FP6-IST-002020

COGNIRON

The Cognitive Robot Companion

Integrated Project

Information Society Technologies Priority

D4.1.1
Evaluation of subgoals extraction algorithm on kinematics data of human motion and on objects displacement

Due date of deliverable: 31/12/2004
Actual submission date: 14/1/2005

Start date of project: January 1st, 2004 **Duration:** 48 months

Organisation name of lead contractor for this deliverable:
EPFL

Revision: final
Dissemination Level: PU

Executive Summary

This report gives a brief overview of the work developped within the first 12 month of RA4, and within the framework of the workpackage WP4.1. Technical details concerning the implementation and the mathematical foundation of the learning architecture, as well as most substanciated references to other approaches to imitation learning can be found in the peer-reviewed publications, given in the references, and which appeared this year as part of the work conducted within this workpackage. This report is complemented by a video of an implementation of the system to reproduce an influencial experiment of developmental psychology on children imitation [1], see the technical description in Section 2.
For recall, workpackage WP4.1 aims at deriving a general policy to drive robot learning by imitation. The goal of the research reported in the present document is three-fold: 1) to determine the constraints of the demonstrated task; 2) to formulate the task metric depending on the importance of the task constraints; 3) to find the optimal controller to imitate the task given the metric. Note that, in 3 above, we consider a situation in which there is no *correspondence problem*, i.e. demonstrator and imitator have the same number of degrees of freedom and it is possible to define a subpart of the workspace in which the imitator's motions can match perfectly that of the demonstrator given the metric. The correspondence problem is addressed separately in deliverable D.4.2.1, and as part of WP4.2.

Role of (topic of deliverable) in Cogniron

Learning from observing and reproducing human actions is fundamental to ensure that the robot could adapt to different environments and users, as well as to enable the robot's ability for long-life acquisition of complex skills. While prior work in the domain assumed that the task representation was known and was best described by a predefined set of key features, we here address the issue of how to learn the optimal representation of a given task.

Relation to the Key Experiments

This research will contribute to the KE3 script *Learning Skills: Arranging and Interacting with Objects*. The script stressed the issue of generalizing over a number of demonstrated tasks. Specifically, the robot must learn to recognize and reproduce a variety of gestures, and, by so doing must learn ways by which it can interact and perform simple manipulations on objects. The research reported here and conducted as part of WP4.1 will provide algorithms for enabling the robot to learn the essence of the demonstrated gestures (Cogniron function CF-RG *Learning to reproduce gestures*), see Section 1.2.3, and to extract the key relationships across gesture and object motions (Cogniron function *CF-LIG: Learning Important Features of a Task*), see Sections 1.2.1 and 1.2.2.

1 Evaluation of subgoals extraction algorithm on kinematics data of human motion and on objects displacement

This section is divided as follows: Section 1.1 introduces briefly the novelty of the approach to imitation learning with respect to the current literature. In Section 1.2, we first describe the general formalism underlying the approach. In Section 2, we briefly summarize the application of the approach in a set of robotics experiments. The reader should refer to publications [2, 3, 4] for all technical details.

1.1 Approach to Robot Programming by Demonstration

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 [2, 5, 6]. 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 [7, 8, 9]. 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.

Robots programming by demonstration has, by now, become a key topic of research in robotics (see [10] for a recent overview of core approaches in the domain). Work in that area tackles the development of robust algorithms for motor control, motor learning, gestures recognition and visuo-motor integration.

Two core issues of imitation learning are known as *"what to imitate"* and *"how to imitate"* [11]. *What to imitate* refers to the problem of determining which of the features of the demonstration are relevant for the achievement of the task [2]. This is issue is the core of the WP4.1. *How to imitate*, also referred to as the *correspondence problem* [9], is the problem of transferring an observed motion into one's own capabilities. Works tackling this issue have followed either an approach in which the correspondence is unique and the imitation must produce an exact, but parameterizable, reproduction of the trajectories [12, 13, 7], or an approach in which only a subset of predefined goals must be reproduced (e.g. [5, 14, 15, 16]). The "how to imitate" issue is addressed as part of WP4.2 and is complementary to the "what to imitate" issue.

While prior work has concentrated on either of these issues separately, in WP4.1, we take an approach in which we combines a method for solving the *what to imitate* problem by extracting the task constraints, with a method for solving the *how to imitate* problem given a set of task constraints. The present document presents the theoretical framework we develop for solving the *what to imitate* problem [2], in incorporating the notion of goal preference and including a method for optimizing the reproduction (*how to imitate*). The later step links the work conducted as part of WP4.1 to work conducted as part of WP4.2.

1.2 Formalism

Let D be the dataset generated by the demonstrator while driven by a controller U. U is such that $D(U) = \{\vec{X}, \vec{X^0}, \vec{\Theta}\}$, where, in the case considered here, $\vec{X} = \{x, \dot{x}, \ddot{x}\}$ and $\vec{X^O} = \{x^O, \dot{x^O}, \ddot{x^O}\}$ (3-dim Cartesian position, speed and acceleration), are the Cartesian trajectories of the hand and the object respectively, and $\vec{\theta} = \{\theta, \dot{\theta}, \ddot{\theta}\}$ (angular position, speed and acceleration) the trajectory of the demonstrator's arm joints.

The imitation process consists, then, of determining a controller U', that generates a dataset $D'(U') = \{\vec{X'}, \vec{X'_0}, \vec{\theta'}\}$, such that J, the *cost function* or the *metric* of the imitation task, is minimal: $\delta J(D, D') = 0$.

Each demonstrated task is defined by a set of constraints $s = 1, ..., S$. For each constraint s, $\exists$ a controller U_s generating a dataset D_{U_s}, such that the associated metric J_{U_s} is minimal: $\delta J_{U_s}(D_{U_s}, D'_{U_s}) = 0$.

Let be an imitation task in which the demonstrator performs a number N of variants of the task. While observing the N demonstrations, the imitator computes the probability $P(s)$ that the demonstrator tried to satisfy the constraint s. Given a set of likely constraints $1, .., s$, the imitator computes the optimal combination of controllers U'_s that satisfy all constraints.

1.2.1 Learning the constraints

If the task's constraints are unknown, these must be learned. We hypothesize that the task constraints consist of all the **invariants** and **correlations** across the data of the dataset D, and we propose to determine those by evaluating the probability distribution of all variables of the dataset.

Invariants
Let X be a variable generated by the distribution $P(X = x)$. Let $\{x_n\}, n = 1, ..N$ be the N observations of X during the demonstrations. x_0 is an invariant of X if $P(X = x_0) = 1$. The task constraint is, thus, $x_n = x_0, \forall n$ and the cost function is expressed as $J(x, x') = \sum_{n=1}^{N} \|x'_n - x_n\| = \|x'_n - x_o\|$.

Correlations
Let X and Y be two variables generated by the distributions $P(X = x)$ and $P(Y = y)$. Let $\{x_n\}, n = 1, ..N$ and $\{y_m\}, m = 1, ..M$ be the N and M observations of X and Y respectively.

X and Y are correlated if $\exists i$, such that $P(X = x_i | Y = y_i) = P(X = x_i) = P(Y = y_i)$. Such a correlation can be found by looking at the covariance table of X and Y, since $cov(x_i, y_i) = 0$.

If this correlation applies to the time interval $[k, .., K]$, x can be expressed as a function f of Y in that time interval. The constraint representing this correlation becomes, then, $x_k = f(y_k)$ and the associated cost function is $J(x', y) = \sum_{i=k}^{K} \|x'_i - f(y_i)\|$.

Therefore, learning the task correlations consists of, first, determining the interval within which there is correlation (looking at the covariance matrix) and, then, of determining the correlation function f. For the latter, one can use several methods from Machine learning. For instance, in order to

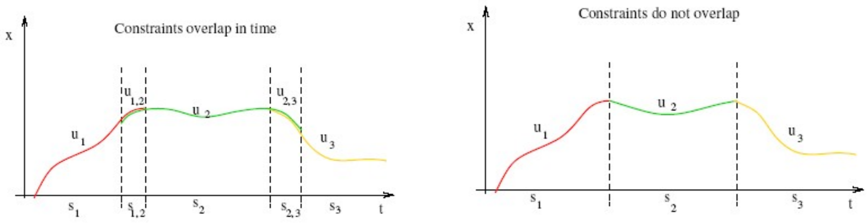

Figure 1: Illustration of a task that combines 3 controllers u_s, satisfying the constraints $s = 1, .., 3$ that do overlap in time (*left*), and that do not overlap in time (*right*).

look for constraints in the temporal precedence across the data (time series), one can use Hidden Markov Models (HMM) or Time Delay Neural Networks (TDNN); while, in order to look at correlated patterns without temporal correlation, one could use auto- or hetero-associative memory (Hopfield).

1.2.2 Learning the metric

Let us assume that the task is described by a known set $s = \{1, ..S\}$ of constraints with correlated cost function J_s.

We consider two cases:

1) If there are more than one constraint *at any given point of time*, the total metric or cost function is the sum of all constraint-based cost functions $J(t) = \sum_{s=1}^{S} w_s * J_s(t)$. Learning the metric consists, then, of determining the weight w_s of the metric; in their simplest form, these weights can be expressed as $w_s = P(s)$ the likelihood that the constraint s (invariant or correlation) has been observed in the dataset. This likelihood reflects the uncertainty associated with the measure of all the variables that define the constraint.

For instance, if the constraint is a spatio-temporal correlation such that $x = h(f(y))$, where f is the generative process modeled by an HMM and $h(z) = exp\frac{(a-z)^2}{b}$ is a Gaussian noise added to the output of the HMM, then, $P(x = f(y))$ is the uncertainty of the measure and is given by the log-likelihood of the HMM on a new set of measure. This log-likelihood models the Gaussian process h.

2) If there is only one constraint at a given time, but several constraints across the whole duration of the task i.e. $J(T) = \sum_{s=1}^{S} \sum_{t=t_s}^{t_{s+1}} J_s(t)$ with $S > 1$, learning the metric consists of determining the time intervals t_s during which each constraint is applied, and of determining whether the order in which constraints are satisfied matters, i.e. if the satisfaction of a given constraint is conditional to the satisfaction of other constraints.

1.2.3 Learning the controller

Let us assume that both the constraints and the metric are known, and that there exists a set $\{u_{s_l}\}$, with $l = 1, .., L$, of controller that satisfy the constraints $s = 1, .., S$ (there can be more than one controller that satisfy each constraint).
We consider two cases.
1) If there is only one constraint at a given time (see Figure 1), for each constraint s, we determine the

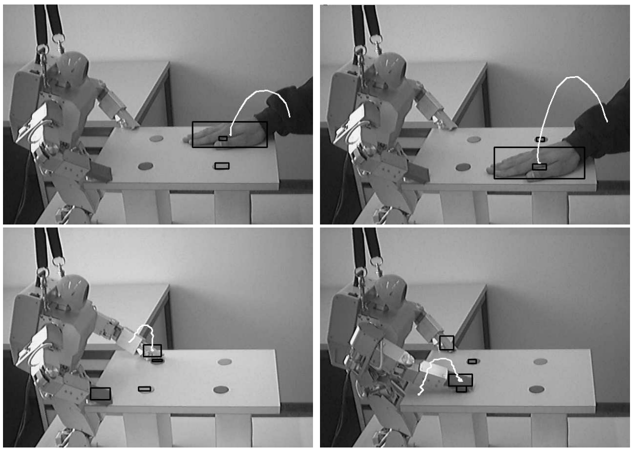

Figure 2: *Top:* demonstration of an ipsi- (*top-left*) and contralateral (*top-right*) motion of the right arm. *Bottom:* reproduction by the humanoid robot of the motion candidate with lowest cost function J. the robot reproduces the contralateral motion of the demonstrator by doing an ipsilateral motion with the other arm, that is closer to the dot to touch.

optimal controller u_{s_l} that satisfies the constraint.

Here is an example:

The controller $x = u_{s_l}(y)$ is a polynomial of order 2, such that $u_{s_l}(y) = a \cdot y + b \cdot y^2$ and the cost function is of the form $J_s(x, y) = (y - y_0) = (a \cdot y + b \cdot y^2 - y_0)$. The condition for optimality is $\delta J = \frac{\delta J}{\delta x} = 0 \Leftrightarrow \delta J = a + b \cdot x = 0 \Rightarrow a = ..; b = ...$

In order to ensure that the combination of all controllers gives a **continuous** output [1], we must add the following constraint:

$$\forall s \; \exists x, \text{s.}t.u_s(x) = u_{s+1}(x)$$

2) If there are several constraints co-occurring at the same time, we must determine the optimal way to combine the controllers in order to satisfy the complete metric J. In doing so, we must learn a new controller $u_{i,j}$ that satisfy two constraints i and j. *At this stage, I have no solution for this part; I will keep working on it .*

2 Implementation

2.1 Implementation for learning arbitrary gestures

The experiment starts with the (human) demonstrator and the (robot) imitator standing in front of a table, facing each other (see Figure 2). On both sides of the table, two colored dots (red and green) have been stamped at equal distance to the demonstrator and imitator's starting positions. In a first set of demonstrations, the demonstrator reaches for each dot alternatively with left and right arm. If the demonstrator reaches for the dot on the left handside of the table with his left arm, it is said to perform an ipsilateral motion. If conversely the demonstrator reaches the dot on the right handside of the table with his left arm, it is said to perform a contralateral motion. Then the demonstrator produces the same ipsilateral and contralateral motions, but without the presence of dots.

[1]Note that the controllers must be chosen such that they provide ultimately a continuous output on the motors.

Each of these motions are demonstrated five times consecutively. In each case, the demonstrator starts from the same starting position. While observing the demonstration, the robot tries to make sense of the experiment by extracting the demonstrator's intention underlying the task. I.e. it determines a set of constraints for the task, by extracting relevant features in a statistical manner. When the demonstration ends, the robot computes the trajectory that satisfies best the constraints extracted during the demonstration and generates a motion that follows this trajectory.
The scenario of our experiment is a replication of a set of psychological experiments conducted with young children and adults [1]. In these experiments, Bekkering and colleagues have shown that children have a tendency to substitute ipsilateral for contralateral gestures, when the dots are present. In contrast, when the dots are absent from the demonstration, the number of substitutions drop significantly. Thus, despite the fact that the gesture is the same in both conditions, the presence or absence of a physical object (the dot) affects importantly the reproduction. When the object is present, object selection takes the highest priority. Children, then, nearly always direct their imitation to the appropriate target object, at the cost of selecting the "wrong" hand. When removing the dots, the complexity of the task (i.e. the number of constraints to satisfy) is decreased, and, hence, constraints of lower importance can be fulfilled (such as producing the same gesture or using the same hand). Similar experiments conducted with adults have corroborated these results, by showing that the presence of a physical object affects the reproduction[2].
These experiments are informative to robotics, in helping us determine how to prioritize constraints (that we will also name goals throughout this paper) in a given task (and as such help us solve the *"correspondence problem"*). For instance, in the particular scenario, knowing the trajectory of the demonstrator's arm and hand path might not allow us to determine unequivocally the angular trajectories of the robot's arm. Indeed, depending on where the target is located, several constraints (goals) might compete and satisfying all of those would no always lead to a solution. For instance, in the case of contralateral motions, the robot's arm is too small to both reach the target and perform the same gesture. In that case, it must find a trade-off between satisfying each of the constraints. This amounts to determining the importance of each constraint with respect to one another.

2.1.1 Experimental setup

The demonstrator's motions are recorded by five X-sens motion sensors, attached to the torso and the upper- and lower-arms. 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 angular trajectories of the shoulder joint (3 degrees of freedom) and the elbow (1 degree of freedom) are reconstructed by taking the torso as referential, with an accuracy of 1.5 degrees.
A color-based stereoscopic vision system tracks the 3D-position of the dots, the demonstrator's hands, and the robot's hands at a rate of 15Hz, with an accuracy of 10 mm. The system uses two Phillips webcams with a resolution of 320x240 pixels. The tracking is based on color segmentation of the skin and the objects in the YCbCr color space.
The humanoid robot is a Fujitsu HOAP-2. It has 25 degrees of freedom (DOF). The robot is 50cm tall. In this experiment, trajectory control affects only the two arms (4 DOFs each). The torso and legs are set to a constant position to support the robot's standing-up posture.

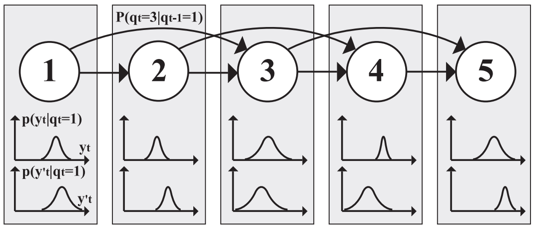

Figure 3: A left-right continuous HMM with 5 hidden states and 2 output variables y_t and y'_t. $P(q_t{=}j|q_{t-1}{=}i)$ is the probability to go from state i to state j at time t. $p(y_t|q_t{=}i)$ and $p(y'_t|q_t{=}i)$ are the emission distributions of variables y_t and y'_t while in state i.

2.1.2 Model

In order to reduce the dimensionality of the dataset to a subset of critical features, we pre-segment the joint angle trajectories and the hand path into a set of keypoints, corresponding to the inflexion points. Encoding of the trajectories follows our earlier work, using Hidden Markov Models (HMMs) [3]. The preprocessing phase gives us enough information to fix the HMM topology, so as to produce highly structured and accurate models during learning. Thus, each of the 4 joint trajectories is encoded in one left-right continuous HMM. Each hidden state represents a key feature j in the trajectory, and is associated with a stochastic representation of the observable y_j, encoding two variables, namely the time lag between two keypoints and the absolute angle. The hand path is represented by a single HMM that encode the keypoints of a Cartesian trajectory, with 3 output distributions for each state, to encode the 3 Cartesian components. The transition probabilities $P(q_t{=}j|q_{t-1}{=}i)$ and the emission distribution $p(y_t|q_t{=}i)$ are estimated by the *Baum-Welch* iterative method. The *forward-algorithm* is used to estimate a log-likelihood value that an observed sequence could have been generated by one of the model.

Let $D = \{\Theta, X, O, h\}$ and $D' = \{\Theta', X', O', h'\}$ be the datasets generated by the demonstrator and imitator respectively. $\{\vec{\theta}_1, \vec{\theta}_2, \vec{\theta}_3, \vec{\theta}_4\}$ are the generalized joint angle trajectories over the demonstrations, $\{\vec{x}_1, \vec{x}_2, \vec{x}_3\}$ the generalized Cartesian trajectory of the hand over the demonstrations, $\{o_{11}, o_{12}, o_{13}\}$ and $\{o_{21}, o_{22}, o_{23}\}$ the 3D location of the first and second dot respectively. We compute $d_{kj} = x_j - o_{kj}$ the distance between the hand and the dots at the end of a trajectory. $h = \{1, 2\}$ corresponds to the usage of the left and right arm respectively.

Following the framework developed in [2], we model the task's cost function as a weighted linear combination of metrics applied to 4 sets of variables, namely the joint angle trajectories, the hand path, the location of the objects at which actions are directed (the dots), and the laterality of the motion (which hand was being used). If we have $N = 4$ joint angles for each arm and $O = 2$ objects, and given the position of the hand and the objects is defined by $P = 3$ variables in the Cartesian space, we define the general cost function J as:

$$
\begin{aligned}
J \;=\; & \alpha_1 \sum_{i=1}^{N} w_1^i \; J_1(\vec{\theta}_i, \vec{\theta}'_i) \\
+ \; & \alpha_2 \sum_{j=1}^{P} w_2^j \; J_2(\vec{x}_j, \vec{x}'_j) \\
+ \; & \alpha_3 \sum_{k=1}^{O} \sum_{j=1}^{P} w_3^{kj} \; J_3(d_{kj}, d'_{kj})
\end{aligned}
$$

[2] In that case, the response latency is used instead of the proportion of errors

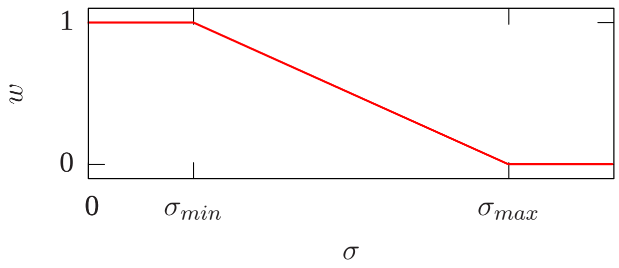

Figure 4: Function used to map a standard deviation σ to a weight factor $w \in [0, 1]$.

$$+ \quad \alpha_4 \; w_4 \; J_4(h, h') \tag{1}$$

J and J_i are normalized and comprised in the interval $[0, 1]$. $J = 0$ corresponds to a perfect reproduction. Optimizing the imitation consists of minimizing J. The w_i^j are factors that weight the importance of the associated set of variables. These factors are extracted from the demonstration and reflect the variance of the data during the demonstration. The factors α_i determine the relative importance of each set of variable. In other words, these parameters fix the importance of each constraint (or goal) in the overall task, and are fixed by the experimenter. α_1 determines the importance of reproducing correctly the joint angle trajectories, α_2 that of reproducing the hand path, α_3 that of placing the hand at the same distance to the object as in the demonstration and α_4, that of using the same hand (i.e. reaching with the same laterality, namely in ipsilateral or contralateral fashion). The cost functions $J_i \in [0, 1]$ associated with each of these different constraints, or goals, are defined as follows:

$$J_{1,2}(\vec{u}, \vec{u}') = 1 - f(\frac{\sum_{t=1}^{T} |u_t - u'_t|}{T}) \tag{2}$$
$$J_3(u, u') = 1 - f(|u - u'|) \tag{3}$$
$$J_4(u, u') = |p(u{=}1) - p(u'{=}2)| \tag{4}$$

where T is the number of data in the trajectory, and $p(u{=}1)$ the probability to use the left arm during the demonstrations. $f(v)$ is a transfer function, represented in Figure 4.
f normalizes and bounds each variable within these minimal and maximal values. This transformation has for effect to eliminate the effect of the noise, intrinsic to each variable, so that their relative effect can be compared.
One way to compare the relative importance of each set of variables (i.e. joint angles, hand path, distance hand-object and laterality) is to look at their variability. If the variance of a given variable is high, i.e. showing no consistency across demonstrations, then, this means that satisfying some particular instance of this variable had little bearing on the task. If the standard deviation of a given variable is low, the value taken by its weight w should be close to 1, so that this variable will have a strong influence in the reproduction of the task. Thus, with f the transfer function in Figure 4, we define:

$$w_{1,2,3}^j = \begin{cases} f(\bar{\sigma}_y) & \text{if } y \text{ is available} \\ 0 & \text{otherwise} \end{cases} \quad \forall j \tag{5}$$
$$w_4 = 2\;|p(h{=}1) - 0.5| \tag{6}$$

	α_1=α_2=α_3=α_4		α_1=$\frac{1}{2}\alpha_2$=$\frac{1}{4}\alpha_3$, α_4=0	
	Dots	No dots	Dots	No dots
Left contralateral	**0.16**	**0.14**	0.22	**0.11**
Right ipsilateral	0.36	0.47	**0.08**	0.16

Table 1: Value of the cost functions J for the optimal trajectory, used for reproducing the demonstration of a contralateral motion with right hand.

To evaluate the variability (mean standard deviation $\bar{\sigma}_y$) of the angular trajectories and of the hand path, we make use of the statistical representation provided by the HMM. After training of an HMM with a set of demonstrations of a given trajectory, we use the *Viterbi algorithm* to retrieve the best sequence of hidden states and associated keypoint values for this trajectory. An estimation of the standard deviation of the whole trajectories is then computed. w_4 represents the importance of using either the left or right hand (laterality of the imitation) and is based on a measure of the probability with which either hand has been used over the whole set of demonstrations. $w_4 = 0$ if there is no preference.
Once the cost function and the relative influence of each constraint have been determined, we generate an optimal (with respect to the cost function J) trajectory. In order to do this, we first generate a set of candidate trajectories for the hand path, using the HMMs and interpolation. To generate the joint angles trajectories corresponding to these hand paths, we have to solve the inverse kinematics equation given by: $\dot{\vec{x}} = \mathbf{J}\dot{\vec{\theta}}$, where **J** is the Jacobian. In order to account for the influence of the observation of the demonstrator's joint trajectories, we add another constraint to the pseudo-inverse solution:

$$\dot{\vec{\theta}} = \gamma(\mathbf{J}^{+}\dot{\vec{x}}) + (1-\gamma)\dot{\vec{\theta}}_d \tag{7}$$

where $\vec{\theta}_d$ is the joint angle trajectory generated by the HMM after training, and γ is a factor used to tune the influence of the two different terms (reproduction of hand path or joint angle trajectories). For each candidate path and associated set of joint trajectories, we compute the value of the cost function J. We, then, proceed to determining a local optimum for J by gradient-descent on γ. The corresponding (locally) optimal trajectory is, then, run on the robot to reproduce the demonstration.

2.1.3 Results

As expected, we have found little variation in either the joint trajectories, the hand paths, the distance hand-object or the laterality in any of the 4 tasks, forcing the satisfaction of all constraints during the reproduction of goal-directed motion. However, the most invariant features are those in relation to the object interaction. When the target dots are not present, their associated weights w_3^j values are zero. As there is no more object in the scene, the hand path and gestures become then the sole relevant features to reproduce.
In order to test the influence of the factors α on the performance of the imitation, we have tested two sets of values. α_1=α_2=α_3=α_4, i.e. no preference in goals. and α_1=$\frac{1}{2}\alpha_2$=$\frac{1}{4}\alpha_3$ with α_4=0 (no preference in hand). For each set, we computed the optimal trajectory. Table 1 gives the values of the cost function in each case.

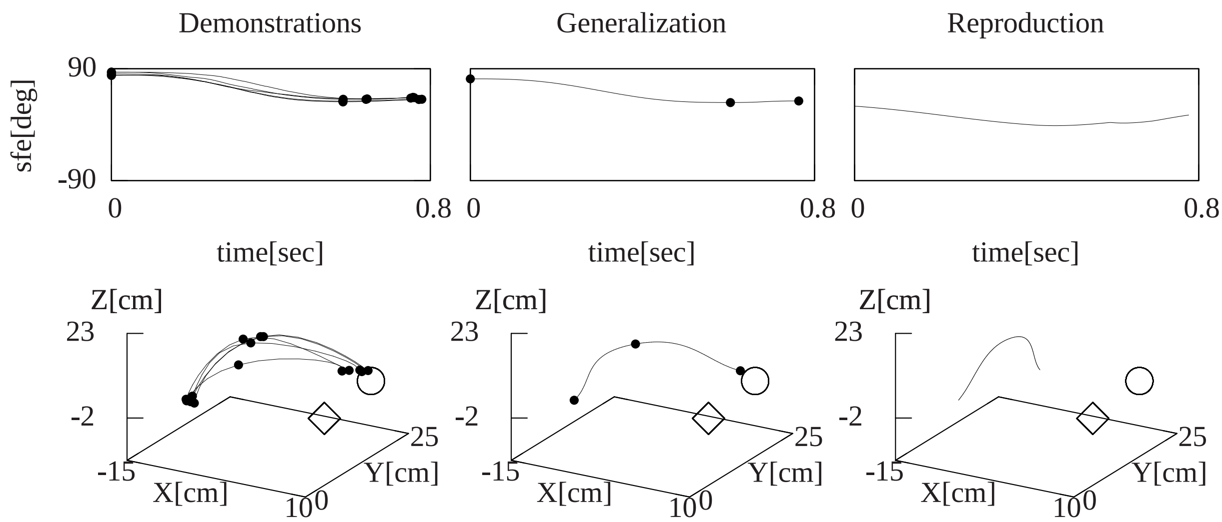

Figure 5: Joint angle and hand path plots of 5 demonstrations of a contraleral motion with right hand (left column), the trajectory retrieved by the HMM model of the 5 demonstrations (middle column), and reproduction of a new motion (right column). The points in the graphs represent the keypoints segmented and retrieved by the HMMs. The square and the circle show the position of the two dots on the table. Only the shoulder flexion-extension is represented for the joint angles.

3 Future Work

In the next 6 months, T12-T18, we will further develop the goal-extraction learning algorithm, combining different methods from Machine Learning (such as Kernel PCA and independent component analysis, for inferring the real dimensionality and optimal (uncorrelated and/or independent) representation of the dataset.
The generality of the extended algorithm will be validated with the humanoid HOAP-2 robot in a set of different manipulatory experiments, by varying importantly the experimental conditions (type of objects, type of metrics to infer, type of constraints and key features,using different human demonstrators).
The workplan for months T18-30 is part of a separate document, see RA4 workplan.

References

[1] H. Bekkering, A. Wohlschlger, and M. Gattis, "Imitation of gestures in children is goal-directed," *Quarterly Journal of Experimental Psychology*, vol. 53A (1), pp. 153–164, 2000.

[2] A. Billard, Y. Epars, S. Calinon, G. Cheng, and S. Schaal, "Discovering optimal imitation strategies," *Robotics and Autonomous Systems*, vol. 47:2-3, pp. 69–77, 2004.

[3] 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)*, Sendai, Japan, September 28 - October 2 2004, pp. 2769–2774.

[4] 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.

[5] R. Dillmann, "Teaching and learning of robot tasks via observation of human performance," *Robotics and Autonomous Systems*, vol. 47:2-3, pp. 109–116, 2004.

[6] 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*, vol. 358, no. 1431, pp. 537–547, 2003.

[7] J. Demiris and G. Hayes, "Imitation as a dual-route process featuring predictive and learning components: A biologically-plausible computational model," in *Imitation in Animals and Artifacs*, C. Nehaniv and K. Dautenhahn, Eds. MIT Press, 2001.

[8] A. Billard and G. Hayes, "Drama, a connectionist architecture for control and learning in autonomous robots," *Adaptive Behavior, Vol. 7:1*, pp. 35–64, 1999.

[9] A. Alissandrakis, C. 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*, vol. 32, pp. 482–496, 2002.

[10] A. Billard and R. Siegwart, "Robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 47:2-3, pp. 65–67, 2004.

[11] C. Nehaniv and K. Dautenhahn, "Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications," in *Interdisciplinary Approaches to Robot Learning*, J. Demiris and A. Birk, Eds. World Scientific Press, 2000, vol. 24, pp. 136–161.

[12] A. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings of the IEEE Intl Conference on Robotics and Automation*, 2002.

[13] A. Ude, C. Atkeson, and M. Riley, "Programming full-body movements for humanoid robots by observation," *Robotics and Autonomous Systems*, vol. 47:2-3, pp. 93–108, 2004.

[14] J. Zhang and B. Rössler, "Self-valuing learning and generalization with application in visually guided grasping of complex objects," *Robotics and Autonomous Systems*, vol. 47:2-3, pp. 117–127, 2004.

[15] M. Skubic and R. Volz, "Acquiring robust, force-based assembly skills from human demonstration," in *IEEE Transactions on Robotics and Automation*, vol. 16:6, 2000, pp. 772 –781.

[16] A. Billard and S. Schaal, "A connectionist model for on-line learning by imitation," in *Proceedings of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, hawaii, 2001.