The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Imitation Learning of Globally Stable Non-Linear Point-to-Point Robot Motions using Nonlinear Programming

S. Mohammad Khansari-Zadeh and Aude Billard

Abstract— **This paper presents a methodology for learning arbitrary discrete motions from a set of demonstrations. We model a motion as a nonlinear autonomous (i.e. time-invariant) dynamical system, and define the sufficient conditions to make such a system globally asymptotically stable at the target. The convergence of all trajectories is ensured starting from any point in the operational space. We propose a learning method, called *Stable Estimator of Dynamical Systems (SEDS)*, that estimates parameters of a Gaussian Mixture Model via an optimization problem under non-linear constraints. Being time-invariant and globally stable, the system is able to handle both temporal and spatial perturbations, while performing the motion as close to the demonstrations as possible. The method is evaluated through a set of robotic experiments.**

I. INTRODUCTION

We use Programming by Demonstration (PbD) [1], also referred to as Learning by Imitation, to teach a robot how to move its limbs to perform a discrete, i.e. point-to-point motion[1] [2]. In PbD, an agent (e.g. human, robot, etc.) shows the robot a task a few times (usually between 3-5 times to make the task bearable for the trainer). To avoid addressing the correspondence problem [3], motions are demonstrated from the robot's point of view by the user guiding the robot passively through the task. In our experiments, this is done either by back-driving the robot or by teleoperating it using motion sensors (see Figure 1). We hence focus on the "what to imitate" problem and derive a means to extract the generic characteristics of the dynamics of the motion. Following [4], we assume that the relevant features of the movement, i.e. those to imitate, are the features that appear most frequently, i.e. the invariants across the demonstration. As a result, demonstrations should be such that they all contain the main features of the desired task, while exploring as much as possible the variations allowed by it.

In this paper, we present a learning procedure that extracts the relevant features of a desired task from demonstrations and formulate these features as a Dynamical System (DS) problem as follows: Consider a state variable $\xi \in \mathbb{R}^d$ that can be used to unambiguously define a discrete motion of a robotic system (e.g. ξ could be a robot's joint angles, the position of an arm's end-effector in Cartesian space, etc). Let the set of N given demonstrations $\{\xi^{t,n}, \dot{\xi}^{t,n}\}_{t=0,n=1}^{T^n,N}$ be instances of a global motion model governed by a first order autonomous Ordinary Differential Equation (ODE):

S.M. Khansari-Zadeh and A. Billard are with LASA Laboratory, School of Engineering, Ecole Polytechnique Federale de Lausanne, Switzerland {mohammad.khansari,Aude.Billard}@epfl.ch

[1]Motions in space stopping at a given target. For example, consider the standard "pick-and-place" task.

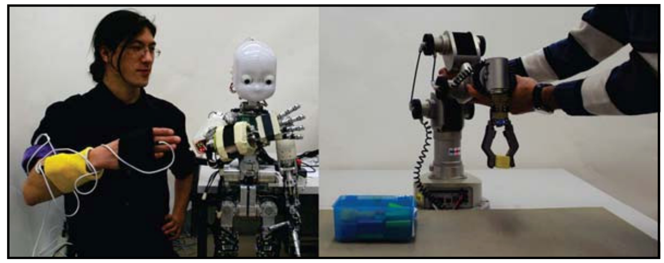


Fig. 1. Demonstrating motions by teleoperating a robot using motion sensors (**left**) or by back-driving it (**right**).

$$\dot{\xi} = f(\xi; \boldsymbol{\theta}) + \epsilon \tag{1}$$

where $f : \mathbb{R}^d \to \mathbb{R}^d$ is a nonlinear continuous and continuously differentiable function with a single equilibrium point $\dot{\xi}^* = f(\xi^*; \boldsymbol{\theta}) = 0$, $\boldsymbol{\theta}$ is the set of parameters of f, and ϵ represents zero mean Gaussian noise. The noise term ϵ encapsulates both inaccuracies in sensor measurements and errors resulting from imperfect demonstrations. Optimal values to estimate the noise-free model of $f(\xi; \boldsymbol{\theta})$ can be obtained based on the set of demonstrations using different statistical approaches[2]. We will further denote the noise-free estimate of f with $\hat{f}$ throughout the paper. Given an arbitrary starting point $\xi^0 \in \mathbb{R}^d$, the evolution of the robot's state, or motions can be computed by integrating from Eq. 1.

Use of such formulation is advantageous in that it enables a robot to adapt its trajectory "on-the-fly" in the face of perturbations. Perturbations may either be due to a sudden displacement of the target with respect to the robot (e.g. imprecise estimation of the target location, or if the target is moving itself or if the robot's end-effector is suddenly pushed away) or to delays in the execution of the task. We will refer to these two types of perturbation as spatial and temporal perturbations, respectively. Throughout this paper we choose to represent a motion in a kinematic coordinate system, and assume that there exists a low-level controller that converts kinematic variables into motor commands (e.g. force or torque).

Definition 1 *The function $\hat{f}$ is globally asymptotically stable at the target ξ^* if by starting from any point $\xi^0 \in \mathbb{R}^d$, the generated motion converges asymptotically to ξ^*, i.e.:*

$$\lim_{t\to\infty} \hat{f}(\xi^t; \boldsymbol{\theta}) = \xi^* \quad \forall \xi^0 \in \mathbb{R}^d \tag{2}$$

$\hat{f}$ is locally asymptotically stable if it converges to ξ^* only when ξ^0 is contained within a subspace $D \subset \mathbb{R}^d$.

[2]In fact, the noise term is eliminated from the model because it follows a zero mean distribution.

We will estimate $\hat{f}$ and learn its parameters $\boldsymbol{\theta}$ so that it optimizes accuracy of the reconstruction of the demonstration data under the constraint of being globally stable in $\mathbb{R}^d$ as per Definition 1. Constructing an estimate of the underlying dynamics of the motion as formulated in Eq. 1 is a non-trivial task. Next we briefly review literature on work done so far to estimate such a function.

II. Related Works

Statistical approaches to modeling robot motions have become increasingly popular as a means to deal with noise inherent in any mechanical system. They have proved to be interesting alternatives to classical control and planning approaches when the underlying model cannot be well estimated. Existing approaches to the *statistical* estimation of f in Eq. 1 use either Gaussian Process Regression (GPR) [5], Locally Weighted Projection Regression (LWPR) [6], or Gaussian Mixture Regression (GMR) [7], where the parameters of the Gaussian Mixture are optimized through Expectation Maximization (EM) [8]. GMR and GPR find an optimal model of $\hat{f}$ by maximizing the likelihood that the complete model represents the data well, while LWPR minimizes the mean-square error between the estimates and the data (for a more detailed comparison and discussion on these methods refer to [9])

Because all of the aforementioned methods do not optimize under the constraint of making the system stable at the attractor, they are not guaranteed to result in a stable estimate of the motion. In practice, they all fail to ensure the global stability and most of the time the local stability of $\hat{f}$ (see Definition 1), and thus may converge to a spurious attractor or completely miss the target (diverging/unstable behavior) even for relatively simple 2 dimensional dynamics. These errors are due to the fact that there is yet no theoretical solution for ensuring stability of arbitrary non-linear autonomous DS [10]. Figure 2 illustrates an example of unstable estimation of a non-linear DS using the above three methods for learning a two dimensional motion. Figure 2(a) presents the stability analysis of the dynamics learnt with GMR. Here in the narrow regions around demonstrations, the trajectories converge to a spurious attractor just next to the target. In other parts of space, they either converge to other spurious attractors far from the target or completely diverge from it. Figure 2(b) shows the obtained results from LWPR. All trajectories inside the black boundaries converge to a spurious attractor. Outside of these boundaries, the velocity is always zero (a region of spurious attractors) hence motion stops once it crosses these boundaries or it does not move when it initializes there. In Figure 2(c), we see that while for GPR trajectories converge to the target in a narrow area close to demonstrations, they are attracted to spurious attractors outside that region.

In all these examples, the critical concern is that there is no generic theoretical solution to determine beforehand for an arbitrary non-linear function whether a trajectory will lead to a spurious attractor, to infinity, or to the desired attractor. Thus, it is necessary to conduct numerical stability analysis

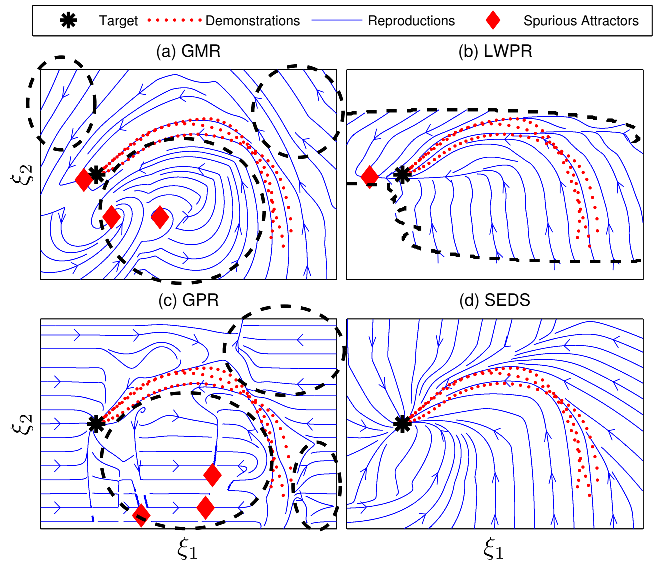


Fig. 2. An example of two-dimensional dynamics learned from three demonstrations using four different methods: GMR, LWPR, and GPR, and SEDS. Regions associated to spurious attractors are isolated with dashed lines. For further information please refer to the text.

to locate the region of attraction of the desired target which may never exist, or be very narrow. Besides, finding this region of attraction becomes computationally costly and a non-trivial task in higher dimensions.

Among works done on DS, [2], [11], [12], [13] ensure the stability of $\hat{f}$ by modulating it with an inherently stable uni-dimensional linear dynamics. The modulation between $\hat{f}$ and the stable linear dynamics is controlled with a time dependent phase variable that exponentially decreases as time passes by. While this is an easy means to ensure global stability, it has two drawbacks: 1) the implicit time-dependency requires a heuristic to re-scale the planned trajectories in time when an unexpected disturbance causes a delay in the reproduction (so-called temporal perturbations); 2) stability resulting from the underlying linear DS is ensured for each dimension separately, and therefore requires the re-synchronization of all DSs, especially when one dimension (e.g. one joint) is perturbed but not the others [9].

In [9], we proposed an iterative method to construct a mixture of Gaussian so as to ensure asymptotic stability at the target. However, this method solely ensures *local* asymptotic stability in a narrow region D (while global stability outside D was ensured through a fixed linear DS). It also relies on determining numerically the region D, which is computationally intensive, hence it is difficult to apply the method to high-dimensional problems.

In this paper we present a method, called *Stable Estimator of Dynamical Systems (SEDS)*, that computes optimal values of $\boldsymbol{\theta}$ while ensuring that $\hat{f}$ is globally stable in $\mathbb{R}^d$. Figure 2(d) shows the results obtained from this method. As can be seen, the system is globally asymptotically stable and all trajectories converge to the target. This ensures that the task can be successfully accomplished starting from any point in the operational space without any need for re-indexing or re-scaling. Inspired by human body motion, we show how such a model can be used to integrate different motions into one single dynamical system. Such modeling is especially useful when one desires to execute a single task in a different manner starting from different areas in space, mainly to consider task constraints, to avoid robot's joint limits, etc. Next we formalize the method presented in this paper.

III. Multivariate Motion Learning

In this section we proceed in three steps: First in Subsection III-A we restate the problem introduced in Eq. 1 in the statistical framework. Then in Subsection III-B we derive the necessary and sufficient conditions to guarantee the global stability of an arbitrary function $\hat{f}$. Then in Subsection III-C we propose an optimization problem to compute optimal values of the parameters $\boldsymbol{\theta}^*$ while satisfying stability conditions.

A. Problem Formulation

We use a probabilistic framework and model $\hat{f}$ via a finite mixture of Gaussian functions. Using such an approach, unknown parameters of $\hat{f}$ become priors π^k, means μ^k and covariance matrices Σ^k of $k = 1..K$ Gaussian functions (i.e. $\theta^k = \{\pi^k, \mu^k, \Sigma^k\}$ and $\boldsymbol{\theta} = \{\theta^1..\theta^K\}$). The mean and the covariance matrix of a Gaussian k are defined by:

$$\mu^k = \begin{pmatrix} \mu_\xi^k \\ \mu_{\dot{\xi}}^k \end{pmatrix} \quad , \quad \Sigma^k = \begin{pmatrix} \Sigma_\xi^k & \Sigma_{\xi\dot{\xi}}^k \\ \Sigma_{\dot{\xi}\xi}^k & \Sigma_{\dot{\xi}}^k \end{pmatrix} \tag{3}$$

Given a set of N demonstrations $\{\xi^{t,n}, \dot{\xi}^{t,n}\}_{t=0,n=1}^{T^n,N}$, each recorded point in the trajectories $[\xi^{t,n}, \dot{\xi}^{t,n}]$ is associated with a probability density function $\mathcal{P}(\xi^{t,n}, \dot{\xi}^{t,n})$:

$$\mathcal{P}(\xi^{t,n}, \dot{\xi}^{t,n}) = \sum_{k=1}^{K} \pi^k \mathcal{N}^k(\xi^{t,n}, \dot{\xi}^{t,n}; \theta^k) \begin{cases} \forall n \in 1..N \\ t \in 0..T^n \end{cases} \tag{4}$$

where $\mathcal{N}^k(\xi^{t,n}, \dot{\xi}^{t,n}; \theta^k)$ is given by:

$$\mathcal{N}^k(\xi^{t,n}, \dot{\xi}^{t,n}; \theta^k) = \frac{1}{\sqrt{(2\pi)^{2d}|\Sigma_\xi^k|}} e^{-\frac{1}{2}([\xi^{t,n},\dot{\xi}^{t,n}]-\mu^k)^T(\Sigma^k)^{-1}([\xi^{t,n},\dot{\xi}^{t,n}]-\mu^k)} \tag{5}$$

Taking the posterior mean estimate of $\mathcal{P}(\dot{\xi}|\xi)$ yields (see [14]):

$$\hat{\dot{\xi}} = \sum_{k=1}^{K} \frac{\pi^k \mathcal{P}(\xi; \mu_\xi^k, \Sigma_\xi^k)}{\sum_{i=1}^{K} \pi^i \mathcal{P}(\xi; \mu_\xi^i, \Sigma_\xi^i)} (\mu_{\dot{\xi}}^k + \Sigma_{\dot{\xi}\xi}^k (\Sigma_\xi^k)^{-1} (\xi - \mu_\xi^k)) \tag{6}$$

The resulting nonlinear function $\hat{f}(\xi; \boldsymbol{\theta})$ from Eq. 6 usually contains several spurious attractors or limit cycles (see Figure 2). Thus parameters $\boldsymbol{\theta}$ need to be determined that lead to an estimate $\hat{f}(\xi; \boldsymbol{\theta})$ with a single asymptotically stable attractor.

B. Stability Analysis

Stability analysis of dynamical systems is a broad subject in the field of dynamics and control, and can generally be divided into two branches: linear and nonlinear systems. While the stability of linear dynamics has been well studied [10] and can be ensured solely by requiring that the eigenvalues of the system are negative, stability analysis of nonlinear dynamical systems is still an open questions and theoretical solutions exist only for particular cases. In this paper, we obtain the sufficient conditions to ensure the global stability of a series of nonlinear dynamical systems given by Eq. 6.

We start by simplifying the notation of Eq. 6 through a change of variable. Let us define:

$$\begin{cases} A^k = \Sigma_{\dot{\xi}\xi}^k (\Sigma_\xi^k)^{-1} \\ b^k = \mu_{\dot{\xi}}^k - A^k \mu_\xi^k \\ h^k(\xi) = \frac{\pi^k \mathcal{P}(\xi; \mu_\xi^k, \Sigma_\xi^k)}{\sum_{i=1}^{K} \pi^i \mathcal{P}(\xi; \mu_\xi^i, \Sigma_\xi^i)} \end{cases} \tag{7}$$

Substituting Eq. 7 into Eq. 6 yields:

$$\hat{\dot{\xi}} = \hat{f}(\xi; \boldsymbol{\theta}) = \sum_{k=1}^{K} h^k(\xi)(A^k \xi + b^k) \tag{8}$$

First observe that $\hat{f}$ is now expressed as a non-linear sum of linear dynamical systems. The nonlinear weighting terms $h^k(\xi)$ in Eq. 8, where $0 < h^k(\xi) \leq 1$, give a measure of the relative influence of each Gaussian locally. Beware that the intuition that the nonlinear function $\hat{f}(\xi; \boldsymbol{\theta})$ should be stable if all eigenvalues of matrices A^k are negative $k = 1..K$, is not true. Here is a simple example in 2D that illustrates why this is not the case and also why estimating stability of non-linear DS even in 2D is non-trivial.

Example: Consider the parameters of a model with two Gaussian functions to be:

$$\begin{cases} \Sigma_\xi^1 = \Sigma_\xi^2 = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \\ \Sigma_{\dot{\xi}\xi}^1 = \begin{bmatrix} -3 & -30 \\ 3 & -3 \end{bmatrix}, \ \Sigma_{\dot{\xi}\xi}^2 = \begin{bmatrix} -3 & 3 \\ -30 & -3 \end{bmatrix} \\ \mu_\xi^1 = \mu_\xi^2 = \mu_{\dot{\xi}}^1 = \mu_{\dot{\xi}}^2 = \mathbf{0} \end{cases} \tag{9}$$

Using Eq. 7 we have:

$$\begin{cases} A^1 = \begin{bmatrix} -1 & -10 \\ 1 & -1 \end{bmatrix}, \ A^2 = \begin{bmatrix} -1 & 1 \\ -10 & -1 \end{bmatrix} \\ b^1 = b^2 = \mathbf{0} \end{cases} \tag{10}$$

The eigenvalues of both matrices A^1 and A^2 are complex with values $-1 \pm 3.16i$. In other words, each matrix determines a stable system. However, the nonlinear combination of the two matrices as per Eq. 8 is stable solely when $\xi_2 = \xi_1$, and is unstable in $\mathbb{R}^d \setminus \{(\xi_2, \xi_1) | \xi_2 = \xi_1\}$ (see Figure 3).

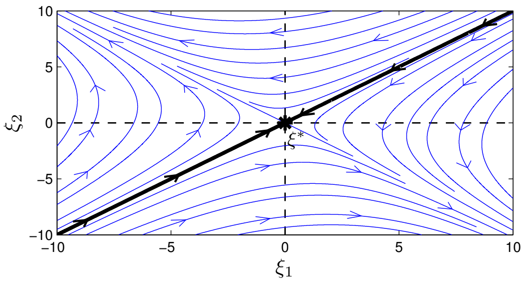


Fig. 3. An example showing that stability of each matrix A^k does not ensure the stability of the whole dynamics. Here the system is stable only for points on the line $\xi_2 = \xi_1$ (drawn in black).

Theorem 1: An arbitrary non-linear dynamical system $\dot{\xi} = \hat{f}(\xi; \boldsymbol{\theta})$ given by Eq. 8 is globally asymptotically stable at the target ξ^* in $\mathbb{R}^d$ if:

$$\begin{cases} \textbf{(a)}\ b^k = -A^k \xi^* \\ \textbf{(b)}\ \frac{A^k + (A^k)^T}{2} \text{ are negative definite} \end{cases} \quad \forall k = 1..K \tag{11}$$

where $(A^k)^T$ corresponds to the transpose of A^k. For the definition of a *negative definite matrix* refer to appendix I.

Proof: See appendix II.

Eq. 11 provides us with sufficient conditions to make an arbitrary nonlinear function given by Eq. 8 globally asymptotically stable at the target ξ^* (we further phrase this equation as *stability conditions*). Such a model is advantageous in that it ensures starting from any point in the space, the trajectory (of, e.g., a robot arm's end-effector) always converges to the target.

C. Learning Algorithm

Section III-B provided us with sufficient conditions on $\boldsymbol{\theta}$ whereby the estimate $\hat{f}(\xi; \boldsymbol{\theta})$ is globally asymptotically stable at the target. It remains now to determine a procedure for computing unknown parameters of Eq. 8, i.e. $\boldsymbol{\theta} = \{\pi^1..\pi^K; \mu^1..\mu^K; \Sigma^1..\Sigma^K\}$ such that the resulting model is globally asymptotically stable. In this section we propose a learning algorithm, called *Stable Estimator of Dynamical Systems (SEDS)*, that computes values of $\boldsymbol{\theta}$ by solving an optimization problem under the constraint of ensuring the model's global asymptotic stability. Given a set of N demonstrations $\{\xi^{t,n}, \dot{\xi}^{t,n}\}_{t=0,n=1}^{T^n,N}$, we compute the optimal values of $\boldsymbol{\theta}$ by solving:

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \frac{1}{N} \sum_{n=1}^{N} \sum_{t=0}^{T^n} \Big((\hat{\xi}^{t,n}(\boldsymbol{\theta}) - \xi^{t,n})^2 + \\ + (\hat{\dot{\xi}}^{t,n}(\boldsymbol{\theta}) - \dot{\xi}^{t,n})^2 \Big) \tag{12}$$

subject to

$$\begin{cases} \textbf{(a)}\ b^k = -A^k \xi^T \\ \textbf{(b)}\ \frac{1}{2}(A^k + (A^k)^T) \text{ are negative definite} \\ \textbf{(c)}\ \Sigma^k_{\xi} \text{ are positive definite} \\ \textbf{(d)}\ 0 < \pi^k \leq 1 \\ \textbf{(e)}\ \sum_{k=1}^{K} \pi^k = 1 \end{cases} \quad \forall k \in 1..K \tag{13}$$

where $\hat{\dot{\xi}}^{t,n}(\boldsymbol{\theta}) = \hat{f}(\hat{\xi}^{t,n}(\boldsymbol{\theta}); \boldsymbol{\theta})$ are computed directly from Eq. 8, and $\hat{\xi}^{t,n}(\boldsymbol{\theta}) = \sum_{i=0}^{t} \hat{\dot{\xi}}^{i,n}(\boldsymbol{\theta}) dt$ generate an estimate of the corresponding demonstrated trajectory ξ^n by starting from the same initial points as were demonstrated, i.e. $\hat{\xi}^{0,n} = \xi^{0,n}$, $\forall n \in 1..N$.

Eq. 12-13 correspond to a Non-linear Programming (NLP) problem [15] that can be solved using different optimization techniques such as Newton-Like algorithms [15], Dynamic Programming [16], etc. In this paper we use a quasi-Newton method to solve the optimization problem [15]. The first and second terms inside the integral of Eq. 12 force the solver to optimize reproduction in both position and velocity spaces, but if necessary, a weighting factor could be used to set the influence of each term. The first two constraints in Eq. 13 are stability conditions from Section III-B. The last three constraints are imposed by the nature of the Gaussian Mixture Model to ensure that Σ^k_{ξ} are positive definite matrices, priors π^k are positive scalars smaller or equal than one, and sum of all priors is equal to one. It can be easily shown that a feasible solution to this NLP problem always exists. For example, a general initialization value for $\boldsymbol{\theta}$ that satisfies such constraint could be:

$$\begin{cases} \Sigma^k_{\xi} = \boldsymbol{I}, \quad \Sigma^k_{\dot{\xi}\xi} = -\boldsymbol{I} & \Rightarrow A^k = -\boldsymbol{I} \\ \mu^k_{\xi} = \mu^k_{\dot{\xi}} = \boldsymbol{0} & \Rightarrow b^k = \boldsymbol{0} \end{cases} \tag{14}$$

where $\boldsymbol{I}$ denotes the identity matrix. Starting from these initial values, the solver tries to optimize the value of $\boldsymbol{\theta}$ such that the cost function J is minimized. However since the proposed NLP problem is non-convex, based on the choice of initial values for the parameters and the complexity of the represented motion, the optimization may converge to a local minimum with a poor estimate of the actual dynamics. In section IV we show that despite the fact that the global minimum may not be attained, the algorithm is able to accurately learn a wide variety of robotic motions.

Note that the total number of parameters is 1, $2d$ and $d(2d+1)$ to estimate the priors, means μ^k and covariance matrices Σ^k of each Gaussian (by construction the covariance matrix is symmetric), respectively. However, the number of parameters can be reduced since the constraints given by Eq. 13-(a) provide an explicit formulation to compute $\mu^k_{\dot{\xi}}$ from other parameters (i.e. μ^k_{ξ}, Σ^k_{ξ}, and $\Sigma^k_{\dot{\xi}\xi}$). Furthermore when constructing $\hat{f}$, the term $\Sigma^k_{\dot{\xi}}$ is not used and thus can be omitted during the optimization. Taking into account all of the above, the total number of learning parameters reduces to $K(1 + \frac{3}{2}(d + d^2))$, where K is the number of Gaussian functions. In other words, learning grows linearly with the number of Gaussians and quadratically with the dimension. In comparison, for the same number of Gaussian functions K, the number of parameters in the proposed method is fewer than in GMR and LWPR[3], and the retrieval time (estimation time of $\dot{\xi}$, given the input ξ) is low and in the same order of GMR and LWPR.

[3]The number of parameters in GMR and LWPR is $K(1+3d+2d^2)$ and $\frac{7}{2}K(d+d^2)$ respectively.

IV. Experimental Evaluations

We conducted a set of robotic experiments to evaluate the performance of SEDS in teaching a robot how to perform a motion[4]. In the first experiment, a 6 degrees of freedom (DOF) industrial Katana-T arm is taught to how to put small blocks into a container[5] (see Figure 4). We use Cartesian coordinates system to represent the motion (the axes ξ_1, ξ_2, and ξ_3 correspond respectively to x_1, x_2, and x_3 in the Cartesian coordinates system). In all of the examples, the coordinate system is attached to the target. In order to have human-like motions, the learnt model should be able to generate trajectories with both similar position and velocity profile to the demonstrations. In this experiment, the task was shown to the robot six times, and was learnt by solving the optimization problem given by Eq. 12-13 using $K = 6$ Gaussian functions. Figure 4(a) illustrates the obtained results for generated trajectories starting from different points in the task space. The direction of motion is indicated by arrows. All reproduced trajectories are able to follow the same dynamics (i.e. having the similar position and velocity profile) as demonstrations.

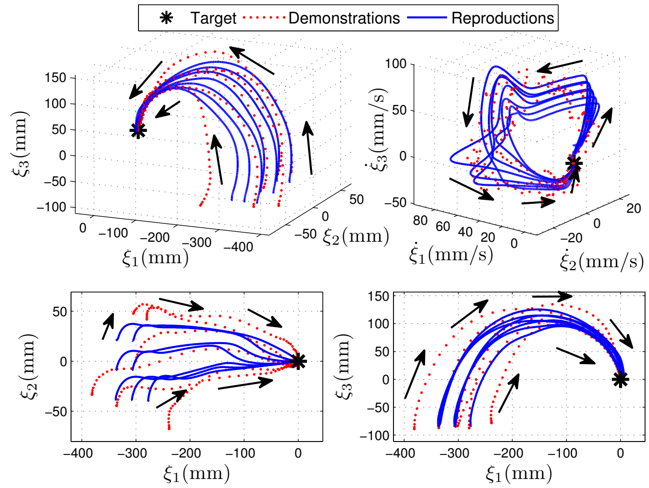
Adaptation to Perturbation: Figure 4(b) shows the robustness of the model to external perturbations. In this graph, the original trajectory is plotted in thin blue line. The thick black line represents the generated trajectory for the case where the target is displaced at $t = 1.5$ second. Having defined the motion as Dynamical Systems, adaptation to a new target position can be done instantly.

While convergence to the target is always ensured from conditions given by Eq. 11, due to the lack of information for points far from demonstrations, the model may reproduce some trajectories that are not consistent with the usual way of doing the task. For example, consider Figure 5-Top, i.e. when the robot starts the motion from the left-side of the target, it first goes around the container and then approaches the target from its right-side. This behavior may not be optimal as one expects the robot to follow the shortest path to the target and reach it from the same side as the one it departed from. However, such a result is inevitable since the information given by the teacher is incomplete, and thus the inference for points far from the demonstrations are not reliable. In order to improve the task's execution, it is necessary to provide the robot with more demonstrations (information) over regions not covered before. By showing the robot more demonstrations and re-training the model with the new data, the robot is able to successfully accomplish the task (see Figure 5-Bottom).

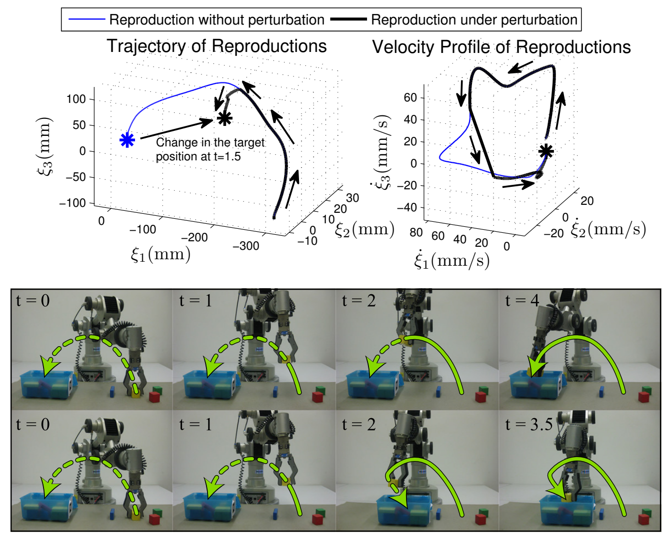
The next three experiments consisted of having the 7DOF right arm of the humanoid robot iCub perform complex motions, containing several non-linearities (i.e.sucessive curvatures) in both position and velocity profile. As before, we use Cartesian coordinate systems to represent these motions.

[4]The main steps of the method presented in this paper and the experimental results are shown in the accompanying video.

[5]The robot is only taught how to move blocks. The problem of grasping the blocks is out of the scope of this paper. Throughout the experiments, we pose the blocks such that they can be easily grasped by the robot.



(a) The ability of the model to reproduce similar trajectories starting from different points in the space.



(b) The ability of the model to on-the-fly adapt its trajectory to a change in the target's position

Fig. 4. The Katana-T arm performing the experiment of putting small blocks into a container. Please see the text for further information.

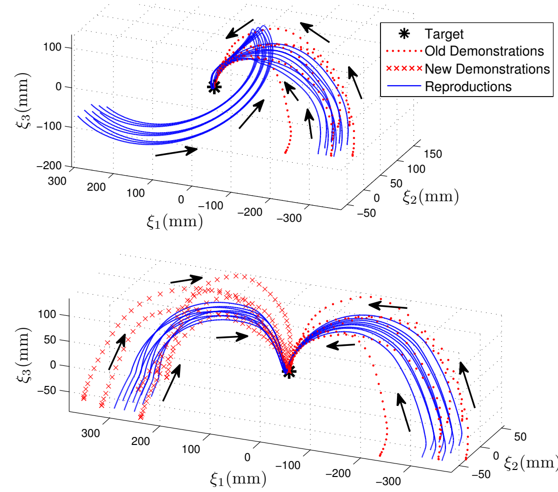


Fig. 5. Improving the task's execution by adding more data for regions far from demonstrations.

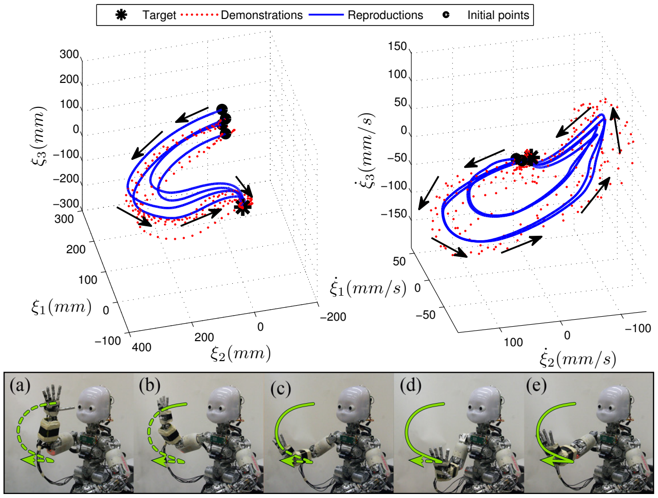


Fig. 6. The first experiment with the iCub. The robot does a semi-spiral motion toward its right-side, and at the bottom of the spiral it stretches its hand forward completely.

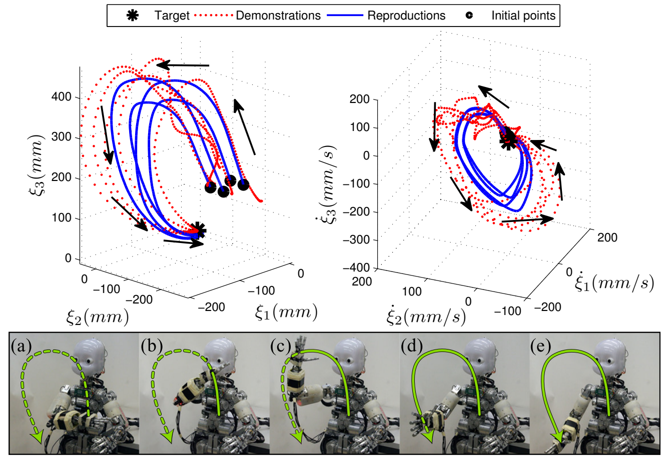


Fig. 7. The second experiment with the iCub. The robot does a semi-circle motion upward and then brings down its arm completely.

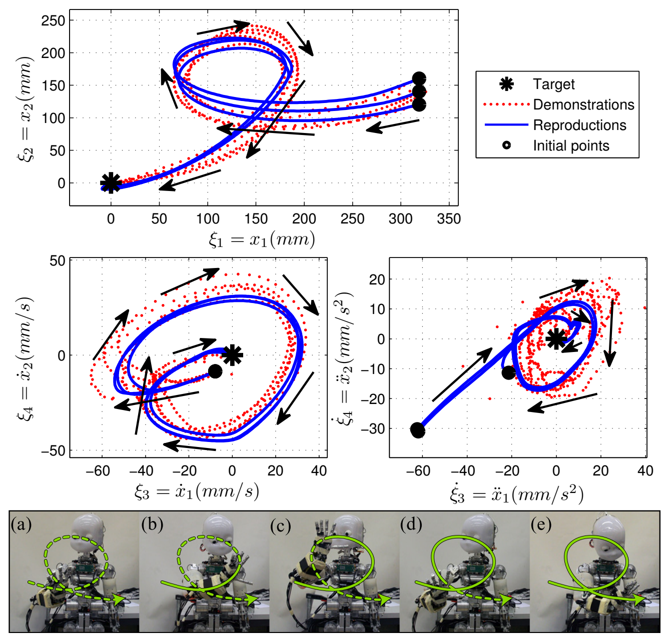


Fig. 8. The third experiment with the iCub. The robot performs a self-intersecting motion.

Figure 6 illustrates the results from the first task, where the iCub starts the motion in front of its face. It then does a semi-spiral motion toward its right-side, and finally at the bottom of the spiral it stretches its hand forward completely. In the second task, the iCub starts the motion close to its left forehand. Then it does a semi-circle motion upward and finally brings down its arm completely (see Figure 7). In the third motion, the iCub performs a loop motion with its right hand, where the motion lies in a vertical plane and thus contains a self intersection point (see Figure 8). Critical to such kinds of motion is the ambiguity in the correct direction of velocity at the intersection point if the model's variable ξ considered to be only the cartesian position (i.e. $\xi = x \Rightarrow \dot{\xi} = \dot{x}$). This ambiguity usually results in reproductions skipping the loop part of the motion. However in this example, defining ξ such that it includes both the cartesian position and velocity (i.e. $\xi = [x; \dot{x}] \Rightarrow \dot{\xi} = [\dot{x}; \ddot{x}]$) can solve this ambiguity. The three experiments were learnt using 5, 4 and 7 Gaussian functions, respectively. In all three experiments the robot is able to successfully follow the demonstrations and to generalize the motion for several trajectories with different starting points.

We also further examine SEDS in a library of 20 different human handwriting motions recorded using a Tablet PC, and compare it against our previous method *Binary Merging* (BM) [9] and those of four alternative methods GMR, LWPR, GPR, and DMP[6]. Figure 9 shows the results for 5 out of 20 motions. Quantitative performance comparison of all the six methods is given in Table I. In this paper, we only focus on the comparison between SEDS and our previous approach BM. The detailed comparison between alternative approaches and BM is provided by [9]. Here, we use the same accuracy measurement proposed by [9], with which a method's accuracy in estimating the overall dynamics of the underlying model $\hat{f}$ is quantified by measuring the discrepancy between the direction and magnitude of the estimated and observed velocity vectors for all training data points.

$$\bar{e} = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{T^n+1}\sum_{t=0}^{T^n} r\left(\frac{1-\dot{\xi}^{t,n}.\hat{\dot{\xi}}^{t,n}}{\|\dot{\xi}^{t,n}\|\|\hat{\dot{\xi}}^{t,n}\|+\epsilon}\right)^2 + q\frac{(\dot{\xi}^{t,n}-\hat{\dot{\xi}}^{t,n})^T(\dot{\xi}^{t,n}-\hat{\dot{\xi}}^{t,n})}{\|\dot{\xi}^{t,n}\|\|\dot{\xi}^{t,n}\|+\epsilon}\right)^{\frac{1}{2}} \tag{15}$$

where r and q are positive scalars that weigh the relative influence of each factor, and ϵ is a very small positive scalar.

Regarding Table I, while both BM and SEDS are able to learn the demonstrated dynamics with relatively similar accuracy, each method has its own advantages and disadvan-

[6]Because GMR requires a fixed set of Gaussians, we used the same number of Gaussians as the one selected for the proposed approach. Similarly, the number of Gaussians used in DMP was initialized with that found by LWPR for the same task.

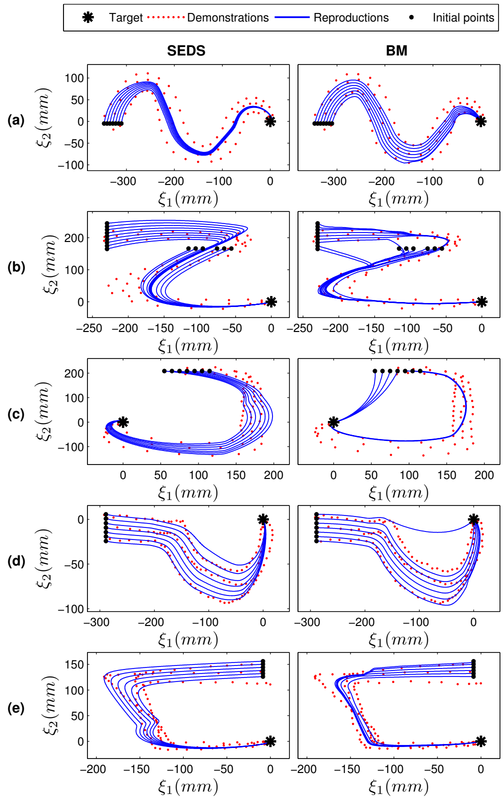


Fig. 9. Performance comparison of the new approach against BM through a library of 20 different handwriting motions. Only 5 of 20 examples are shown here.

TABLE I

PERFORMANCE COMPARISON OF SEDS AGAINST ALTERNATIVE APPROACHES IN LEARNING 20 HUMAN HANDWRITING MOTIONS

Method	Is stability ensured?	Average / Range of error $\bar{e}$	Average / Range of No. of Parameters
SEDS	Yes	0.23 / [0.14-0.38]	65 / [26 - 91]
BM	Yes	0.21 / [0.14-0.53]	98 / [56 - 196]
DMP	Yes	0.88 / [0.37 - 2.02]	92 / [29 - 182]
GMR	No	0.16 / [0.10 - 0.37]	75 / [30 - 105]
LWPR	No	0.22 / [0.08 - 0.49]	609 / [168 - 1239]
GPR	No	0.05 / [0.03 - 0.09]	2190 / [1806 - 3006]†

† GPR's learning parameter is of order of $d(d+1)$; however, it also requires keeping all training datapoints to estimate the output $\dot{\xi}$. Hence the total number of required parameters is $d(d+1)+2n*d$, where n is the total number of datapoints.

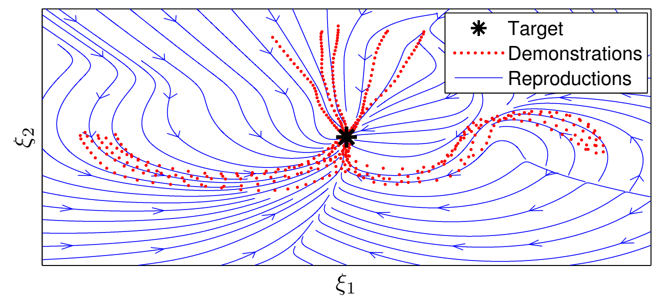


Fig. 10. Embedding different ways of performing a task in one single model. The robot follows an arc, a sine, or a straight line starting from different points in the workspace.

tages. The new method outperforms the previous approach in that it can better generalize the motion for trajectories far from demonstrations, while in BM trajectories go directly toward the target if they started outside of the region enclosed by demonstrations (see Figure 9-b,c, and d). On the other hand, BM offers more flexibility in learning complex motions since it unfolds a motion into a set of discrete joint-wise partitions and ensures the stability conditions locally in each partition. In contrast, SEDS is more constraining by the fact that it tries to fit a motion with a single globally stable dynamics. Moreover, BM is able to determine the (locally) optimum number of Gaussian functions automatically while the proposed method needs a user to predefine it beforehand.

In the last experiment, we show how the proposed method can be used to integrate different motions into one single dynamics (see Figure 10). In this experiment, the task is learnt using $K = 7$ Gaussian functions, and demonstrations data are recorded using a Tablet-PC. Regarding Figure 10, the robot is able to approach to the target following an arc, a sine, or a straight line path starting from the left, right, or top-side of the task space, respectively. While reproductions locally follow the desired motion around each set of demonstrations, they smoothly switch from one motion to another in areas between demonstrations. The proposed method offers a simple but reliable procedure to teach a robot different ways of performing a task.

V. CONCLUSIONS AND FUTURE WORK

In this paper we present a method in which arbitrary discrete motions are modeled as nonlinear autonomous dynamics. The sufficient conditions to make the motion globally stable are provided, and a learning method, called SEDS, is proposed to estimate the model's parameters. This method enables a robot to perform a task starting from any point in the operational space while keeping the motion as similar as possible to the demonstrations. The obtained results illustrate the ability of SEDS in handling both temporal and spatial perturbations. Using SEDS, the robot is able to on-the-fly adapt its trajectory to a change in the target position.

However, one should emphasize that the optimization problem given by Eq. 12 is non-convex, and convergence to the global minimum cannot be ensured. But, often the obtained approximation of the minimum is good enough to accurately model the represented motion. Furthermore, similar to GMM, SEDS requires a user to predefine the number of Gaussian functions, i.e. K. While it still remains an open question, in practice we noticed that the Bayesian Information Criterion (BIC) [17] can be used as a relatively good estimate to the optimum number of K. Finally, an assumption made in this paper is that represented motions can be modeled with a first order time-invariant ODE. While the nonlinear function given by Eq. 8 is able to model a wide variety of motions, the method cannot be used for some special cases violating this assumption. Most of the time, this limitation can be tackled through a change of variable (for example see Figure 8).

VI. ACKNOWLEDGMENTS

This work was supported by the European Commission through the EU Project AMARSI (FP7-ICT-248311). The authors kindly thank Jan Peters for his insightful comments during the development of this work, and Eric Sauser for preparing the experimental setup with the iCub robot.

APPENDIX I
POSITIVE AND NEGATIVE DEFINITE MATRIX

A $d \times d$ real symmetric matrix A is positive definite if $\xi^T A \xi > 0$ for all non-zero vectors $\xi \in \mathbb{R}^d$, where ξ^T denotes the transpose of ξ. We call A a negative definite matrix if $\xi^T A \xi < 0$. For a non-symmetric matrix, A is positive (negative) definite if and only if its symmetric part $\tilde{A} = (A + A^T)/2$ is positive (negative) definite.

APPENDIX II
PROOF OF THEOREM 1

We start the proof by first recalling the standard Lyapunov Stability theorem [10]:

Lyapunov Stability Theorem: An arbitrary function $\dot{\xi} = \hat{f}(\xi; \boldsymbol{\theta})$ is asymptotically stable at the point ξ^*, if a continuous and continuously differentiable Lyapunov function $V(\xi)$ can be found such that:

$$\begin{cases} (\mathbf{a})\ V(\xi) > 0 & \forall \xi \in \mathbb{R}^d \quad \& \quad \xi \neq \xi^* \\ (\mathbf{b})\ \dot{V}(\xi) < 0 & \forall \xi \in \mathbb{R}^d \quad \& \quad \xi \neq \xi^* \\ (\mathbf{c})\ V(\xi^*) = 0 \quad \& \quad \dot{V}(\xi^*) = 0 \end{cases} \tag{16}$$

Note that $\dot{V}$ is a function of both ξ and $\dot{\xi}$. However, since $\dot{\xi}$ can be directly expressed in terms of ξ using Eq. 8, one can finally infer that $\dot{V}$ only depends on ξ.

Consider a Lyapunov function $V(\xi)$ of the form:

$$V(\xi) = \frac{1}{2}(\xi - \xi^*)^T(\xi - \xi^*) \qquad \forall \xi \in \mathbb{R}^d \tag{17}$$

First observe that $V(\xi)$ is a quadratic function and hence satisfies condition Eq. 16.a. Condition given by Eq. 16.b follows from taking the first derivative of $V(\xi)$ w.r.t. time, we have:

$$\begin{aligned}
\dot{V}(\xi) &= \frac{dV}{dt} = \frac{dV}{d\xi}\frac{d\xi}{dt} \\
\Rightarrow &= \frac{1}{2}\frac{d}{d\xi}\left((\xi - \xi^*)^T(\xi - \xi^*)\right)\dot{\xi} \\
\Rightarrow &= (\xi - \xi^*)^T\dot{\xi} = (\xi - \xi^*)^T\hat{f}(\xi; \boldsymbol{\theta}) \\
\Rightarrow &= (\xi - \xi^*)^T \underbrace{\sum_{k=1}^{K} h^k(\xi)(A^k\xi + b^k)}_{=\dot{\xi} \text{ (see Eq. 8)}} \\
\Rightarrow &= (\xi - \xi^*)^T \sum_{k=1}^{K} h^k(\xi)(A^k(\xi - \xi^*) + \underbrace{A^k\xi^* + b^k}_{=0 \text{ (see Eq. 11-a)}}) \\
\Rightarrow &= (\xi - \xi^*)^T \sum_{k=1}^{K} h^k(\xi)A^k(\xi - \xi^*) \\
\Rightarrow &= \sum_{k=1}^{K} \underbrace{h^k(\xi)}_{h^k > 0} \underbrace{(\xi - \xi^*)^T A^k (\xi - \xi^*)}_{<0 \text{ (see Eq. 11-b)}} \qquad (18) \\
\Rightarrow &< 0 \qquad \forall \xi \neq \xi^*
\end{aligned}$$

Conditions given by Eq. 16.c are satisfied when substituting $\xi = \xi^*$ into Eqs. 17 and 18:

$$V(\xi^*) = \frac{1}{2}(\xi - \xi^*)^T(\xi - \xi^*)\Big|_{\xi = \xi^*} = 0 \tag{19}$$

$$\dot{V}(\xi^*) = \sum_{k=1}^{K} h^k(\xi)(\xi - \xi^*)^T A^k (\xi - \xi^*)\Big|_{\xi = \xi^*} = 0 \tag{20}$$

Therefore, an arbitrary ODE function $\dot{\xi} = \hat{f}(\xi; \boldsymbol{\theta})$ given by Eq. 8 is globally asymptotically stable if conditions of Eq. 11 are satisfied.

REFERENCES

[1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, *Handbook of Robotics*. MIT Press, 2008, ch. Robot Programming by Demonstration.

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

[3] K. Dautenhahn and C. L. Nehaniv, "The agent-based perspective on imitation," pp. 1–40, 2002.

[4] A. Billard, Y. Epars, S. Calinon, G. Cheng, and S. Schaal, "Discovering optimal imitation strategies," *Robotics and Autonomous Systems, Special Issue: Robot Learning from Demonstration*, vol. 47, pp. 69–77, 2004.

[5] C. Rasmussen and C. Williams, *Gaussian processes for machine learning*. Springer, 2006.

[6] S. Vijayakumar and S. Schaal, "Locally weighted projection regression: An o(n) algorithm for incremental real time learning in high dimensional space," in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, 2000.

[7] 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*, vol. 37, no. 2, pp. 286–298, 2007.

[8] A. Dempster and N. L. D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society B*, vol. 39, no. 1, pp. 1–38, 1977.

[9] S.-M. Khansari-Zadeh and A. Billard, "BM: An iterative algorithm to learn stable non-linear dynamical systems with gaussian mixture models," in *Proceeding of the International Conference on Robotics and Automation (ICRA)*, 2010, pp. 2381–2388.

[10] J. Slotine and W. Li, *Applied Nonlinear Control*. Prentice-Hall, 1991.

[11] M. Hersch, F. Guenter, S. Calinon, and A. Billard, "Dynamical system modulation for robot learning via kinesthetic demonstrations," *IEEE Transactions on Robotics*, pp. 1463–1467, 2008.

[12] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Philosophical Transactions: Biological Sciences (The Royal Society)*, no. 1431, pp. 537–547, 2003.

[13] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *International conference on robotics and automation (ICRA)*, 2009.

[14] D. Cohn and Z. Ghahramani, "Active learning with statistical models," *Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.

[15] M. S. Bazaraa, H. Sherali, and C. Shetty, *Nonlinear programming: Theory and Algorithms*, 3rd Edition, Ed. John Wiley & Sons, 2006.

[16] R. D. Robinett, *Applied dynamic programming for optimization of dynamical systems*. SIAM, 2005.

[17] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, pp. 461–464, 1978.