ROBotic Open-architecture Technology for
Cognition, Understanding and Behavior

Project No. 004370

RobotCub
Development of a Cognitive Humanoid Cub

Instrument:	Integrated Project
Thematic Priority:	IST - Cognitive Systems

5.3 An algorithm for functional imitation

Due date: **01/03/2006 (+45 days)**
Submission Date:

Start date of project: **01/09/2004** Duration: **60 months**

Organisation name of lead contractor for this deliverable: **EPFL**

Responsible Person: **Prof. A. Billard, Micha Hersch, Florent Guenter**

Revision: **1.0**

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	**PU**
PP	Restricted to other programme participants (including the Commission Service)	
RE	Restricted to a group specified by the consortium (including the Commission Service)	
CO	Confidential, only for members of the consortium (including the Commission Service)	

Contents

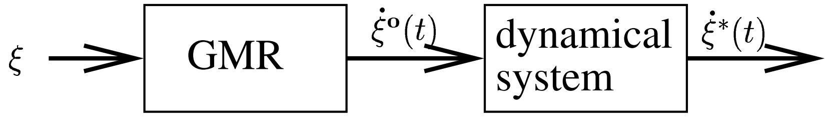

Figure 1: The demonstrated signals ξ are fed into a Gaussian Mixture Regression (GMR) algorithm, which produces a generalized observed velocity profile $\dot{\xi}^{\mathbf{o}}(t)$. This velocity profile is used to modulate a dynamical system and produces a velocity profile $\dot{\xi}^{*}(t)$ that can be performed by the robot.

Abstract

This deliverable presents a dynamical system to control imitation of simple manipulation tasks in a humanoid robot. The system is generic in that it makes no assumption on the number of degrees of freedom of the actuators of the demonstrator and the imitator. It allows to solve partly the correspondence problem and shows flexibility in the face of changes in the initial conditions and perturbations during the reproduction.

1 Introduction

It has been shown by [8, 6] that the dynamical system control paradigm is very well suited to control robots meant to interact with humans, because its ensures adaptivity to changes in the environment and to external perturbations. In a previous deliverable (Deliverable 5.2), we presented a dynamical system to control imitation of reaching movements. Here, we extend this work to control imitation of simple manipulation tasks, involving the robot's two arms and torso, and simple objects. This new system integrates a method developed elsewhere [1] for extracting the important features of a task to imitate with our dynamical system.

2 Overview

The aim of the algorithm is to reproduce the qualitative features common to several demonstrated trajectories, while adapting to different initial conditions and target locations. The algorithm can be decomposed into two main steps, illustrated in Fig 1.

$\xi(t) \in \mathbb{R}^s$ describes the complete state of the robot at each time step. In the application described in the rest of this document, ξ consists of the joint angles $\theta \in \mathbb{R}^n$ of the robot's upper torso and of the end-effectors' locations $\mathbf{x} \in \mathbb{R}^m$

$$\xi = [\mathbf{x}^T \ \theta^T]^T. \tag{1}$$

Note that the system, which is described next, does not make any assumption on the form of data and, thus, ξ could be composed of other variables, such as, for instance, the position of the objects to be manipulated or the same data projected in a latent space (e.g. after PCA decomposition as done in [3]).

After having being exposed to several demonstrations of the task, the algorithm extracts a generalized form of the original demonstration $\dot{\xi}^{\mathbf{o}}(t)$ (section 3). The generalized trajectories are then used to modulate an internal set of *desired trajectories* $\dot{\xi}^{\mathbf{s}}(t)$, resulting in a final set of trajectories $\dot{\xi}^{*}(t)$ which can actually be performed.

3 Generalization with Gaussian Mixture Regression

In this section, we briefly summarize the Gaussian Mixture Regression (GMR [4]) procedure used to obtain a single "model" trajectory from several demonstrations. This application has been described in details in [3, 1]. The principle of this method is to express the joint distribution of an "input" and an "output" variable in a Gaussian Mixture Model (GMM). In our case, the output variables are the $\dot{\xi}$ and the input variable is the time t. If we put those variables in a vector $v = [t\ \dot{\xi}^T]^T$, it is possible to express its probability density function as

$$p(v) = \sum_{k=1}^{K} \pi_k \mathcal{N}\big(v; \mu_k, \mathbf{\Sigma}_k\big), \tag{2}$$

where π_k is a weight factor and $\mathcal{N}(v; \mu_k, \mathbf{\Sigma}_k)$ is a Gaussian function with mean μ_k and covariance matrix $\mathbf{\Sigma}_k$:

$$\mathcal{N}\big(v; \mu_k, \mathbf{\Sigma}_k\big) = \big((2\pi)^d |\mathbf{\Sigma}_k|\big)^{-\frac{1}{2}} \exp\big(-\frac{1}{2}(v-\mu_k)^T \mathbf{\Sigma}_k^{-1}(v-\mu_k)\big) \tag{3}$$

where d is the dimensionality of the vector. The mean vector μ_k and covariance matrix $\mathbf{\Sigma}_k$ can be separated into their respective components:

$$\begin{aligned} \mu_k &= [\mu_{k,t}^T\ \mu_{k,\dot{\xi}}^T]^T & (4)\\ \mathbf{\Sigma}_k &= \begin{pmatrix} \mathbf{\Sigma}_{k,t} & \mathbf{\Sigma}_{k,t\dot{\xi}} \\ \mathbf{\Sigma}_{k,\dot{\xi}t} & \mathbf{\Sigma}_{k,\dot{\xi}} \end{pmatrix} & (5) \end{aligned}$$

This GMM can be trained using a standard E-M algorithm, taking the demonstrations as training data. Once the model is trained, it is possible to recover the most likely "output" variable $\dot{\xi}^{\mathbf{o}}$, given the observed "input" variable t.

$$\dot{\xi}^{\mathbf{o}} = \operatorname*{argmax}_{\dot{\xi}} P(\dot{\xi}|t) = \sum_{k=1}^{K} h_k \big(\mu_{k,\dot{\xi}} + \mathbf{\Sigma}_{k,\dot{\xi}t} \mathbf{\Sigma}_{k,t}^{-1}(t - \mu_{k,t})\big), \tag{6}$$

where the h_k are given by:

$$h_k = \frac{\pi_k \mathcal{N}(t; \mu_{k,t}, \mathbf{\Sigma}_{k,t})}{\sum_{k=1}^{K} \pi_k \mathcal{N}(t; \mu_{k,t}, \mathbf{\Sigma}_{k,t})}. \tag{7}$$

Thus, in our application, after training, the GMM can be used to generate a movement by taking the movement with the highest likelihood, given the trained model. This movement is taken to be the one to imitate. This method is illustrated in Fig. 7, where one sees a set of trajectories (light gray lines), the gaussian mixture components modeling them (ellipses) and the trajectory with the highest likelihood (thick line). Moreover, the variances of the GMM can provide an indication about the variability of the observed variables. Variables with low variability across demonstrations can be interpreted as more relevant to the task than variables with high variability.

4 Dynamical modulation of the demonstration

Let $\xi^{\mathbf{s}}(t)$ be a variable describing the current state of the robot. Again, this variable can contain the end-effector location and and/or the robot joint angles. The dynamical system

$$\ddot{\xi}^{\mathbf{s}}(t) = \alpha(-\dot{\xi}^{\mathbf{s}}(t) + \beta(\xi^{\mathbf{g}} - \xi^{\mathbf{s}}(t))), \tag{8}$$

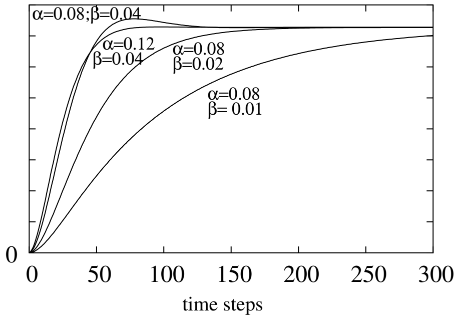

Figure 2: The dynamics described by equation 8 for various values of the parameters. Throughout this document the values $\alpha = 0.08$ and $\beta = 0.02$ were used.

with scalar constants $0 < \beta, \alpha < 1$, will smoothly bring the robot to a target state $\xi^{\mathbf{g}}$ with a straight line and remain there. Because this dynamical system creates a stable attractor on the target, the latter will be reached despite possible perturbations. This dynamical system is similar to the VITE model of human reaching movements [2], and has been used to enable a robot to perform robust reaching motions [5].
We start from this dynamical system and modulate it by a trajectory $\dot{\xi}^{\mathbf{o}}(t)$ abstracted from several demonstrations. The idea is to compute the difference between the velocity profiles $\dot{\xi}^{\mathbf{o}}(t)$ and of the reaching trajectory provided by the dynamical system described above. During the movement production, this difference, called $\dot{\xi}^{\mathbf{e}}(t)$, is added to the velocity provided by the dynamical system. Since the resulting end-effector positions and joint angle configuration may not be consistent with each other, a consistent solution $\dot{\xi}^{*}(t)$ for those variables is found by minimizing an *imitation metric* H.

More precisely, the difference between the (generalized) observed movement and the internal reaching movement $\mathbf{x}^{\mathbf{s_1}}(t)$ is computed as follows:

$$\ddot{\mathbf{x}}^{\mathbf{s_1}}(t) = \alpha(-\dot{\mathbf{x}}^{\mathbf{s_1}}(t) + \beta(\mathbf{x}^{\mathbf{g_1}} - \mathbf{x}^{\mathbf{s_1}}(t))) \quad (9)$$

$$\dot{\mathbf{x}}^{\mathbf{e}}(t) = \dot{\mathbf{x}}^{\mathbf{o}}(t) - \dot{\mathbf{x}}^{\mathbf{s_1}}(t) \quad (10)$$

$$(11)$$

When the action is directed toward a new goal $\mathbf{x}^{\mathbf{g_2}}$ this difference $\dot{\mathbf{x}}^{\mathbf{e}}(t)$ is reinjected into the dynamical system to obtain the desired end-effector and joint angle velocities $\dot{\mathbf{x}}^{\mathbf{d}}(t), \dot{\theta}^{\mathbf{d}}(t)$:

$$\ddot{\mathbf{x}}^{\mathbf{s_2}}(t) = \alpha(-\dot{\mathbf{x}}^{\mathbf{s_2}}(t) + \beta(\mathbf{x}^{\mathbf{g_2}} - \mathbf{x}^{*}(t))) \quad (12)$$

$$\dot{\mathbf{x}}^{\mathbf{d}}(t) = \dot{\mathbf{x}}^{\mathbf{s_2}}(t) + \gamma\dot{\mathbf{x}}^{\mathbf{e}}(t) \quad (13)$$

where $\mathbf{x}^{*}(t)$ is the actual end-effector location and γ is the modulation factor and vanishes to zero at the end of the movement. For the experiments presented here, we have $\gamma = 2 * \max((T - t/T)^2, 0)$, where T is the duration of the observed movement. As shown on Fig. 3, this ensures a smooth decay to zero. What happens in the equations described above is that the difference (in velocities) between the demonstrated movement $\dot{\mathbf{x}}^{\mathbf{o}}(t)$ and a straight line movement $\dot{\mathbf{x}}^{\mathbf{s_1}}(t)$ is computed and kept in variable $\dot{\mathbf{x}}^{\mathbf{e}}(t)$. This difference is then used to modulate the dynamical system (acting on $\mathbf{x}^{\mathbf{s_2}}(t)$) that reaches the target $\mathbf{x}^{\mathbf{g_2}}$. It must be noted that the influence of this modulation is more complex than just adding a velocity component to the straight line, it also influences the dynamical system itself through the position $\mathbf{x}^{*}(t)$. This means that even if the starting position and the target do not change between the demonstration and the imitation, the system will reproduce a trajectory which is somewhat different from the original

trajectory.
The same modulation is applied to the joint angle data.

$$\begin{aligned}
\ddot{\theta}^{\mathbf{s_1}}(t) &= \alpha(-\dot{\theta}^{\mathbf{s_1}}(t) + \beta(\theta^{\mathbf{g_1}} - \theta^{\mathbf{s_1}}(t))) && (14)\\
\dot{\theta}^{\mathbf{e}}(t) &= \dot{\theta}^{\mathbf{o}}(t) - \dot{\theta}^{\mathbf{s_1}}(t) && (15)\\
\ddot{\theta}^{\mathbf{s_2}}(t) &= \alpha(-\dot{\theta}^{\mathbf{s_2}}(t) + \beta(\theta^{\mathbf{g_2}} - \theta^{*}(t))) && (16)\\
\dot{\theta}^{\mathbf{d}}(t) &= \dot{\theta}^{\mathbf{s_2}}(t) + \gamma\dot{\theta}^{\mathbf{e}}(t), && (17)
\end{aligned}$$

where $\theta^{*}(t)$ is the actual joint configuration.

In general the variables $\dot{\mathbf{x}}^{\mathbf{d}}(t), \dot{\theta}^{\mathbf{d}}(t)$ will not be consistent. In order for them to be consistent the end-effector and joint angle velocities $\dot{\mathbf{x}}(t), \dot{\theta}(t)$ must satisfy the following equality:

$$\dot{\mathbf{x}}(t) = \mathbf{J}(t)\dot{\theta}(t), \tag{18}$$

where $\mathbf{J}(t)$ is the jacobian of the kinematic function at $\theta(t)$. In order to have consistent values, we find the values $(\dot{\mathbf{x}}^{*}, \dot{\theta}^{*})$ that optimize the imitation metric

$$H(\dot{\mathbf{x}}^{*}, \dot{\theta}^{*}, \dot{\mathbf{x}}^{\mathbf{d}}, \dot{\theta}^{\mathbf{d}}) = \frac{1}{2}(\dot{\mathbf{x}}^{*} - \dot{\mathbf{x}}^{\mathbf{d}})^{T}\mathbf{W_x}(\dot{\mathbf{x}}^{*} - \dot{\mathbf{x}}^{\mathbf{d}}) + \frac{1}{2}(\dot{\theta}^{*} - \dot{\theta}^{\mathbf{d}})^{T}\mathbf{W}_{\theta}(\dot{\theta}^{*} - \dot{\theta}^{\mathbf{d}}), \tag{19}$$

where $\mathbf{W_x} \in \mathbb{R}^{m \times m}$ and $\mathbf{W}_{\theta} \in \mathbb{R}^{n \times n}$ are diagonal matrices setting the relative importance of the end-effector location and joint angle respectively. In this application, $\mathbf{W}_{\theta}(t)$ and $\mathbf{W}_{x}(t)$ are weighting matrices which represent the time-varying constraints during the task. These are computed based on a measure of the statistical variations and relations across the different variables $\{\mathbf{\Sigma}_{t,\dot{\theta}^{\mathbf{d}}}, \mathbf{\Sigma}_{t,\dot{\mathbf{x}}^{\mathbf{d}}}\}$, see Section 3. Thus, we define:

$$\begin{aligned}
\mathbf{W}_{\theta} &= \left(\mathbf{\Sigma}_{t,\dot{\theta}^{\mathbf{d}}}\right)^{-1}\\
\mathbf{W}_{x} &= \left(\mathbf{\Sigma}_{t,\dot{\mathbf{x}}^{\mathbf{d}}}\right)^{-1} && (20)
\end{aligned}$$

For simplicity in the notation, the dependence on time was dropped in the above equation. The optimization is performed under the consistency constraint expressed by Eq. 18. The solution of this constrained optimization problem is given by:

$$\dot{\theta}^{*} = \left(\mathbf{W}_{\theta} + \mathbf{J}^{T}\mathbf{W_x}\mathbf{J}\right)^{-1}\left(\mathbf{W}_{\theta}\dot{\theta}^{\mathbf{d}} + \mathbf{J}^{T}\mathbf{W_x}\dot{\mathbf{x}}^{\mathbf{d}}\right) \tag{21}$$

This solution $\dot{\theta}^{*}$ is then executed by the robot.

5 Results

5.1 Simulations

In this section we illustrate the properties of the algorithm presented above on one dimensional trajectories. Fig. 4 shows a model (or observed) trajectory (solid line) that the system should imitate. The system has to imitate that trajectory and reach the same target, but starts from four different positions.
We see that the reproduced trajectories (dashed lines) replicate the qualitative features of the model trajectory (the two velocity peaks) and reach the target. However, the speed pattern is accelerated, the further the goal is from the initial position.
In Fig. 5, the effect of a sudden target displacement is shown. Similarly to Fig. 4 the system has to imitate an observed trajectory (solid line) starting from a different position. However, as indicated by the dotted line, the target suddenly switches to a new location. The dashed-dotted line shows how the system adapts its trajectory to reach the new target location.

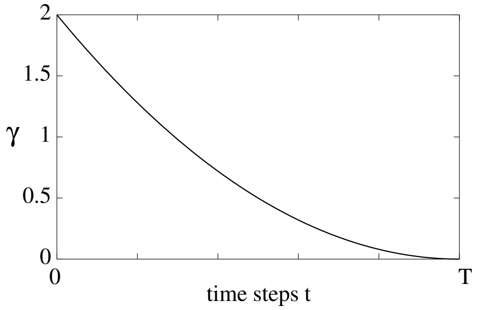

Figure 3: The behavior of the modulation factor γ. It smoothly reaches 0 by the end of the movement T, in order to let the robot reach the goal.

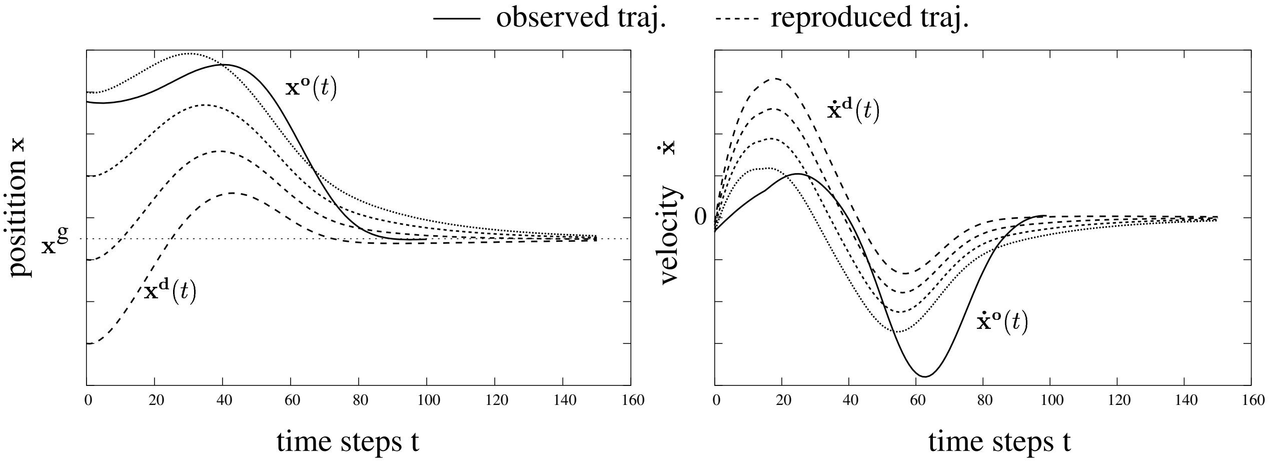

Figure 4: The reproduced trajectories (dashed line) are qualitatively similar to the observed trajectory, although they reach the goal from different starting points.

5.2 A grasping experiment

In order to validate this framework, we tested the system on a grasping task with the Hoap2 robot. The goal of the task was for the robot to grasp a bucket. There are two difficulties inherent to this task. First, the robot must reach the bucket handle with the hand in a particular orientation is order to be able to grasp it. This implies a particular arm configuration, as the robot has no wrist. Second, the robot has to avoid bumping into the bucket when approaching the handle. More precisely, this means that it has to approach the bucket from above. We use five degrees of freedom for this task (one for the torso and four for the arm). The robot is taught through kinesthetics. That is, the demonstrator moves the robot limbs to make it perform the task, as shown in Fig 6. After several demonstrations, the bucket is brought to a new position, and the robot has to grasp it. The hypothesis is that the dynamical system will bring the robot to the right location, while it will learn from the demonstrations that the bucket must be approached from above. If the bucket handle is not approached from above, the robot will hit the bucket while executing the movement, and will not be able to grasp it.

The results are displayed on Fig. 7 and 8. On Fig 7, the end-effector velocities of all demonstrations are shown (light gray) along with the generalized velocity profile (thick line). The ellipses show the gaussian

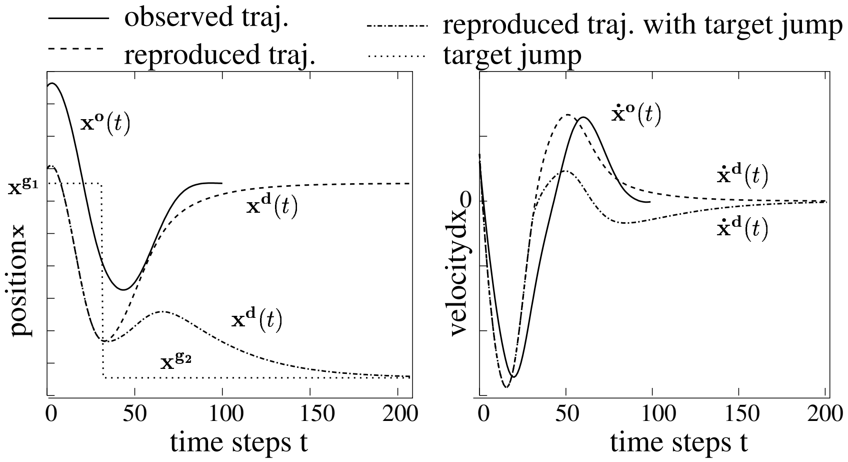

Figure 5: The effect of a sudden target displacement. If the target $\mathbf{x^{g_2}}$ suddenly switches (dotted line), the system smoothly adapts its trajectory to reach it.

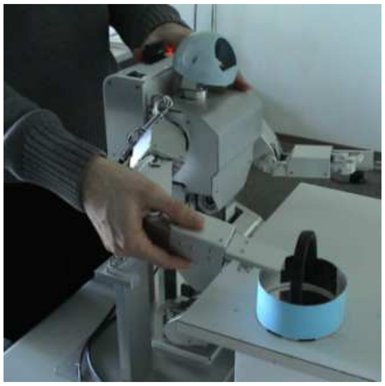

Figure 6: The demonstrator teaches the robot kinesthetically by moving the limbs of the robot.

mixture components.
On Fig. 8, top, one sees the model trajectory of the end-effector (solid line), the reproduced trajectory starting from a different point (curved dotted line) and the trajectory obtained without the modulation (straight dotted line). The pictures below show the results. With the model trajectory, generalized from several demonstrations, the robot is able to grasp the bucket handle (left). When the robot starts from a new location, the modulation allows the robot to grasp the handle (middle). Without the modulation (right), the robot hits the bucket handle and is thus not able to grasp it.

6 Discussion

The results presented above show that it is possible to modulate a dynamical system in order to obtain, to some extent, desirable properties of an observed trajectory. This way, it is possible to keep the adaptability of the dynamical system, in particular to changes in starting position and to target displacements, while retaining important properties of the demonstrations.
It must be noted that the algorithm described here is not limited to end-effector and joint angles variables. The same procedure could be applied with other variables, consistency conditions and imitation metrics

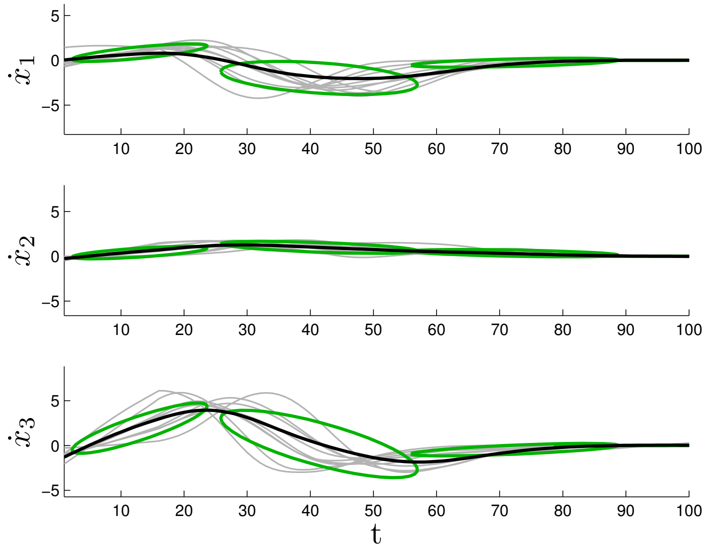

Figure 7: The generalization of the end-effector velocity. The end-effector velocities of all demonstrations are shown (light gray) along with the generalized velocity profile (thick line). The ellipses show the gaussian mixture components.

as well.

In the system described above, the exact effect of the modulation is difficult to foresee as the interplay between modulation and the dynamical system contains a feed-back loop. Moreover, it is difficult to rigorously define what would be an optimal modulation that would provide a yardstick for evaluating the algorithm. However, the experiments presented in the report, show that, for a particular task like grasping the bucket, it is possible to see if a modulation is good enough.

In the experiments presented here, the modulation factor γ had to be fixed by hand. It should be possible to automatically set this factor on a trial and error process. The robot would start with a starting value for γ. It would perform the task and update this value in case of failure, until it performs the task successfully.

Another limitation of the algorithm is that the observed trajectory used for the modulation is anchored to a time signal t. This means that the modulation cannot adapt to external perturbations, for example if the movement is obstructed. One way to overcome this limitation would be to express the modulation trajectory as a dynamical system. Ideally, the ultimate aim would be to obtain a new dynamical system, which takes the modulation into account. This way, the robot could build up a motor repertoire of dynamical systems.

We will work toward improving the system to overcome these limitations in the future.

References

[1] A. Billard, S. Calinon, and F. Guenter. Discriminative and adaptive imitation in uni-manual and bi-manual tasks. *Robotics and Autonomous Systems*, 54, 2006.

[2] D. Bullock and S. Grossberg. Neural dynamics of planned arm movements: Emergent invariants and speed-accuracy properties during trajectory formation. *Psychological Review*, 1988.

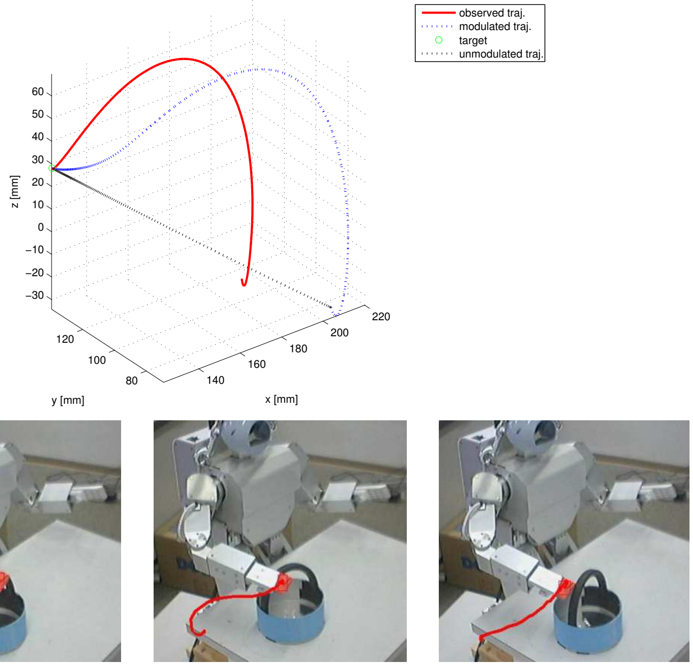

Figure 8: The grasping task. Above, the generalized trajectories of the demonstrations, the reproduced modulated trajectory and the unmodulated trajectory (straight path). Below are the corresponding trajectories (from left to right) when performed by the robot and tracked by a stereovision system. Using the unmodulated system the robot is not able to grasp the bucket handle.

[3] S. Calinon, F. Guenter, and A. Billard. On learning, representing and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man and Cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation*, 36, 2006.

[4] Z. Ghahramani and M.I.Jordan. *Advances in Neural Information Processing Systems 6*, chapter Supervised learning from incomplete data via an EM approach. Morgan Kaufmann Publishers, 1994.

[5] M. Hersch and A.G. Billard. A biologically-inspired model of reaching movements. In *Proceedings of the 2006 IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, Pisa, Feb. 2006.

[6] A.J. Ijspeert, J. Nakanishi, and S. Schaal. Trajectory formation for imitation with nonlinear dynamical systems. In *Proceedings of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS2001)*, pages 752–757, 2001.

[7] J.P. Scholz and G. Schöner. The uncontrolled manifold concept: identifying control variables for a functional task. *Experimental Brain Research*, 126:289–306, 1999.

[8] G. Schöner, M. Dose, and C. Engels. Dynamics of behaviour: theory and application for autonomous robot architecture. *Robotics and Autonomous Systems*, 16:213–245, 1995.