Learning Nonlinear Multi-Variate Motion Dynamics for Real-Time Position and Orientation Control of Robotic Manipulators

E. Gribovskaya and Aude Billard

Abstract— **We present a generic framework that allows *learning non-linear dynamics* of motion in manipulation tasks and generating dynamical laws for control of *position and orientation*. This work follows a recent trend in Programming by Demonstration in which the *dynamics* of an arm motion is learned: position and orientation control are learned as multi-variate dynamical systems to preserve correlation within the signals. The strength of the method is three-fold: i) it extracts dynamical control laws from demonstrations, and subsequently provides concurrent smooth control of both position and orientation; ii) it allows to generalize a motion to unseen context; iii) it guarantees on-line adaptation of the motion in the face of spatial and temporal perturbations. The method is validated to control a four degree of freedom humanoid arm and an industrial six degree of freedom robotic arm.**

I. INTRODUCTION

Humanoid robots are primarily thought as platforms meant to interact with humans helping them to accomplish typical daily life chores. One would expect humanoid robots to be endowed with skillful manipulation capacities similar in flexibility and precision to that displayed by humans. To develop such capacities, it has become crucial to depart from conventional planners and consider alternative methods of control: designing task- and robot-specific controllers seems nowadays a time-consuming and ineffective solution. Therefore, preference gradually changes in favor of flexible and generic control methods that can adapt to various tasks and robots' constraints [5], [15].

Programming by Demonstration [2] (PbD) has appeared as one way to respond to this growing need for intuitive control methods. PbD formulates user-friendly methods by which a human user can show to a robot how to accomplish a specific task. Here, we focus on teaching a robot manipulation activities, a class of tasks that requires both coordinated motion of limbs and accurate positioning of an end-effector.

The idea that motions of biological systems may be governed by dynamical systems has been a recurrent topic of discussion in human motion science and biology. More recently, this concept has been adopted in robotics with the hope to endow robots with motions showing the similar level of flexibility and efficiency as those seen in animals and humans.

Dynamical Systems (DS) provide effective and elegant means of encoding motions; they encode trajectories through time-independent functions that defines the temporal evolution of the motions. Generalization of the motions to unobserved parts of the space results immediately from the application of the functions to the new set of input variables.

From the motion science perspective, a motion during a manipulation task consists of two major parts: a *transport* phase followed by a *grasping* phase. The phases are linked through an *approach vector* (a vector that defines a motion direction just prior to grasping an object). The transport phase is described by a translational and orientational control policy. Both of these unfold in time simultaneously and in coordination, resulting in a smooth and natural-looking motion. The translational component brings a robot's hand in the proximity of a manipulated object, while the orientational component tends to align the hand with the approach vector. Therefore, for successful accomplishment of manipulation tasks, these components should be encoded and reproduced together, replicating a coordination pattern.

Dynamical Systems have been successfully applied in robotics [13], [14] to control the translational component of motion. However, to this date no generic Dynamical Systems framework has been proposed for tackling control of arbitrary non-linear movements in the complete operational space (i.e. dynamical control of both position and orientation of an end-effector) .

In this work, we develop a generic framework for learning Dynamical Systems to control the transport phase of a motion. We demonstrate its generalization abilities to reproduce a task in contexts unvisited during training (this property is highly relevant for PbD applications, where demonstrated data covers just a limited part of a workspace), and the ability of the system to recover from spatio-temporal perturbations in real-time (robustness to spatio-temporal perturbations distinguishes encoding with autonomous DS from a body of existing time-indexed encodings).

II. RELATED WORKS

The work presented here mainly relates to two fields, therefore in this section, we review related work in each of the involved domains.

A. Motion planning

Planning of manipulation motions is the topic of a body of research, see, e.g., [10], [16], [19]. Conventional planners introduce an effective way to motion generation if all information about environment is known and modelled

This work was supported by the European Commission through the EU Projects FEELIX-GROWING (FP6-IST-045169) and ROBOT@CWE (FP6-034002).

E. Gribovskaya and Aude Billard are with Learning Algorithms and Systems Laboratory (LASA), Ecole Polythechnique Fédérale de Lausanne, Switzerland elena.gribovskaya@epfl.ch, aude.billard@epfl.ch

before a robot starts to move. However, planners may fail to accomplish a task if environment changes rapidly and unpredictably (as when interacting with humans): under perturbations, a motion has to be replanned, and this process may be too slow to be computed in real time if perturbations are frequent. The current approaches to path generation for robotic arms further assume that the wrist axis is aligned with the direction of a motion, and the final alignment of the wrist axis with a desired orientation may be performed in the second stage (once the arm is in the vicinity of an object). Alternatively to using planners for the second stage of the transport part of a motion, methods of automatic grasping [4], [9], [18] address a problem of control of the wrist orientation and fingers postures. Once a hand is brought into the proximity of an object to be manipulated, grasping algorithms generate a motion that aligns the wrist's axis with an approach vector and organizes the fingers into a grasping configuration. This two-stage process increases the total duration of a movement; decoupling of trajectory generation for position and orientation control makes the adaptation to perturbations cumbersome, requiring finely tuned heuristics.

Coupled generation of position and orientation control through dynamical systems as we propose here, helps to overcome these problems; it also ensures that generated motions look smoother and more predictable for humans working with a robot.

B. Trajectory encoding

A number of recent approaches to PbD, including our prior work, investigate the use of dynamical systems to encode the demonstrated motion [13], [8], [11]. All of the above approaches use a pre-defined dynamical system as a motion primitive, and learn a function that modulates this primitive to fit a desired trajectory or velocity profile. Furthermore, such systems operate in a single dimension, that is a considerable limitation as it decreases the precision of fitting and discards information about correlation between the variables.

In this paper, we investigate a method whereby the Gaussian Mixture Models directly embed the multi-variate dynamics of a motion. This is advantageous in that i) it allows for concurrent control of the translational and orientational component of a motion; ii) the system is autonomous, i.e. time-independent, thus giving more resilience to perturbations leading the system from its desired path; iii) the correlation across the dynamics driving each variable is embedded in the encoding, thus ensuring that the correlation across the variable and the dynamics of motion of each variable separately are preserved.

III. PROBLEM STATEMENT

Let us consider that the state[1] of our robotic system can be unambiguously described by a variable ξ and that the workspace of the robot forms a sub-space X in $\mathbb{R}^N$. Consider further that the state of our robotic system is governed by an *Autonomous Dynamical System* $\langle X, f \rangle$.

Let us further assume that the state transition map f is a non-linear continuously differentiable function and that the motion model is driven by a first order differential equation with a single equilibrium point $\bar{\xi}$, such that:

$$\forall t \in T = [t_0; \infty]; [\xi; \dot{\xi}] \in X \subset \mathbb{R}^N \quad (1)$$

$$\dot{\xi}(t) = f(\xi(t)); \quad \dot{\bar{\xi}} = f(\bar{\xi}) = 0. \quad (2)$$

Let the set of M N-dimensional demonstrated datapoints $\{\xi^i, \dot{\xi}^i\}_{i=1}^M$ be instances of the above motion model. The problem then consists in building an estimate $\hat{f}$ of f based on the set of demonstrations. To this end, we will approximate the function in a subregion[2] $C \subset X$, so that:

$$\hat{f} : C \rightarrow C; \quad \hat{f}(\xi(t)) \approxeq f(\xi(t)), \forall \xi \in C. \quad (3)$$

C is further referred to as the *region of applicability* of a learned dynamics.

Here, we plan motions in a referential related to a manipulated object, therefore, without loss of generality, we can transfer the attractor $\bar{\xi}$ to the origin, so that $\bar{\xi} = 0 \in C \subset X$ is now the equilibrium point of f and by extension of its estimate $\hat{f}$, i.e. $\hat{f}(0) = f(0) = 0$. If C is contained within the *region of attraction* Δ of $\bar{\xi}$, then the estimate $\hat{f}$ is asymptotically stable at $\bar{\xi}$ and any motion initiated from $\xi(t_0) \in C$ will asymptotically converge to the target $\bar{\xi}$.

A. Learning Multivariate Dynamics

To construct $\hat{f}$ from the set of demonstrated trajectories, we follow a statistical approach and define $\hat{f}$ using *Gaussian Mixture Models* (GMM).

GMMs define a joint probability distribution function $\mathcal{P}(\xi^i, \dot{\xi}^i)$ over a training set of demonstrated trajectories $\{\xi^i, \dot{\xi}^i\}_{i=1}^M$, as a mixture of a finite set of K Gaussian distributions $\mathcal{N}^1..\mathcal{N}^K$ (with μ^k and Σ^k being the mean value and the covariance matrix of a Gaussian distribution $\mathcal{N}^k$):

$$\mathcal{P}(\xi^i, \dot{\xi}^i) = \frac{1}{K} \sum_{k=1}^{K} \pi^k \mathcal{N}^k(\xi^i, \dot{\xi}^i; \mu^k, \Sigma^k) \quad (4)$$

here

$$\mu^k = [\mu^k_\xi \ ; \ \mu^k_{\dot{\xi}}] \text{ and } \Sigma^k = \begin{pmatrix} \Sigma^k_\xi & \Sigma^k_{\xi\dot{\xi}} \\ \Sigma^k_{\dot{\xi}\xi} & \Sigma^k_{\dot{\xi}} \end{pmatrix} \quad (5)$$

where each Gaussian probability distribution $\mathcal{N}^k$ is given by:

$$\mathcal{N}^k(\xi^i, \dot{\xi}^i; \mu^k, \Sigma^k) = \quad (6)$$

$$\frac{1}{\sqrt{(2\pi)^{2d}|\Sigma^k|}} e^{-\frac{1}{2}(([\xi^i, \dot{\xi}^i] - \mu^k)^T (\Sigma^k)^{-1} ([\xi^i, \dot{\xi}^i] - \mu^k))}.$$

π^k are prior probabilities of the mixture's components

[1] The state of a dynamical system represents the *minimum amount of information* required to describe the effect of past history on the future development of this system [12].

[2] Estimating the dynamics in the whole state-space X would be practically infeasible due to the excessive number of demonstrations that this would require.

The GMMs model is initialized using the *k-means* clustering algorithm starting from a uniform mesh and iteratively refined through Expectation-Maximization (EM) [7].

To generate a new trajectory from a learned GMMs, one can then sample from the probability distribution function given by Eq.4. This process is called Gaussian Mixture Regression (GMR).

Taking the posterior mean estimate of $\mathcal{P}(\dot{\xi}|\xi)$, the estimate of our function $\dot{\hat{\xi}} = \hat{f}(\xi)$ can be expressed as a non-linear sum of linear dynamical systems, given by:

$$\dot{\hat{\xi}} = \sum_{k=1}^{K} h_k(\xi)(\Sigma_{\dot{\xi}\xi}^{k}(\Sigma_{\xi}^{k})^{-1}(\xi - \mu_{\xi}^{k}) + \mu_{\dot{\xi}}^{k}), \qquad (7)$$

where $h_k(\xi) = \frac{\mathcal{P}(\xi;\mu_{\xi}^{k},\Sigma_{\xi}^{k})}{\sum_{k=1}^{K}\mathcal{P}(\xi;\mu_{\xi}^{k},\Sigma_{\xi}^{k})}$, $h_k(\xi) > 0$, and $\sum_{k=1}^{K} h_k(\xi) = 1$.

B. Learning position and orientation control

The learning algorithm described above makes no assumption concerning the type of variables to be used for training, thus we are unconstrained in our choice of variables for motion learning. Here, we choose that: a) the translation component of a motion of an end-effector is described by a vector of Cartesian coordinates $\mathbf{x} \in \mathbb{R}^3$; b) the orientation of the end-effector is described by a pair of variables $\{\mathbf{s}, \phi\}$, that correspond to the axis and an angle of rotation [1]. According to this representation, the orientation of a moving referential $x'y'z'$ with respect to a fixed referential xyz (see Figure 1) is described by a rotational axis $\mathbf{s} \in \mathbb{R}^3$ and an angle $\phi \in [0; 2\pi]$.

The axis/angle representation allows to define the orientation of a rigid body in space; this encoding is similar to the quaternion representation and can be easily converted into the latter to facilitate the computations; the axis/angle representation is advantageous for Dynamical Systems motion generation as it does not require renormalization at each time step as quaternions do, and allows for a more compact representation than rotational matrices.

Therefore, a robot should learn the following functions from demonstrations:

$$\dot{\mathbf{x}} = \hat{f}_{\mathbf{x}}(\mathbf{x}),\ [\dot{\mathbf{s}}; \dot{\phi}] = \hat{f}_o(\mathbf{s}, \phi); \qquad (8)$$

where $C_{\mathbf{x}} \subset \mathbb{R}^3,\ C_o \subset \mathbb{R}^3 \times [0; 2\pi]$.

C. Optimal Control for Inverse Kinematics

Conventional inverse kinematics approaches considering mainly trajectory following may lead to unfavorable joint postures and poor performance near singularities. A number of recent works propose to reformulate the IK problem in terms of optimal control [17], [11]. We follow the same approach by formulating the IK as an optimization problem in which we aim at 1) following as closely as possible a generated trajectory (here given by the learned dynamics); and 2) finding a joint space solution which is the closest to the center of the joint space θ_0. To tackle the problem of avoiding joint boundaries, we additionally impose hard constraints on the optimization problem.

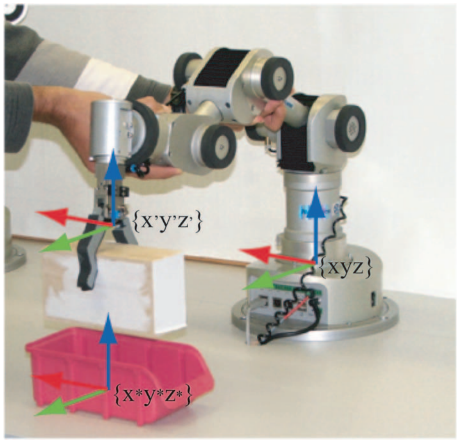


Fig. 1. We encode tasks in a referential located at a target $\{x^*y^*z^*\}$; this referential moves with the target and is defined with respect to a global fixed $\{xyz\}$(usually attached to static parts of the robot). The end-effector coordinates $\{x'y'z'\}$ are thus defined with respect to the target's referential.
.

TABLE I

ON-LINE TASK REPRODUCTION

```
1   Learn the estimates $\hat{f}_x, \hat{f}_o$ of the dynamics underlying the position
    and orientation of the end-effector's motion:

2   Detect a target position in the global referential $\{xyz\}$,
    see Figure 1: $\{\mathbf{x}^*, \mathbf{s}^*, \phi^*\}$

3   Recompute the current position of the end-effector in
    the target referential $\{x^*y^*z^*\}$: $\{\mathbf{x}_0, \mathbf{s}_0, \phi_0\}$

4   LOOP from $t = 0$ until the target position is reached

6       Infer the velocity at the next time step through GMR (Eq. 7):
        $\dot{\hat{\mathbf{x}}}_t = \sum_{k=1}^{K} h_{\mathbf{x}}^{k}(\mu_{\dot{\mathbf{x}}}^{k} + \Sigma_{\dot{\mathbf{x}}\mathbf{x}}^{k}(\Sigma_{\mathbf{x}}^{k})^{-1}(\mathbf{x}_{t-1} - \mu_{\mathbf{x}}^{k}))$
        $[\dot{\hat{\mathbf{s}}}_t; \dot{\hat{\phi}}_t] = \sum_{k=1}^{K} h_{\mathbf{s}_t,\phi}^{k}(\mu_{\dot{\mathbf{s}},\dot{\phi}}^{k} +$
        $+\Sigma_{\dot{\mathbf{s}},\dot{\phi},\mathbf{s},\phi}^{k}(\Sigma_{\mathbf{s},\phi}^{k})^{-1}([\mathbf{s}_{t-1}; \phi_{t-1}] - \mu_{\mathbf{s}_{t-1},\phi_{t-1}}^{k}))$

8       Solve the Inverse Kinematics problem (Eq. ??) to find: $\dot{\theta}_t$

9       Send command $\theta_t$ to a robot and get motors feedback

10      Compute the actual position and orientation of the end-effector
        $\mathbf{x}_t, \mathbf{s}_t, \phi_t$

10  END
```

In the previous work of ours [3], we define a metric of imitation $\mathbb{H}(\dot{\theta}_t) : \mathbb{R}^{N_j} \rightarrow \mathbb{R}$ (N_j is a number of joints (DOFs) in a manipulator) that balances the effect of following a particular trajectory in the orientation and the Cartesian space, as follows:

$$\begin{aligned}\mathbb{H}(\dot{\theta}_t) &= (J_{\mathbf{x}}\dot{\theta}_t - \dot{\hat{\mathbf{x}}}_t)^T(\hat{\Sigma}_{\mathbf{x}})^{-1}(J_{\mathbf{x}}\dot{\theta}_t - \dot{\hat{\mathbf{x}}}_t) \qquad (9)\\ &+ (J_{\omega}\dot{\theta}_t - \hat{\omega}_t)^T(\hat{\Sigma}_{\omega})^{-1}(J_{\omega}\dot{\theta}_t - \hat{\omega}_t)\\ &+ (\dot{\theta}_t - (\theta_0 - \theta_{t-1}))^T\Sigma_{\theta}^{-1}(\dot{\theta}_t - (\theta_0 - \theta_{t-1}));\end{aligned}$$

where $\dot{\hat{\mathbf{x}}}_t$ and $\hat{\omega}_t$ are the translational and rotational velocities generated by the estimates of the dynamics given by $\hat{F}_{\mathbf{x}}(\mathbf{x}), \hat{F}_{\mathbf{s}}(\mathbf{s}), \hat{F}_{\phi}(\phi)$; $\hat{\Sigma}_{\dot{\mathbf{x}}}^{-1}, \hat{\Sigma}_{\omega}^{-1} = \hat{\Sigma}_{\dot{\phi}}^{-1}\hat{\Sigma}_{\dot{\mathbf{s}}}^{-1}$ are the estimated variance at a point $\{\dot{\hat{\mathbf{x}}}_t, \hat{\omega}_t\}$; Σ_θ^{-1} is a weight matrix of joint values that is built so as to have maximum values on the joint boundaries and rapidly decrease as joint values approach the center of the workspace; $J_{\mathbf{x}}, J_\omega$ are respectively position and orientation Jacobian matrices of a robot's arm; a symbol "^{T}" refers to the matrix transpose.

Discretizing in time and assuming a local linear approximation of the derivative at each time step: $\dot{\xi}_t = \xi_t - \xi_{t-1}$, we then minimize $\mathbb{H}$ on a set $[\theta_{min}; \theta_{max}]$, where $\theta_{min}, \theta_{max}$ are lower and upper joint boundaries.

A solution of the minimization problem in Eq.(9) has the following form:

$$\dot{\theta} = (\tilde{J}_{\mathbf{x}} + \tilde{J}_\omega + \Sigma_\theta^{-1})^{-1}(\tilde{J}_{\mathbf{x}}\dot{\hat{\mathbf{x}}}_t + +\tilde{J}_\omega\hat{\omega}_t + \Sigma_\theta^{-1}(\theta_0 - \theta_{t-1})) \tag{10}$$

$$\omega_t = J_\omega\dot{\theta}_t, \ \ \dot{\mathbf{x}}_t = J_{\mathbf{x}}\dot{\theta}_t;$$

where $\tilde{J}_{\mathbf{x}} = (J_{\mathbf{x}}^T\hat{\Sigma}_{\mathbf{x}}^{-1}J_{\mathbf{x}})$, $\tilde{J}_\omega = J_\omega^T\hat{\Sigma}_\omega^{-1}J_\omega$.

If the IK problem is not over-constrained (the rank of the Jacobian is not less than a number of controlled DOFs in the operational space) and the robot operates inside the workspace, the optimization problem will have a solution that coincides with the output of a learned dynamical controller. However, when controlling an under-constrained manipulator or if it is in the proximity of singularities, the generalized IK presented above may be considered as a source of intrinsic perturbations affected the dynamical systems that generate a motion. Indeed, as a robot cannot reach a position precisely, the IK solution lies in the neighborhood of the initial signal. The final position and orientation will however always be reached given there exists the exact IK solution at these points. Table I summarizes the steps followed during the reproduction.

IV. Experimental results and Discussion

A. Set-up

We validated the above method in two experiments, see Figure 2 with a 4 degree of freedom arm of a humanoid robot HOAP-3 from Fujitsu and a 6 degree of freedom industrial-type KATANA arm from Neuronics.

The KATANA arm was taught to put a rectangular wooden brick into a rectangular container slightly bigger than the brick; see Figure 2-top. Thus, in order to accomplish the task, the robot had to precisely adapt its orientation and position with respect to the container before releasing the object.

The clench of the HOAP-3 is rather small, therefore it can grasp only thin objects. In this task the robot had to grasp a box that was particularly thin along one dimension; see Figure 2-bottom.

The tasks were chosen because (1) as the objects are asymmetrical, both tasks required a specific orientation of the robot's end-effector when grasping the object, (2) the tasks could be described by dynamical systems with single attractors, and (3) their success/failure could be easily estimated.

In both experiments the robots were shown tasks 4 times by a human user guiding their arms. Position and orientation of manipulated objects were tracked with a stereovision system using Augmented Reality Toolkit (ARToolKit) markers.

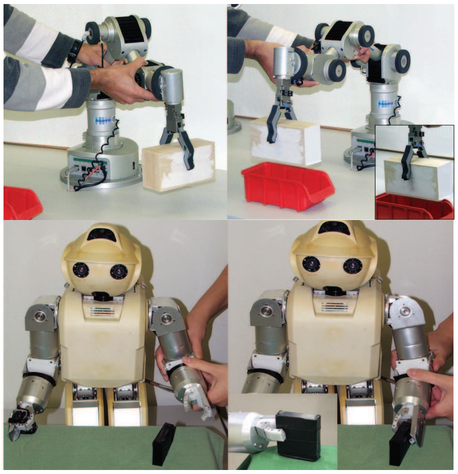


Fig. 2. Set-up of the experiments. Left: a 6 DOF KATANA arm places a wooden brick into a container. To accomplish the task the robot has to lift the brick, move it along a curved trajectory, while rotating its gripper for this to be aligned with the container. These orientation constrained at target are highlighted in the frames embedded in the right figures. Right: the HOAP-3 grasps a box. To accomplish the task the HOAP-3 should approach to the box with a specific orientation, as its clench is small.

B. Results

After training, we tested the system by letting the robots reproduce the tasks manipulating the objects placed at different locations in space (to demonstrate generalization capacities of the learned dynamics) and by generating spatial perturbations during reproduction (to demonstrate abilities to recover from perturbations in real-time).

The results of the experiments are summarized in Figures 3-6.

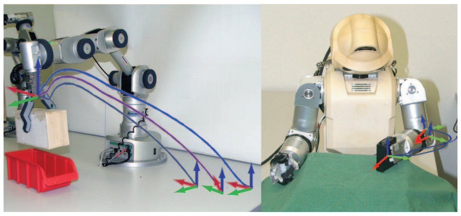


Fig. 3. Reproduction of dynamically generated trajectories starting from three locations and orientations not seen during training (the orientations at the onset of a motion are depicted by the frames of reference). This shows the generalization abilities of the learned dynamical controllers.

The dynamical dependencies between variables and their derivatives exhibit non-linearities; see Figures 4-6, however, the proposed method manages to accurately grasp those dependencies with relatively sparse GMMs representations. Therefore, overall memory requirements for storing motion models are much less in comparison with memory-based approaches. Furthermore, correlation between variables within positional and orientational components is strong (note the existence of regular patterns within signals $\mathbf{x}, \{\mathbf{s}, \phi\}$), and

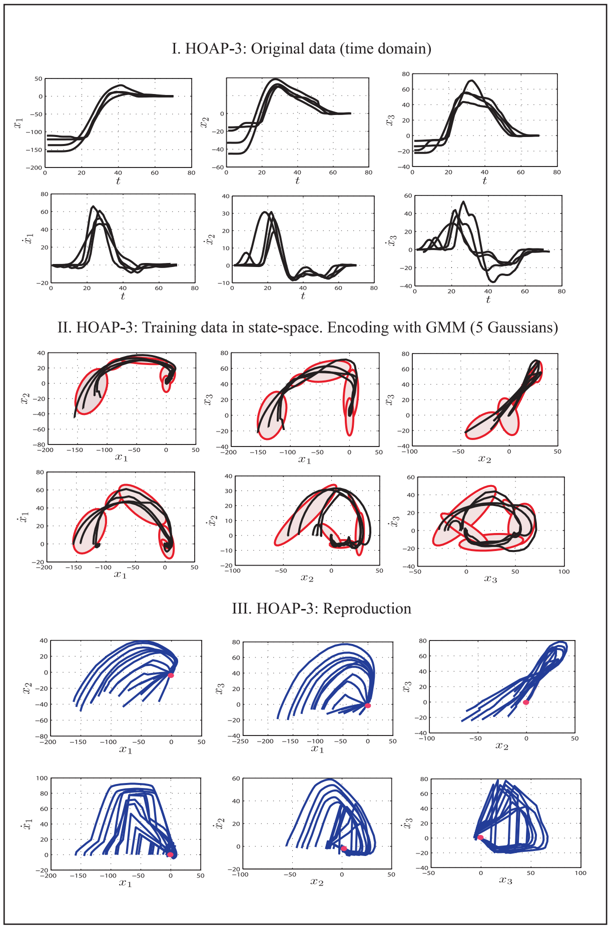


Fig. 4. Results of encoding the transport phase of demonstrations in the experiment with HOAP-3.

thus their multi-variate learning and control leads to more faithful reproduction of demonstrated motions.

To test the generalization abilities and the robustness to perturbations we performed experiments in different conditions, by varying starting positions and shifting the objects to be grasped. The results are presented in Figures 3, 7; in both experiments learning resulted in estimates of the motion dynamics, locally stable in an area at least twice wider than that covered by the demonstrations. The robots also accurately reached the desired position with the desired orientation from all starting locations and irrespective of the perturbations.

As the KATANA arm is a 6DOFs manipulator, when specifying both the location and orientation of the end-effector, the inverse kinematics is fully determined; however, the robot tends to bump into joint boundaries as the axes of several motors are aligned. By using the inverse kinematics controller that brings the arm closer to the center of the workspace we smoothly avoid the joint boundaries. If one merely assigns a joint's value to the boundary value each time it tries to leave an admissible interval, this joint in most cases will stick to the boundary value for the rest of a motion.

In contrast, for the HOAP-3 robot, the inverse kinematics

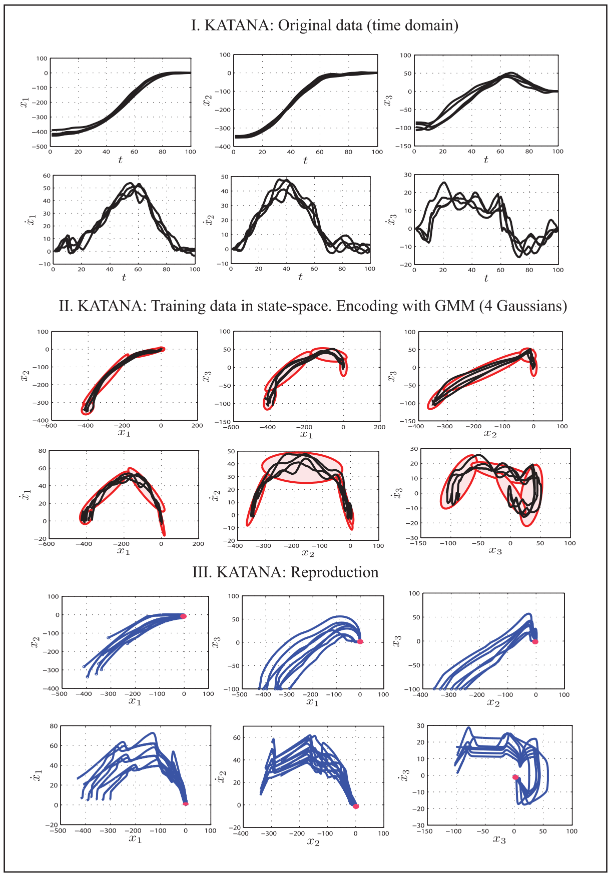


Fig. 5. Results of encoding the transport phase of demonstrations in the experiment with KATANA.

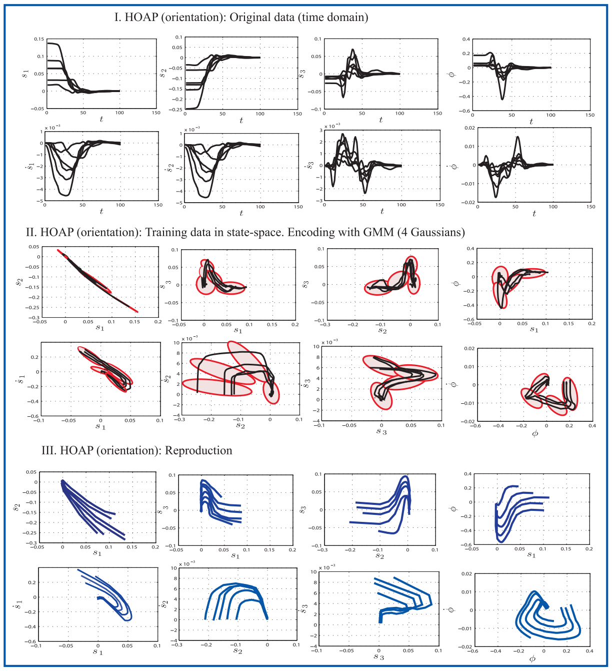
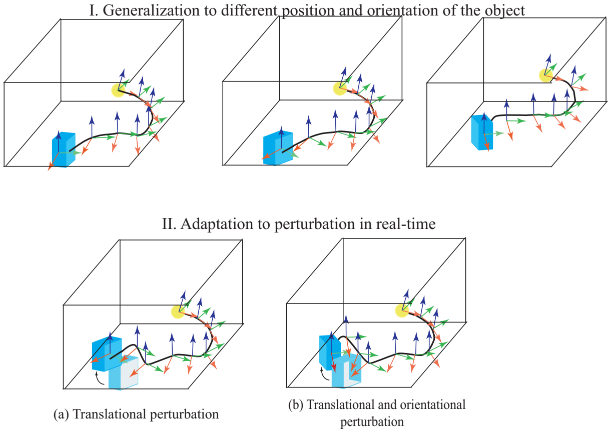


Fig. 6. Results of encoding the orientation phase of demonstrations in an experiment with HOAP-3. Note, the existence of non-linear correlation between an axis and an angle of rotation.



I. Generalization to different position and orientation of the object

II. Adaptation to perturbation in real-time

(a) Translational perturbation

(b) Translational and orientational perturbation

Fig. 7. Experiments with the humanoid robot HOAP-3. Referentials display the change in the orientation of the robot's end-effector along the motion. Starting positions of the hand are highlighted by yellow circles. I. Generalization abilities of the method: the robot successfully grasped a box placed in different positions in the workspace. These configurations of the box have not been observed by the robot during demonstration. II. Real-time adaptation to perturbations: while the robot was moving towards the box its position was perturbed (a), both position and orientation were perturbed (b). Control of position and orientation through dynamical systems allows the robot to smoothly adapt to both types of perturbations.

is under-defined. Thus, we cannot guarantee to find a precise solution that satisfy both the position and orientation dynamics along a motion. In this case exploiting an optimization mechanism that balances between the constraints on position and orientation taking onto account variance in the demonstrations helps to find a solution that leads to a successful reproduction.

C. Discussion and Conclusion

In this paper we presented a controller based on dynamical systems that allows to reproduce manipulation tasks in contexts unvisited during demonstration. Learning and reproducing position and orientation control simultaneously offers a "pre-shape" control strategy that bears similarity to the way humans approach objects (i.e. by orienting a hand while approaching the target [6]). Encoding orientation and position in two separate dynamical systems (while coupling them through the generalized IK) endows the robot with the ability to smoothly adapt to perturbations affecting either of these two constraints separately or simultaneously. If a target object is just shifted from its original location, the robot's hand will not change the orientation and will remain aligned with the desired approach vector; this vector will be translated closer to the object as the effect of dynamical position control. In further work, we will evaluate the coupling of position and orientation control directly in a single dynamical system. As the orientation was getting aligned with the desired approaching vector during transportation part of motion, the tasks were accomplished faster than in the case of subsequent translational and rotational positioning.

Please, note, that for the sake of brevity we omitted some details regarding enforcing the local stability of a dynamical system at the attractor. Further details on how this can be tackled, including source code can be found at: **http://lasa.epfl.ch/elena/learning-dynamics.htm**.

References

[1] S. L. Altmann. *Rotations, Quaternions, and Double Groups*. Claredon Press, Oxford, 1986.

[2] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Robot Programming by Demonstration. In *Handbook of Robotics*, volume chapter 59. MIT Press, 2008.

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

[4] M. Buss and H. Hashimoto. Manipulation skill modeling for dexterous hands. pages 3403–3408 vol.4, May 1994.

[5] 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*, 37(2):286–298, 2007.

[6] M. Christel and A. Billard. Comparison between macaques' and humans' kinematics of prehension: the role of morphological differences and control mechanisms. *Behavioral and Brain Research*, 131:169–184, 2002.

[7] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Statistic Society*, 39:1–38, 1977.

[8] K. Dixon and P. Khosla. Trajectory representation using sequenced linear dynamical systems. *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, 4:3925–3930, 26-May 1, 2004.

[9] S. Ekvall and D. Kragic. Learning and evaluation of the approach vector for automatic grasp generation and planning. In *Proceeding of the IEEE International Conference on Robotics and Automation*, 2007.

[10] R. Fikes and N. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, pages 189–205, 1971.

[11] M. Hersch, F. Guenter, S. Calinon, and A. Billard. Dynamical System Modulation for Robot Learning via Kinesthetic Demonstrations. *IEEE Transactions on Robotics*, 1, 2008. Accepted.

[12] D. Hinrichsen and A. Pritchard. *Mathematical Systems Theory*. Springer Berlin, 2000.

[13] A. Ijspeert, J. Nakanishi, and S. Schaal. Trajectory formation for imitation with nonlinear dynamical systems. *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, 2:752–757 vol.2, 2001.

[14] A. Ijspeert, J. Nakanishi, and S. Schaal. Learning rhythmic movements by demonstration using nonlinear oscillators. In *In Proceedings of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS2002*, pages 958–963, 2002.

[15] D. Kulic, W. Takano, and Y. Nakamura. Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains. *The International Journal of Robotics Research*, 27(7):761–784, 2008.

[16] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[17] J. Peters, M. Mistry, F. Udwadia, and S. Schaal. A unifying methodology for the control of robotic systems. *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2005.

[18] J. Tegin, S. Ekvall, D. Kragic, J. Wikander, and B. Iliev. *Intelligent Service Robotics*, chapter Demonstration-based learning and control for automatic grasping, pages 1861–2784. Springer Berlin / Heidelberg.

[19] E. Yoshida, C. Esteves, I. Belousov, J.-P. Laumond, T. Sakagushi, and K. Yokoi. Planning 3-d collision-free dynamic robotic motion through iterative reshaping. *IEEE Transactions on Robotics*, 24(5):1186–1198, 2008.