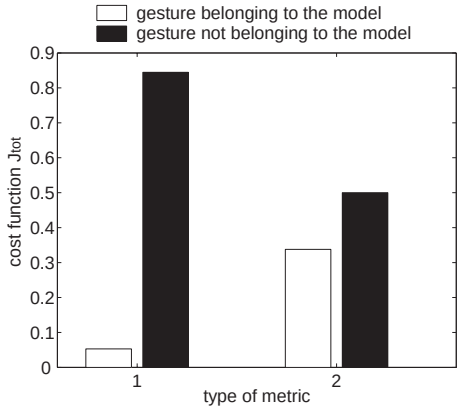

Fig. 9. Comparison of two metrics to evaluate the quality of a reproduced trajectory. *Left:* using the error measure e based on HMM encoding of the dataset. *Right:* using the RMS error e' with the trajectories rescaled in time. The white and black bar corresponds respectively to the data belonging to the model, and not belonging to the model. The error based on HMM encoding discriminates better the 2 datasets.

(J=0 corresponds to a perfect reproduction). e is a measure of distance across the observed data D' and the training data D. A transformation function $f()$ normalizes and bounds each variable within minimal and maximal values (see Figure 7). This results in the elimination of the effect of the noise, intrinsic to each variable, so that the relative importance of each variable can be compared.

The metric uses the HMM representation of the data to compute the error value e, robust to distortion in time. The *Viterbi algorithm* is first used to retrieve the best sequence of states $\{q_1, q_2, \ldots, q_T\}$, given the observation data $D' = \{x'_1, x'_2, \ldots, x'_T\}$ of length T. If $\{\mu_1, \mu_2, \ldots, \mu_T\}$ is the sequence of means associated with the sequence of states, we define:

$$e = \frac{1}{T} \sum_{t=1}^{T} |x'_t - \mu_t| \tag{3}$$

We have compared this error measure to the most commonly used *root mean square* (RMS) error, calculated with signals rescaled in time, using the dataset shown in Figure 8. The RMS error is computed as:

$$e' = \frac{1}{T} \sum_{t=1}^{T} |x'_t - x_t| \tag{4}$$

The results of the metrics calculated using e or e' are presented in Figure 8. Each data has been tested with the two models, and should produce respectively a low value of J if they belong to the corresponding model, and a high value if they do not. The metric using the HMM representation of the time-series gives better results than the one using the static error computed on rescaled signals (see Figure 9). Indeed, HMM can deal with the distortions in time in the 2 datasets.

Multidimensional case: When data have K dimensions, the metric J_{tot} is expressed as:

$$J_{tot} = \frac{1}{K} \sum_{i=1}^{K} w_i \, J(D_i, D'_i) \tag{5}$$

$w_i \in [0, 1]$ weight the importance of each set of variables. These factors are extracted from the demonstration and reflect the variance of the data during the demonstration. To evaluate this variability, we also use the statistical representation provided by the HMM. The *Viterbi algorithm* is used to retrieve the best sequence of states $\{q_1, q_2, \ldots, q_T\}$, given the observation data D'. If $\{\sigma_1^i, \sigma_2^i, \ldots, \sigma_T^i\}$ is the sequence of standard deviations of variable i, associated with the sequence of states, we define:

$$w_i = f(\frac{1}{T}\sum_{t=1}^{T}\sigma_t^i) \tag{6}$$

If the variance of a given variable is high, i.e. showing no consistency across demonstrations, then, satisfying some particular instance of this variable will have little bearing on the task. The factors w_i in the cost function equation reflect this assumption: if the standard deviation of a given variable is low, the value taken by the corresponding w_i are close to 1. This way, the corresponding variable will have a strong influence in the reproduction of the task.

A mean standard deviation is thus calculated over the whole path, and is transformed by a function (see Figure 7) to give a weight $w_i \in [0, 1]$ to estimate the relevance of dataset i.

w_i can then motivate the use of either a direct joint angles controller or an inverse kinematics controller. In order to use both controllers simultaneously, one can extend the inverse kinematics solution to encapsulate constraints on the joint angles, as in [8].

Since the demonstrator and the robot do not share the same embodiment (they differ in the length of their arms and in the range of motion of each DOF), there is a *correspondence problem* [15]. Here, this problem is solved by hands. The joint angle trajectories of the demonstrator are automatically shifted and scaled, when required, to ensure that these fit within the range of motion of the robot.

4 Results and performance of the system

The training set consists of the joint angle trajectories and hand path of 4 subjects performing the 6 different motions. The test set consists of 4 other subjects performing the same 6 motions, with only the hand path trajectories, to demonstrate the recognition ability of the model even in the face of missing data. Once a gesture has been recognized by the model based on the hand path only, the complete joint angle trajectories can be retrieved.

23 motions have been recognized correctly (recognition rate of 96%). The only error has happened for one instance of the *knocking on a door* motion, that has been confused with the *waving goodbye* motion. This is not surprising. Indeed, when projecting these two motions on the first principal component of their respective model, one observes that the resulting trajectories are quite similar (both motions involve a principal oscillatory component). Thus, it can be difficult to classify them correctly using exclusively the time series after projection. The robustness of the system could be improved by comparing the principal components extracted from the test set with the ones extracted from the training set, in combination with the HMM classification. The drawback is that the system would not be able to recognize a similar gesture performed in a different situation. For example, the principal components of an alphabet letter are not

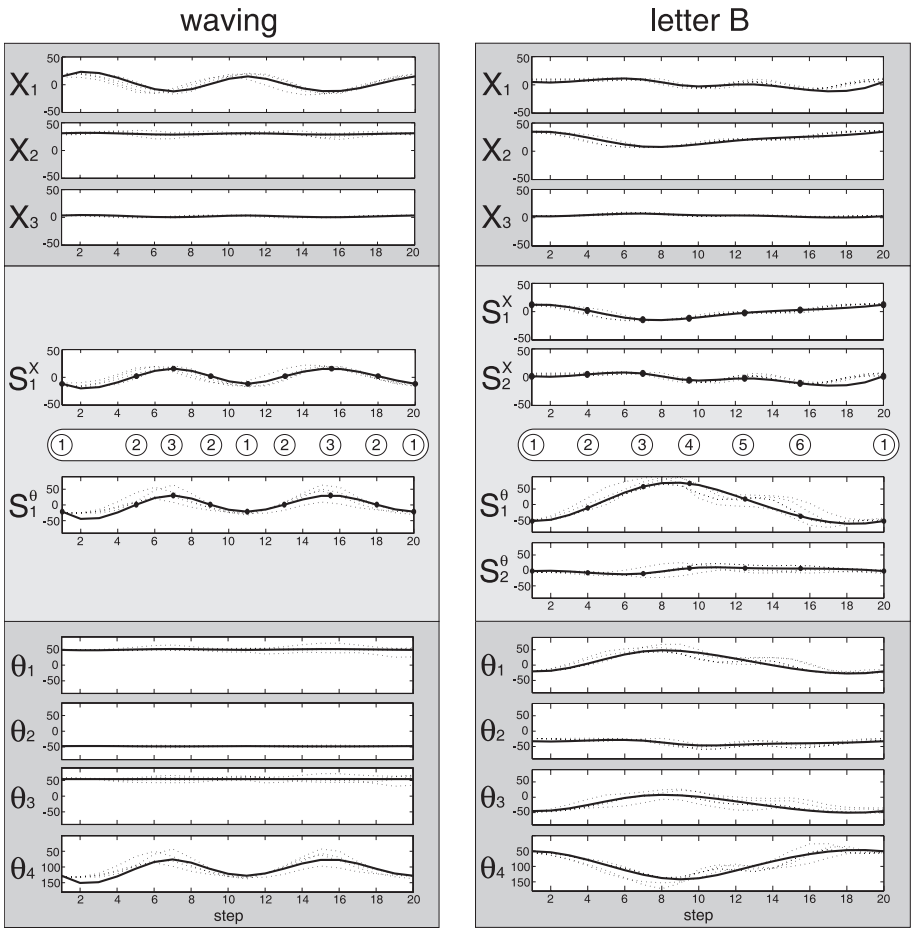

Fig. 10. Demonstration, transformation by PCA, encoding in HMM and retrieval of the 2 different motions *waving goodbye* and *drawing letter B*. The 5 demonstrations in visual coordinates $\{x_1, x_2, x_3\}$ and motor coordinates $\{\theta_1, \theta_2, \theta_3, \theta_4\}$ are represented by dotted lines. The retrieved generalized trajectory is in bold line.

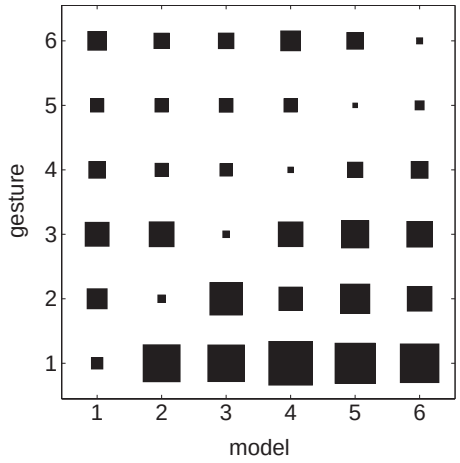

Fig. 11. Cost function J_{tot}, when testing the gestures of the test set with the different HMMs (the size of the square is proportional to J_{tot}). Each row corresponds to a specific gesture: *1)* drawing letter A, *2)* drawing letter B, *3)* drawing letter C, *4)* waving goodbye, *5)* knocking on a door, *6)* drinking. These gestures are tested with the 6 corresponding models. For each row, the column with lowest value indicates what is the best model corresponding to the gesture.

the same if the user writes it on a table or on a blackboard. After projection, the resulting signals are however similar, and can be recognized by HMM.

Figure 1, 2 and 10 shows the encoding and decoding of the 6 motions. As expected, 2 principal components are sufficient to represent the hand path when drawing each of the 3 letters, as well as when performing the *knocking* gesture. For *waving* and *drinking*, a single component is sufficient. Consistently, 2 principal components are sufficient to represent the joint trajectories when drawing the 3 letters of the alphabet, while only a single component is required to represent the gestures of *waving*, *knocking* and *drinking*.

The resulting signals for *letter A*, *waving*, *knocking* and *drinking* are modeled by a HMM of 3 states. *letter B* is modeled with 6 states, and *letter C* with 4 states. The keypoints in the trajectories correspond roughly to inflexion points or relevant points describing the motion. The number of states found by the BIC criterion grows with the complexity of the signals to model.

Figure 11 represents the values of the cost function J, when testing the gestures of the test set with the different HMM models. The weights w_i found by the system are quite always similar for the motor and visual representations, which means that both representations could be used to reproduce the motion, with a slight preference to the visual representation. Indeed, we see on Figure 10 that there is not so much difference between the variations of the signals in both representations. It can be due to the experimental setup, where the motion of the users are recorded only in one situation. In the next experiments, different

situations or environments should be used to provide more variations in one or the other dataset.

5 Discussion on the model

The combination of PCA and HMM is used successfully in our application to reduce the dimensionality of a dataset and to extract the primitives of a motion. Preprocessing of the data using PCA removes the noise and reduces the dimensionality of the dataset, making the HMM encoding more robust. The parameters of the whole model are then $\{v^x, v^\theta, \bar{x}, \bar{x}, \pi, A, \mu^\theta, \mu^x, \sigma^\theta, \sigma^x\}$. This requires less parameters than HMM encoding of the raw data (see [13, 7]), as the number of output values and number of states are optimally reduced.

The advantage of encoding the signals in HMMs, instead of using a static clustering technique to recognize the signals retrieved by PCA, is that it provides a better generalization of the data, with an efficient representation, robust to distortion in time. As an example, let us consider a situation, where two demonstrators A and B raise a glass at the same speed, drink with different speed, and put the glass back on a table simultaneously. In HMM, the difference in amplitude are fitted by a Gaussian, for each state and each variable. The distortions in time are handled by using a probabilistic description of the transitions between the states, while a simple normalization in time would not have generalized correctly over the 2 demonstrations.

The model is general in the sense that no information concerning the data is encapsulated in the PCA preprocessing or in the HMM classification, which makes no assumption on the form of the dataset. However, extracting the statistical regularities is not the only mean of identifying the relevant features in a task, and it would probably not allow learning of a more complicated task. In further work, we will exploit the use of other machine learning techniques to extract the optimal representation. In addition, we will consider the use of explicit pointers (e.g. speech) in combination to statistics, in order to extract the key-features more robustly and more efficiently.

Finally, it would be interesting to extend the model to using *asynchronous HMM*. Such models have been exploited successfully in speech processing to model the joint probability of pairs of asynchronous sequences describing the same sequence of events (e.g. visual lip reading and audio signals) [2]. It could be used in our application to learn and retrieve the best alignment between two sequences in visual and motor representations. It can be useful since the datasets are not always synchronized, but still needs to have correspondence between the different representations.

5.1 Similarity with works in Psychology and Ethology

Some of the features of our model bear similarities with those of theoretical models of animal imitation. These models often assume that data are correctly discretized, segmented, and classified. By using PCA and HMM to encode the

information, it is still possible to keep the elements of these frameworks that are relevant to a robotic system, offering at the same time a probabilistic description of the data, more suitable for a real-world application.

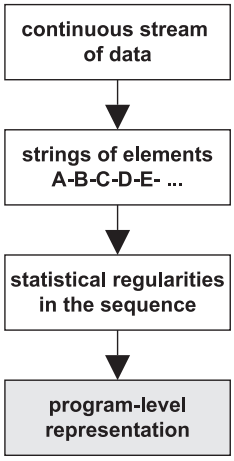

Fig. 12. R.W. Byrne's string parsing imitation model (schema inspired from [6])

Imitation using String Parsing R.W. Byrne has suggested to study the cognitive processes underlying animal's imitation by using *String Parsing* to segment a continuous task into basic elements [6]. Imitating a task with a sequential or hierarchical organization of basic actions have been observed in species as diverse as rats and apes, to learn complex feeding skills. The process requires an effective segmentation of the elements, so that imitation learning becomes a practical method to acquire more complex skills from basic elements. Such a segmentation allows to reuse known features, and extract the underlying structure in the observed behavior. If the whole task is perceived as a discrete sequence of items, the statistical regularities can be extracted, and the hierarchical structure is discovered by observing several times the same task.

Similarly, in a robotic system using PCA and HMM, the structure that underlies a sequence of elements can be acquired statistically by observing regularities across multiple demonstrations. Moreover, in HMM learning algorithm, a discrete set of key-features is extracted from a continuous flow of motion data, and the sequential organization of the key-features is learned by the model. The structure of the observed behavior is described probabilistically by transition probabilities between the key-features. By using a fully-connected model, it is thus possible to extract statistically the sequential structure of a task, with recurring elements and optional parts that are not always needed.

In our implementation, HMMs are used to find the key-features in the trajectories, and learn gestures by extracting the regularities in the sequences. Two

concurrent stochastic processes are involved, one modeling the sequential structure of the data, and one modeling the local properties of the data.

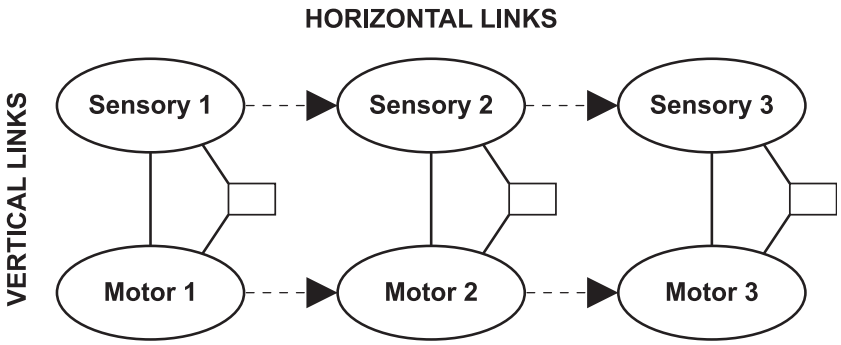

Fig. 13. C.M. Heyes and E.D. Ray's Associative Sequence Learning (ASL) model (schema inspired from [11])

Associative Sequence Learning (ASL) C.M. Heyes and E.D. Ray's *Associative Sequence Learning* (ASL) mechanism [12, 11] suggests that imitation requires a *vertical association* between a model's action, as viewed from the imitator's point of view, and the corresponding imitator's action. The vertical links between the sensory representation of the observed task and the motor representation are part of a repertoire, where elements can be added or refined. ASL suggests that the links are created essentially by experience, with a concurrent activation of sensory and motor representations. A few of them can also be innate, as biological data seem to indicate. The model pictures that the mapping between the sensory and motor representation can be associated with a higher level representation (boxes depicted in Figure 13).

The horizontal links model the successive activation of sensory inputs to learn the new task, that activates at the same time the corresponding motor representation, to copy the observed behavior. Repetitive activation of the same sequence strengthens the links to help motor learning. Depending on the complexity of task organization, numerous demonstrations may be needed to provide sufficient data to extract the regularities. The more data are available, the more evident is the underlying structure of the task, to clarify which elements are essential, which are optionals, and which are variations in response to changing circumstances.

This stresses the need of a probabilistic framework in a robotic application that can extract invariants across multiple demonstrations. Such a model is in agreement with a HMM decomposition of the task, where the underlying structure is learned by using multiple demonstrations. If a given pattern appears frequently, its corresponding transition links are strengthen. If each action

perceived by the imitator is also coded as an action that it can execute, the reproduction of a task can be considered. Any action that the imitator is able to perform can then also be recognized by observation of a model demonstrating the task.

Each hidden state in a HMM can output multimodal data. It can thus model multiple variables in different frames of reference. Its role is to make a link between the different datasets, and can be considered as a label or as a higher level representations common to the visual and motor data (see Figure 4). Indeed, in HMM, the sequence of states is not observed directly, and generates the visual or motor representation.

Thus, the architecture of a multivariate HMM has a horizontal process to associate the elements in a sequential order, by learning transition probabilities between the hidden states, and a vertical process that associates each sensory representation to appropriate motor representation, which is done through the hidden state. If data are missing from one part or the other (visual or motor representation), it is still possible to recognize a task, and retrieve a generalization of the task in the other representation, if required.

By using a similar representation of the ASL model in Figure 13, our system focus on learning the *horizontal links*. The *vertical associations* represented through the hidden states, are still hard-coded, specifying prior knowledge on the robot architecture. It involves a simple rescaling of the joint angles to fit the range of motion of the robot.

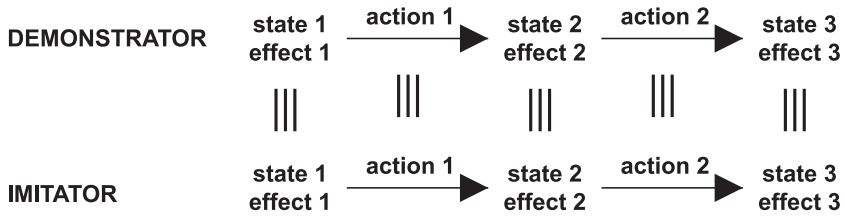

Fig. 14. C.L. Nehaniv and K. Dautenhahn's algebraic framework to map states, effects and/or actions of the demonstrator and the imitator (schema inspired from [17])

Algebraic framework for the correspondence problem The *correspondence problem* refers to the problem of creating an appropriate mapping between what is performed by the demonstrator and what is reproduced by the imitator. The two agents may not share the same embodiments (e.g. difference in limb lengths, sensors or actuators). Such correspondences can be constructed at various levels of granularity, reflecting the choice of a sequence of subgoals. C.L. Nehaniv and K. Dautenhahn have suggested a general algebraic framework [17, 16], to address the matching problem in both natural and artificial systems, and interpret the ASL model in this framework. It consists of representing the behavior performed by the demonstrator and imitator as two automata structures,

with states and transitions. The *states* are the basic elements segmented from the whole task, that can produce *effects*, i.e. responses in the environment (e.g. object displaced). An *action* is the transition from one state to another state.

An imitation process is defined as a partial mapping process (relational homomorphism) between the demonstrator and imitator *states*, *effects*, and *actions*. An observer (e.g. external observer, demonstrator or imitator) decides which of the *states*, *actions* or *effects* are the most important ones to imitate, by fixing an imitation metric. Different metrics are used to yield solutions to different correspondence problems. These metrics also allow to formally quantify the success of an imitation.

This notation is closely related to the one used in our system. A HMM is an extension of the automata depicted in this algebraic framework. The difference is that these automata are described stochastically, and are thus more suitable to be used with noisy data, in a real-world application. Each hidden state outputs probabilistically distributed values that can be seen as *effects*, and the equivalent of *actions* are the transitions between hidden states, also probabilistically defined. Note that encoding the data in HMM does not resolve the correspondence problem, but gives a suitable framework for its representation, to treat the *what-to-imitate* and the correspondence problem in a common framework.

In further research, our work will address the correspondence problem paradigm [1, 16, 17, 15]. The contribution in our system would be to ensure a robust mapping between the two agents. In Alissandrakis et al work, an external observer decides which mappings are relevant to the imitation, i.e decides which of the states, the actions or the effects should be mapped. The similarity is measured by observer-dependent metrics. The contribution of our work would be to extract the relevant features of a task from the statistical regularities across the multiple demonstrations, instead of specifying them in advance.

6 Conclusion

This chapter has presented an implementation of a HMM-based system to encode, generalize, recognize and reproduce gestures, with representation of the data in visual and motor coordinates. The model has been tested and validated in a humanoid robot, using kinematics data of human motion.

The framework offers a stochastic method to model the process underlying gesture imitation. It makes a link between theoretical concepts and practical applications. In particular, it stresses the fact that the observed elements of a demonstration, and the organization of these elements should be stochastically described to have a robust robot application, that takes account of the high variability and the discrepancies across demonstrator and imitator points of view.

Acknowledgments

Thanks to the undergraduate students C. Dubach and V. Hentsch who helped to develop the motion tracking with x-sens and vision. This work was supported in part by the Swiss National Science Foundation, through grant 620-066127 of the SNF Professorships program, and 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.

References

1. A. Alissandrakis, C.L. Nehaniv, and K. Dautenhahn. Imitating with alice: Learning to imitate corresponding actions across dissimilar embodiments. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 32(4):482–496, 2002.
2. S. Bengio. Multimodal speech processing using asynchronous hidden markov models. *Information Fusion*, 5(2):81–89, 2004.
3. A. Billard, Y. Epars, S. Calinon, G. Cheng, and S. Schaal. Discovering optimal imitation strategies. *Robotics and Autonomous Systems*, 47(2-3):69–77, 2004.
4. A. Billard and G. Hayes. Drama, a connectionist architecture for control and learning in autonomous robots. *Adaptive Behavior*, 7(1):35–64, 1999.
5. A. Billard and R. Siegwart. Robot learning from demonstration. *Robotics and Autonomous Systems*, 47(2-3):65–67, 2004.
6. R.W. Byrne. Imitation without intentionality. using string parsing to copy the organization of behaviour. *Animal Cognition*, 2:63–72, 1999.
7. S. Calinon and A. Billard. Stochastic gesture production and recognition model for a humanoid robot. In *Proceedings of the IEEE/RSJ Intl Conference on Intelligent Robots and Systems (IROS)*, pages 2769–2774, Sendai, Japan, September 28 - October 2 2004.
8. S. Calinon, F. Guenter, and A. Billard. Goal-directed imitation in a humanoid robot. In *Proceedings of the IEEE Intl Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, April 18-22 2005.
9. J. Demiris and G. Hayes. *Imitation as a Dual-Route Process Featuring Predictive and Learning Components: A Biologically-Plausible Computational Model*, chapter 13, pages 327–361. MIT Press, c. nehaniv and k. dautenhahn edition, 2001.
10. R. Dillmann. Teaching and learning of robot tasks via observation of human performance. *Robotics and Autonomous Systems*, 47(2-3):109–116, 2004.
11. C.M. Heyes. Causes and consequences of imitation. *Trends in Cognitive Sciences*, 5:253–261, 2001.
12. C.M. Heyes and E.D. Ray. What is the significance of imitation in animals? *Advances in the Study of Behavior*, 29:215–245, 2000.
13. T. Inamura, I. Toshima, and Y. Nakamura. *Acquiring Motion Elements for Bidirectional Computation of Motion Recognition and Generation*, volume 5, pages 372–381. Springer-Verlag, b. siciliano and p. dario edition, 2003.
14. I.T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 1986.
15. C. Nehaniv and K. Dautenhahn. *Of Hummingbirds and Helicopters: An Algebraic Framework for Interdisciplinary Studies of Imitation and Its Applications*, volume 24, pages 136–161. World Scientific Press, j. demiris and a. birk edition, 2000.

16. C.L. Nehaniv and K. Dautenhahn. Like me? - measures of correspondence and imitation. *Cybernetics & Systems: An International Journal*, 32(1-2):11–51, 2001.
17. C.L. Nehaniv and K. Dautenhahn. *The correspondence problem*, pages 41–61. MIT Press, 2002.
18. L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, February 1989.
19. S. Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242, 1999.
20. S. Schaal, A. Ijspeert, and A Billard. Computational approaches to motor learning by imitation. *Philosophical Transaction of the Royal Society of London: Series B, Biological Sciences*, 358(1431):537–547, 2003.
21. G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.