Learning Dynamical System Modulation for Constrained Reaching Tasks

Micha Hersch, Florent Guenter, Sylvain Calinon and Aude G. Billard

Learning Algorithms and Systems Laboratory - LASA, School of Engineering

École Polytechnique Fédérale de Lausanne - EPFL

Station 9, 1015 Lausanne, Switzerland

Email:{micha.hersch, florent.guenter, sylvain.calinon, aude.billard}@epfl.ch

Abstract— In this paper we combine kinesthetic demonstrations and dynamical systems to enable a humanoid robot to imitate constrained reaching gestures directed toward a target. Using a learning algorithm based on Gaussian Mixture Regression, the task constraints are extracted from several demonstrations. Those constraints take the form of desired velocity profiles for the end-effector and joint angle variables, with associated covariance matrices describing the variations allowed around the desired profiles. Those constraints are then used to modulate a dynamical system which has the reaching target as attractor. This way, the reaching trajectory can be reshaped in order to satisfy the constraints of the task, while preserving the adaptability and robustness provided by the dynamical system. In particular, the system can adapt to changes in the initial conditions and to target displacements occurring during the movement execution.

We first evaluate the potential of this method on experiments involving the Hoap3 humanoid robot putting an object into a box. We then show how a manipulation tasks can be executed as sequences of such constrained reaching movement. This is illustrated on a packaging task performed by the robot.

I. Introduction

The control of humanoid robots operating in human environments presents a huge challenge for roboticists. A major part of this challenge is related to the fact that, unlike traditional industrial robots, domestic robots must operate in dynamic and uncontrolled environments. The dynamical system approach to robot control has been argued to be a promising direction to tackle the problems linked to such requirements [1]. However, if this approach has been quite successful in designing attractor behaviors [2], obstacle avoidance behaviors [3], cyclic movements [4], the problem of designing a dynamical system that produces goal-directed trajectories satisfying specific constraints inferred by demonstrations does not have a fully satisfying solution yet. This problem has been addressed by [5] and this paper continues along this line. More specifically, we extend this work by considering multiple and possibly redundant variables to imitate, in a *Programming by Demonstration* framework.

Constrained reaching movements are present in nearly every manipulation task. For example when approaching an object in a particular way for grasping, or transporting an object while keeping a particular orientation or avoiding an obstacle. The system presented below can enable a humanoid robot to exe-

Fig. 1.   A human operator kinesthetically demonstrates a task to the robot, i.e. by grabbing and moving its limbs while the robot records joint angle trajectories.

cute such movements without being specifically programmed for that purpose. Rather, it is a general-purpose system that first **learns a task by generalizing over a set of kinesthetic demonstrations** performed by a human operator (see Fig. 1). Further, it can **adapt the learned behavior to different initial conditions and to changes in the environment while performing the task.**

The rest of this paper is structured as follows: Section II presents an overview of the system. The more detailed presentation starts with Section III which describes how the constraints are extracted from the demonstrations. Section IV then describes how those constraints are used to modulate a dynamical system. Experiments using the Hoap3 humanoid robot and involving domestic tasks are presented in Section V. Finally, in Section VI, a conclusion highlights the strengths and limitations of our system.

II. System Overview

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

$$\xi = [\mathbf{x}^T \ \theta^T]^T. \qquad (1)$$

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 information flow of the algorithm is illustrated in Fig. 2.
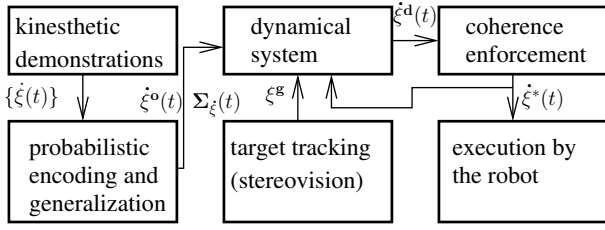
Fig. 2. A schematic overview of the system. In this paper, the variable ξ contains the joint angles θ and the end-effector location $\mathbf{x}$. Trajectories provided by kinesthetic demonstrations are fed into a learning system, the output of which, $\dot{\xi}^{\mathbf{o}}(t)$, is used to modulate a dynamical system. This dynamical system has as attractor a target $\xi^{\mathbf{g}}$ given by a stereo-vision tracking system and outputs a desired velocity $\dot{\xi}^{\mathbf{d}}(t)$. Coherence is then enforced between the elements of this desired velocity, resulting in a velocity $\dot{\xi}^{*}(t)$ performed by the robot.

After having being exposed to several demonstrations $\{\dot{\xi}(t)\}$ of the task, the algorithm extracts a generalized form of the original demonstration $\dot{\xi}^{\mathbf{o}}(t)$ using a probabilistic model, with associated covariance matrices representing the variations and correlations allowed along the trajectories. The generalized trajectories and associated covariance matrices are then used to modulate a dynamical system. Coherence constrained are applied to the output of the dynamical system $\dot{\xi}^{\mathbf{d}}(t)$, resulting in a final set of trajectories $\dot{\xi}^{*}(t)$ which can actually be performed.

Note that the system described below 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 as done in [6].

III. CONSTRAINT EXTRACTION

In this section, we briefly summarize the Gaussian Mixture Regression (GMR [7]) procedure used to obtain a single "model" trajectory from several demonstrations. This application has been described in details in [6], [8]. The principle of this method is to model the joint distribution of "input" and "output" variables as a Gaussian Mixture Model (GMM). In our case, the output variables are the velocities $\dot{\xi}$ and the input variable is the time t. If we join those variables in a vector $\upsilon = [t\ \dot{\xi}^{T}]^{T}$, it is possible to model its probability density function as

$$p(\upsilon) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\upsilon; \mu_k, \Sigma_k), \qquad (2)$$

where π_k is a weighting factor (the prior) and $\mathcal{N}(\upsilon; \mu_k, \Sigma_k)$ is a Gaussian function with mean μ_k and covariance matrix Σ_k:

$$\mathcal{N}(\upsilon; \mu_k, \Sigma_k) = \left((2\pi)^d |\Sigma_k|\right)^{-\frac{1}{2}} \cdot \qquad (3)$$
$$\exp\left(-\frac{1}{2}(\upsilon - \mu_k)^T \Sigma_k^{-1}(\upsilon - \mu_k)\right), \quad (4)$$

where d is the dimensionality of the vector υ. The mean vector μ_k and covariance matrix Σ_k can be separated into their respective input and output components:

$$\mu_k = [\mu_{k,t}^T\ \mu_{k,\dot{\xi}}^T]^T \qquad (5)$$

$$\Sigma_k = \begin{pmatrix} \Sigma_{k,t} & \Sigma_{k,t\dot{\xi}} \\ \Sigma_{k,\dot{\xi}t} & \Sigma_{k,\dot{\xi}} \end{pmatrix} \qquad (6)$$

This GMM is trained using a standard Expectation-Maximization (EM) algorithm, taking the demonstrations as training data. EM starts with an initial estimate of the parameters and converges to a local maximum of the likelihood function. As initialization is important, the initial parameters are thus estimated by a rough segmentation of the data using k-means clustering. The number of components is found using Bayesian Information Criterion (BIC), as described in [6]. After training, we thus obtain a joint probability density function for the input and the output. Because it is a GMM, the conditional probability density function, i.e., the probability of the output conditioned on the input is also a GMM. Hence, it is possible to recover the expected output variable $\dot{\xi}^{\mathbf{o}}$, given the observed input variable t:

$$\dot{\xi}^{\mathbf{o}}(t) = \sum_{k=1}^{K} h_k(t)\left(\mu_{k,\dot{\xi}} + \Sigma_{k,\dot{\xi}t}\Sigma_{k,t}^{-1}(t - \mu_{k,t})\right), \quad (7)$$

where parameters $h_k(t)$ are given by:

$$h_k(t) = \frac{\pi_k \mathcal{N}(t; \mu_{k,t}, \Sigma_{k,t})}{\sum_{k=1}^{K} \pi_k \mathcal{N}(t; \mu_{k,t}, \Sigma_{k,t})}. \qquad (8)$$

The covariance matrix of this conditional probability distribution is given by:

$$\Sigma_{\dot{\xi}}(t) = \sum_{k=1}^{K} h_k^2(t)\left(\Sigma_{k,\dot{\xi}} - \Sigma_{k,\dot{\xi}t}\Sigma_{k,t}^{-1}\Sigma_{k,t\dot{\xi}}\right) \qquad (9)$$

Thus, in our application, the GMM can be used to generate a smooth movement by taking the expected velocities $\dot{\xi}^{\mathbf{o}}(t)$ conditioned on time t. 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 expected trajectory (thick line). Moreover, $\Sigma_{\dot{\xi}}(t)$ provides an indication about the variability and correlations across the observed variables. Indeed, at any given time step, variables with low variability across demonstrations can be interpreted as more relevant to the task than variables with high variability.

IV. DYNAMICAL SYSTEM MODULATION

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

$$\ddot{\xi}^{\mathbf{s}}(t) = \alpha(-\dot{\xi}^{\mathbf{s}}(t) + \beta(\xi^{\mathbf{g}} - \xi^{\mathbf{s}}(t))), \qquad (10)$$

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, as illustrated in Fig. 3, left. Because this dynamical system creates a stable attractor on the target, the latter will be reached despite possible perturbations. This dynamical system
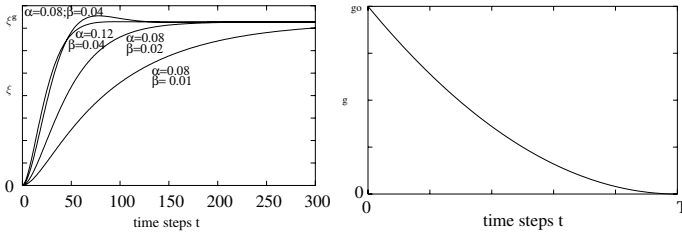
Fig. 3. Left: dynamics of the signal ξ according to (10) for various values of the parameters. Throughout this document the values $\alpha = 0.08$ and $\beta = 0.02$ were used. Right: behavior of the modulation factor γ. It smoothly reaches 0 at the end of the movement T, in order to let the robot reach the goal. Throughout this paper, γ_0 is set to 1.5.
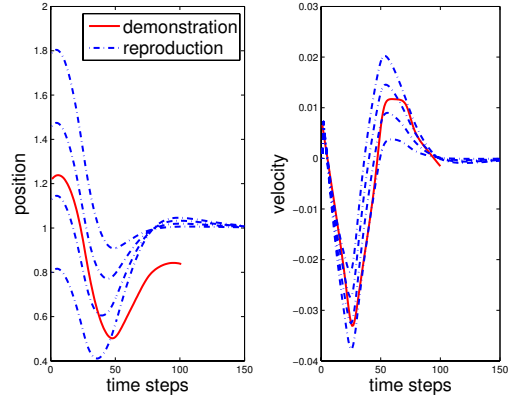

Fig. 4. The reproduced trajectories (dash-dotted line) are qualitatively similar to the observed trajectory (solid line), although they reach the goal from different initial positions.

is similar to the VITE model of human reaching movements [9], and has been used to enable a robot to perform robust reaching motions [2].

We start from this dynamical system and modulate it by a trajectory $\dot{\xi}^{\mathbf{o}}(t)$ abstracted from the demonstrations. The idea is simply to compute a weighted average between the velocities $\dot{\xi}^{\mathbf{o}}(t)$ extracted from the demonstrations (7) and the velocities given by the dynamical system (10). Since the resulting end-effector positions and joint angle configuration may not be consistent, a consistent solution $\dot{\xi}^*(t)$ for those variables is found by minimizing an *imitation metric H*.

More precisely, we run two concurrent dynamical systems, one for each set of variables (end-effector location and joint angles). So, when reaching for a target position $\mathbf{x}^{\mathbf{g}}$, we run the following system:

$$\ddot{\mathbf{x}}^{\mathbf{s}}(t) = \alpha(-\dot{\mathbf{x}}^{\mathbf{s}}(t) + \beta(\mathbf{x}^{\mathbf{g}} - \mathbf{x}^*(t))) \quad (11)$$

$$\dot{\mathbf{x}}^{\mathbf{s}}(t) = \dot{\mathbf{x}}^*(t) + \ddot{\mathbf{x}}^{\mathbf{s}}(t) \quad (12)$$

$$\dot{\mathbf{x}}^{\mathbf{d}}(t) = \gamma(t)\dot{\mathbf{x}}^{\mathbf{s}}(t) + (\gamma_0 - \gamma(t))\dot{\mathbf{x}}^{\mathbf{o}}(t), \quad (13)$$

where $\mathbf{x}^*(t)$ and $\dot{\mathbf{x}}^*(t)$ are the real position and velocity and $\gamma(t)$ is a factor weighting the influence of the dynamical system versus the demonstration. This factor varies with the time according to

$$\gamma(t) = \gamma_0 \cdot \max(((T-t)/T)^2, 0), \quad (14)$$

where T is the duration of the observed movement. As shown in Fig. 3, right, this ensures a smooth decay to zero and is necessary in order to guarantee that the target will be reached. The duration of movement reproduction can exceed T. In the dynamical system above (11), one can notice that the damping is not made on the actual velocity of the system $\dot{\mathbf{x}}^*(t)$ but on $\dot{\mathbf{x}}^{\mathbf{s}}(t)$ which is the velocity component specified by the dynamical system. This way, (11) does not correct the velocity specified by the demonstrations. It only adjusts for the position. Indeed, the aim is to keep the velocity similar to the demonstrated ones while adjusting for the positions in order to reach the target.

The same system modulation is applied to the joint angle

data.

$$\ddot{\theta}^{\mathbf{s}}(t) = \alpha(-\dot{\theta}^{\mathbf{s}}(t) + \beta(\theta^{\mathbf{g}} - \theta^*(t))) \quad (15)$$

$$\dot{\theta}^{\mathbf{s}}(t) = \dot{\theta}^*(t) + \ddot{\theta}^{\mathbf{s}}(t) \quad (16)$$

$$\dot{\theta}^{\mathbf{d}}(t) = \gamma(t)\dot{\theta}^{\mathbf{s}}(t) + (\gamma_0 - \gamma(t))\dot{\theta}^{\mathbf{o}}(t), \quad (17)$$

In general the variables $\dot{\mathbf{x}}^{\mathbf{d}}(t), \dot{\theta}^{\mathbf{d}}(t)$ will not be consistent. Consistency of the end-effector and joint angle velocities $\dot{\mathbf{x}}(t), \dot{\theta}(t)$ is ensured by applying the constraint:

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

where $\mathbf{J}(\theta(t))$ 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: [1]

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

where $\mathbf{W}_{\mathbf{x}} \in \mathbb{R}^{m \times m}$ and $\mathbf{W}_\theta \in \mathbb{R}^{n \times n}$ are matrices setting respectively the relative importance and covariance of the end-effector location and joint angle variables. As mentioned above, the covariance matrices of the demonstrated velocities, $\mathbf{\Sigma}_{\dot{\xi}}(t)$ (9), give an indication of how relevant the corresponding variables are at time t. Thus, $\mathbf{W}_\theta$ and $\mathbf{W}_{\mathbf{x}}$ are directly determined by the inverse of $\mathbf{\Sigma}_{\dot{\xi}}$. The optimization is performed under the consistency constraint expressed by (18). The solution of this constrained optimization problem (as shown in [8]) is given by:

$$\dot{\theta}^* = (\mathbf{W}_\theta + \mathbf{J}^T \mathbf{W}_{\mathbf{x}} \mathbf{J})^{-1} (\mathbf{W}_\theta \dot{\theta}^{\mathbf{d}} + \mathbf{J}^T \mathbf{W}_{\mathbf{x}} \dot{\mathbf{x}}^{\mathbf{d}}) \quad (20)$$

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

A. 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). In the

---

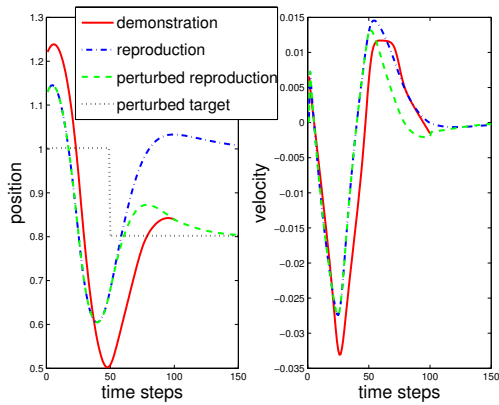[1]Dependence on time is omitted to simplify the notation.

Fig. 5. The effect of a sudden target displacement. If the target suddenly switches (dotted line), the system smoothly adapts its trajectory to reach it.

Fig. 6. The experimental setup for the object-in-the-box task. The Hoap3 humanoid robot puts a cylinder into a box. The box is tracked by a stereo-vision system mounted in the eyes of the robot. The white line represents the trajectory depicted in Fig 8.

Fig. 7. The Gaussian Mixture Model trained with the 26 kinesthetic demonstrations of the object-into-the box task. Only the the models corresponding to the velocities of the end-effector are represented. $\dot{x}_1, \dot{x}_2, \dot{x}_3$ correspond to the velocities along the x,y and z directions of the Cartesian space. The thin lines represent the demonstrations and the thick lines the generalization $\dot{\xi}^o(t)$ learned using the GMR. The ellipses represent the Gaussian components of the joint probability distribution. The models and trajectories corresponding to the joint angle velocities are not represented.

global system, this model trajectory is the outcome of the GMR. The system has to imitate that trajectory and reach the same target, but starts from four different positions. The exact features of the trajectory that need to be replicated can vary from task to task, and it is therefore difficult to evaluate the quality of the imitation. Nonetheless, we see that the reproduced trajectories (dash-dotted 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 the next section, results of physical experiments are displayed to allow a better evaluation of the system.

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.

V. EXPERIMENTS

The simulation results presented above illustrate that the system is able to reproduce some of the features of a model trajectory. We now turn to physical experiments with clear success/failure criteria to confirm the effectiveness of the system. For those experiments, we use a Hoap3 humanoid robot built by Fujitsu. This robot has two cameras located in its eyes, and four backdrivable degrees of freedom (dofs) in each arm. The fifth dof, the wrist, is not backdrivable and could thus not be used for kinesthetic demonstrations. The robot is controlled in position through a built-in on-board controller.

A. Putting an object into a box

The first experimental task consists in putting an object (here a cylinder) into a box. This is a constrained reaching task in the sense that the robot must reach the box from above, without hitting it. In the setting considered here, the robot knows (through vision tracking of a colored patch) where the box is, but has no information relative to its size. The system
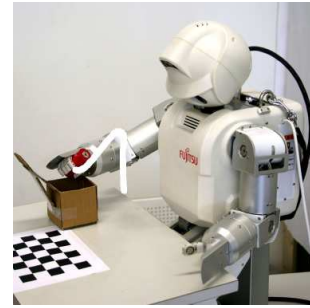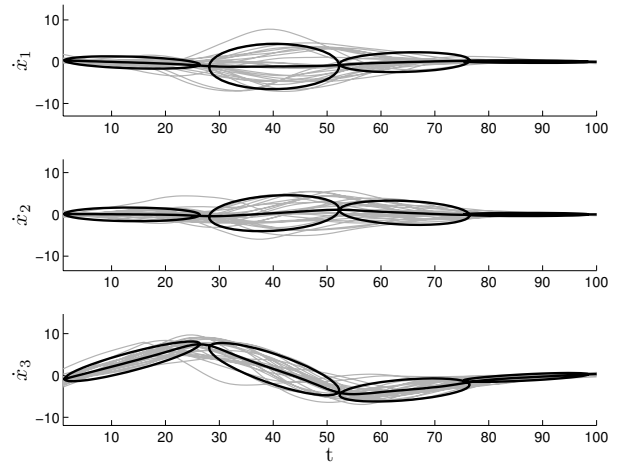
must extract from the kinesthetic demonstrations made a human operator that it must first reach up above the box and then down into the box (see Fig. 1). The experimental setup is shown in Fig. 6. During the demonstrations, the human operator makes the robot passively put the cylinder into the box, varying the initial arm posture and the position of the box. Using the method described in section III, velocities for all joint angles and the end-effector are extracted.

The corresponding Gaussian Mixture Model figures in Fig. 7. In this figure, one can notice that, unlike the horizontal components, the extracted velocity of the vertical component $\dot{x}_3$ is significantly different from zero, indicating that it must go up and then down. Moreover, we see that the variance is smaller for this vertical component (elongated ellipses), i.e. the trajectory is more constrained.
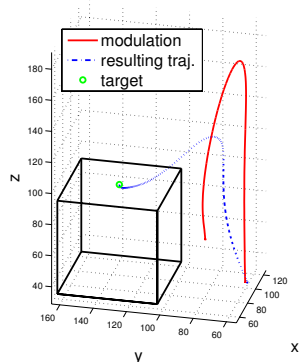
4

Fig. 8. A successful constrained reaching execution. The solid line shows the trajectory used for modulation and the dashed line shows the trajectory resulting from the modulated dynamical system. This trajectory is also represented in Fig 6.
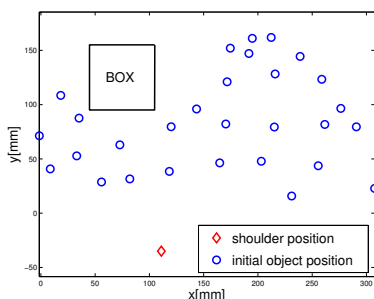

Fig. 9. A top-view of the starting location of the cylinder. The robot is facing upward. The robot successfully puts the cylinder into the box for all tested initial configurations. The robot is prevented from reaching some initial positions by its torso or the box itself, which explains the white areas near the shoulder and at the right of the box.
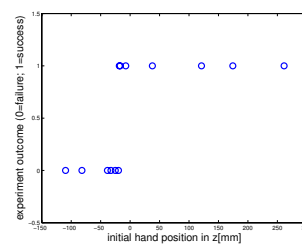

Fig. 10. The robustness to variations in the vertical initial position. The lower lines represents failures, the upper line success. The robot cannot successfully perform the task if the initial position is much lower than the demonstrated ones, which started from the height of the table (about 40 mm).
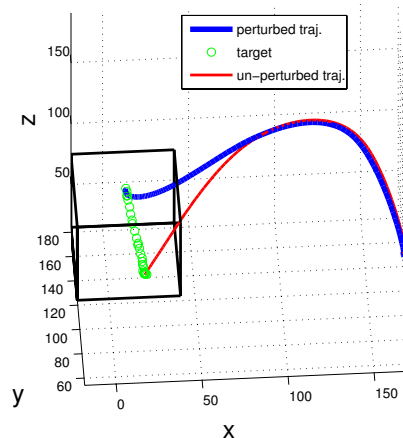

Fig. 11. The box is moved while the robot is reaching for it. The thin line is the trajectory without the perturbation. If the box is displaced during the movement, as shown by the circles, the system adapts its trajectory (thick line).

During reproduction, the robot tracks the position of the box using its stereo-vision system. The reaching target is set to be the top of the box, and the starting position is given by the motor sensors. At the end of the trajectory, the robot releases the cylinder and the task is performed successfully if the cylinder falls into the box. Fig. 8 shows such a successful trajectory. The same trajectory is drawn on Fig. 6 to illustrate its effect in the experimental setup.

In order to evaluate the adaptiveness of the system, trials were made with different starting locations. The initial positions were distributed on the horizontal plane of the table. For each initial position, the task was performed successfully. It was not possible to find a starting position on that horizontal plane for which the robot failed. Fig. 9 plots the starting configurations that where tried. One sees that trials were made from different sides and distances from the box. For the plane of the table, the whole workspace was covered.

However, if the initial position is set much below the height of the table, the robot cannot perform the task successfully. This is illustrated in Fig. 10, where one sees the success or failure of the robot depending from the initial position in z.

In order to evaluate the robustness to perturbations, the box was displaced during the execution of the task. An example is displayed in Fig. 11. During movement execution, the box is displaced, and the robot adapts its trajectory to reach it.

B. Packaging

As mentioned above, constrained reaching movements are present in many domestic tasks and applications. To conclude our set of experiments, we extend the previous task in order to have a more complete application. The task considered here is a simple packaging task, which consists of putting and object into a box, closing it and ringing a bell. This task can be segmented into six constrained reaching subtasks:

1) reach for the object (and grab it);
2) put the object into the box;
3) hold the box with one hand
4) close the box with the other hand;
5) reach for the handle of a bell
6) ring the bell.

For this experiment, each of those six subtasks is trained separately, i.e., the task segmentation is made manually. This segmentation could also be performed automatically, but this is not the focus the present paper. The object, the bell handle and the box do not have a fixed location. The task is performed
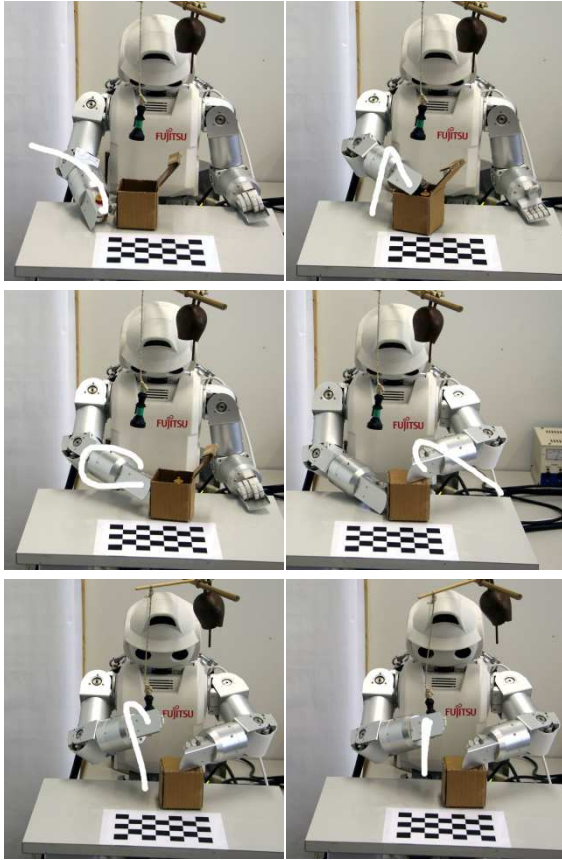
5

Fig. 12. The trajectories generated for the execution of the packaging task. In the first row, the robot reaches for the object and puts it into the box. In the second row, it holds the box with the right hand and closes it with the left hand. In the third row, it reaches for the bell and rings it.

by successively performing the constrained reaching motions for each subtask. The setup and an example of resulting trajectories are illustrated in Fig. 12. For this kind of tasks, it is crucial that the robot adapt to changes in the environment, as those changes are sometimes cause by the robot itself. For the example of Fig. 12, one sees that the box has moved between the first and the second subtask (the two upper pictures). The robot has slightly moved the box while putting the object in it, but could nevertheless perform the task successfully.

VI. CONCLUSION

In this paper we have presented an algorithm that enables a robot to perform constrained reaching tasks in a dynamic environment. This algorithm consists first of extracting constant features of the movement and then use them in conjunction with a dynamical system. This allows the system to reproduce a task, while adapting to new external conditions.

Of course this system still leaves room for improvement. For example, one can notice that the trajectories generated by the system are similar to the demonstrated ones especially at their beginning. Due to the decaying of $\gamma(t)$, at the end of the trajectory the influence of the dynamical system gets bigger and the similarity gets smaller. Thus, tasks that involve a specific modulation at the end of the trajectory may not be as successfully reproduced as the tasks presented in this paper. Moreover, if the dynamical system is adaptive, the modulation itself is not. Thus, if the robot arm is halted by an external intervention, one could have discontinuities in the velocities when resuming the movement. Work is currently done to overcome those limitations.

In fine, this system presents an interesting combination of learning and wiring. The fact that the target has to be reached is wired in by the use of the dynamical system. This gives the adaptability to various initial conditions and to target displacements. The learning component, present in the form of Gaussian Mixture Regression, gives the generality of the system, its ability to perform many different tasks. Those tasks can thus be learned without the need of building a model of the environment. In our example, the system does not need any information about size of the box. This information is implicitly given by the demonstrations. This property is very valuable for humanoids operated by unexperienced users as it does not require any programming skills.

REFERENCES

[1] G. Schöner, M. Dose, and C. Engels, "Dynamics of behavior: theory and applications for autonomous robot architectures," *Robotics and Autonomous Systems*, vol. 16, pp. 213–245, 1995.

[2] M. Hersch and A. Billard, "A biologically-inspired model of reaching movements," in *Proceedings of the 2006 IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2006.

[3] I. Iossifidis and G. Schöner, "Autonomous reaching and obstacle avoidance with the anthropomorphic arm of a robotic assistant using the attractor dynamics approach," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004, pp. 4295–4300.

[4] L. Righetti and A. Ijspeert, "Programmable central pattern generators: an application to biped locomotion control," in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, 2006.

[5] A. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds., 2002, pp. 1547–1554.

[6] 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*, vol. 36, no. 5, 2006, in Press.

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

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

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