

Learning Partially Contracting Dynamical Systems from Demonstrations

Harish Ravichandar

Department of Electrical Engineering and Computer Sciences
University of Connecticut United States
harish.ravichandar@uconn.edu

Iman Salehi

Department of Electrical Engineering and Computer Sciences
University of Connecticut United States
iman.salehi@uconn.edu

Ashwin Dani

Department of Electrical Engineering and Computer Sciences
University of Connecticut United States
ashwin.dani@uconn.edu

Abstract: An algorithm for learning the dynamics of point-to-point motions from demonstrations using an autonomous nonlinear dynamical system, named contracting dynamical system primitives (CDSP), is presented. The motion dynamics are approximated using a Gaussian mixture model (GMM) and its parameters are learned subject to constraints derived from partial contraction analysis. Systems learned using the proposed method generate trajectories that accurately reproduce the demonstrations and are guaranteed to converge to a desired goal location. Additionally, the learned models are capable of quickly and appropriately adapting to unexpected spatial perturbations and changes in goal location during reproductions. The CDSP algorithm is evaluated on shapes from a publicly available human handwriting dataset and also compared with two state-of-the-art motion generation algorithms. Furthermore, the CDSP algorithm is also shown to be capable of learning and reproducing point-to-point motions directly from real-world demonstrations using a Baxter robot.

Keywords: Dynamical Systems, Learning from Demonstrations, Contraction Analysis, Gaussian Mixture Mode

1 Introduction

Learning from demonstration (LfD) [1–5] is a paradigm for training robots using demonstrations of a task. To this end, robots should be given the ability to learn motion plans from demonstrations shown by the user, e.g., in a human robot collaboration context, a robot must be able to both understand and learn from a non-expert user through the demonstrations of a task [6–9]. While there are several ways to represent the motions, we focus on the dynamical systems-based approach in which it is assumed that the motion of interest is driven by an underlying dynamical system. This paper presents an algorithm to learn the dynamics of complex point-to-point motions from few demonstrations.

One approach to learn the motions is training a statistical model such as a Gaussian Mixture Model (GMM) and a neural network (NN) using the available data. However, a simplistic approach of training these models does not provide any stability or convergence guarantees on the learned model. Hence, the trajectories of the learned model, especially the ones starting from regions of the state space without demonstrations, may not converge to the goal location of interest. Indeed, convergence to the correct goal location is crucial in many tasks, such as reaching, object manipulation, pick and place, pouring, and insertion. In this paper, partial contraction analysis (cf. [10]) is used to

develop a learning algorithm that both accurately reproduces the demonstrations and guarantees the convergence of all the trajectories of the learned system to the desired goal location. The primitives learned using our algorithm are referred to as contracting dynamical system primitives (CDSP). The dynamical system is represented using a GMM and its parameters are learned from demonstrations under the derived constraints.

While Lyapunov-based analysis studies the stability of a dynamical system with respect to a particular equilibrium point, contraction analysis is used to study incremental stability and the convergence of trajectories with respect to each other [11, 12]. Partial contraction analysis is a generalization of contraction analysis that studies the convergence to specific properties (e.g. an equilibrium point) [10]. Thus, a partially contracting dynamical system learned from demonstrations will generate trajectories that (a) accurately reproduce the demonstrations and (b) are guaranteed to converge to a desired goal location. Additionally, since partial contraction analysis leads to exponential stability, it is expected that embedding the partial contraction property in the learning framework would lead to a dynamical system which is robust under perturbations and can quickly adapt to changes in goal location.

Lyapunov-based LfD approaches that use various parametrization of Lyapunov function such as quadratic Lyapunov function ($V(x) = x^T x$) [3], weighted sum of asymmetric quadratic functions (WSAQF) [13], and neurally-imprinted Lyapunov candidate (NILC) [14] are studied in the literature. Alternatively, the contraction analysis-based LfD approach is recently developed in [15]. In the contraction analysis, the notion of distance between neighboring trajectories is defined as a virtual displacement δx . In [15], the dynamical system model is approximated by a NN model and is learned such that the squared length $\delta x^T \delta x$ between neighboring trajectories is reduced. Thus, this approach cannot learn complex motions (for e.g. *DoubleBendedLine* and *Leaf.1* from the publicly available LASA handwriting dataset [3]). In contrast, our work uses a generalized notion of virtual displacement $\delta z = \Theta(x) \delta x$ which leads to a generalized definition of squared length $\delta z^T \delta z = \delta x^T M(x) \delta x$, where $M(x) = \Theta(x)^T \Theta(x)$ is a contraction metric [11]. The generalized metric allows for learning of complex shapes, such as *DoubleBendedLine* and *Leaf.1* from LASA handwriting dataset. The metric $M(x)$, which is a positive definite symmetric matrix, is parametrized using polynomial functions in x of finite degree and its parameters are learned directly from the demonstrations. However, using a generalized metric leads to state-dependent constraints in the optimization problem for parameter learning, which must be satisfied at all the points of the state space of interest. Thus, making the optimization problem intractable. To circumvent this issue, a novel relaxation approach, inspired by sum of squares (SOS) decomposition, is proposed to eliminate the state dependence from the constraints (see Section 2). To illustrate the benefits of our approach, detailed experimental evaluations on the LASA handwriting dataset and comparisons with the two state-of-the-art learning algorithms are provided in Section 3. Furthermore, the CDSP algorithm’s ability to learn and reproduce point-to-point motions directly from real-world demonstrations is illustrated on a Baxter robot.

Related Work

LfD is a widely used approach for imitation learning, in which a new task or skill is learned from demonstrations provided by humans (see [16] for a comprehensive review). One way to accomplish this is to directly learn the policy or the dynamics involved in the task of interest [2, 3, 17–21]. In [21], task models are represented using Hidden Markov Models (HMMs) and are combined with a motion planning approach based on probabilistic road maps (PRMs). In [20], a keyframe-based LfD approach is proposed that learns tasks by aligning and clustering the keyframes of obtained demonstrations. In [17], a geometric approach is proposed that adapts available demonstrations to new start and goal locations via optimization. Inverse optimal control (IOC) or inverse reinforcement learning (IRL) is another framework used for imitation learning [22–26]. In IOC/IRL approaches, an objective function underlying a task is learned from the data and is optimized to reproduce trajectories of robot motion. In [27], a comparison between trajectory-based methods and optimal control methods is provided. Reinforcement learning-based approaches typically require the system to explore the state space for learning the policy which can be difficult for human-robot interaction applications. LfD-based approaches are more practical in such settings. The approach proposed in this paper, falls under the category of dynamical system-based LfD.

Dynamical system-based LfD methods have received a lot of attention in the recent past. One of the first dynamical system-based frameworks, called the Dynamic Movement Primitives (DMPs) [1, 2, 28], uses a stable dynamical system containing linear proportional derivative (PD)-like term coupled with a nonlinear term for encoding a desired trajectory. These two terms are coupled through a so-called phase variable. Tasks such as drumming [29], pouring [30], and pick and place have been recreated using DMPs on various robotic platforms. The DMP formulation, however, models multi-dimensional systems by learning one dynamical system for each dimension separately, thereby neglecting the combined effect of all the dimensions [3]. The idea of movement primitives is extended to probabilistic framework in [31]. In [18], task-parametrized GMMs (TpGMM) are used to design control policies for motion generation through generalization of available demonstration. However, the algorithm in [18] does not provide any stability guarantees for the learned dynamical systems. An algorithm to encode motion dynamics using Neurally Imprinted Vector Fields (NiVF) is presented in [19]. Stability of the learned system is verified using constraints derived through Lyapunov analysis. The NiVF algorithm restricts the NN learning process to an approach called extreme learning machines (ELM). While NiVF is shown to be capable of learning a variety of motions, the stability property is restricted to finite regions of the state space.

An LfD method called Stable Estimator of Dynamical Systems (SEDS) that learns globally stable dynamical systems directly in higher dimensional state space using Gaussian mixture models (GMMs) is developed in [3]. To keep the attractor properties of synthesized dynamical system, Lyapunov stability conditions are used to learn the parameters of GMMs, ensuring asymptotic stability of the goal location. An important limitation of SEDS is that a quadratic Lyapunov function ($V = x^T x$) is used in the development of stability constraints. Thus, SEDS can only model trajectories whose 2-norm distances to the target decrease monotonically in time. In [13], an approach called Control Lyapunov Function-based Dynamic Movements (CLF-DM) is introduced. This approach parametrizes the energy function using a weighted sum of asymmetric quadratic functions (WSAQF) and learns the parameters of the WSAQF from demonstrations. The learned energy function is later used during run time to ensure global stability of the generated trajectories. While it can encode a wide variety of motions, CLF-DM suffers from the disadvantage of the online correction signal potentially interfering with the learned dynamical system. More specifically, as pointed out in [32], CLF-DM has the disadvantage of solving a separate optimization problem for parameter selection of the control Lyapunov function, which can lead to numerical stability issues in parameter selection. In [32], an algorithm called τ -SEDS is introduced that uses diffeomorphic transformations along with the SEDS algorithm to generalize the class of motions that can be learned. The diffeomorphic transformation is used to transform the task space such that the transformed demonstrations are consistent with a quadratic Lyapunov function and hence SEDS can be used to learn the dynamics in the transformed space. The learned system is then back-transformed to the original task space. Our method, on the other hand, is capable of encoding motions that are partially contracting with respect to any state-dependent positive definite metric $M(x)$, which aids in the accurate learning of complex shapes such as *BendedLine*, *DoubleBendedLine*, *Leaf_1*, and *Leaf_2* from the LASA dataset [3]. Comparisons on four complex shapes from the LASA handwriting dataset indicate that the CDSP algorithm, on average, results in higher reproduction accuracy than the CLF-DM [13] and DMP [2] algorithms (see Section 3). Furthermore, quantitative comparisons indicate that the models learned using the CDSP algorithm, on average, generate motions that adapt better to sudden changes of goal location and spatial perturbations.

2 Learning Dynamics from Demonstrations

Consider a state variable $x(t) \in \mathbb{R}^n$ at time t and let a set of N demonstrations of a point-to-point motion be solutions to the following autonomous dynamical system

$$\dot{x}(t) = f(x(t)) \quad (1)$$

where $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a nonlinear continuously differentiable function. Each demonstration corresponds to a point-to-point motion ending at the goal location $x^* \in \mathbb{R}^n$. The n th demonstration consists of the trajectories of the states $\{x_n(t)\}_{t=0}^{t=T_n}$ and the trajectories of the state derivatives $\{\dot{x}_n(t)\}_{t=0}^{t=T_n}$.

Remark 1 *In the case of point-to-point motions, the position trajectories start from various initial locations and end at the final goal location. Additionally, the velocity and acceleration are zero at the goal location.*

The nonlinear function $f(\cdot)$ defined in (1) is modeled using a GMM and the resulting autonomous dynamical system [3, 33] is given by

$$\dot{x}(t) = \sum_{k=1}^K h_k(x(t)) (A_k x(t) + b_k) = f(x(t), x(t)) \quad (2)$$

where $h_k(x) = \frac{p^{(k)}p(x|k)}{\sum_{i=1}^K p^{(i)}p(x|i)}$ is the scalar weight associated with the k th Gaussian such that $0 \leq h_k(x) \leq 1$ and $\sum_{k=1}^K h_k(x) = 1$, $p(k) = \pi_k$ is the prior probability, $A_k(x) = \Sigma_{k,\dot{x}} (\Sigma_{k,x})^{-1}$, $b_k = \mu_{k,\dot{x}} - A_k \mu_{k,x}$, $\mu_k = [\mu_{k,x}, \mu_{k,\dot{x}}]^T$ and $\Sigma_k = \begin{pmatrix} \Sigma_{k,x} & \Sigma_{k,x,\dot{x}} \\ \Sigma_{k,x,\dot{x}} & \Sigma_{k,\dot{x}} \end{pmatrix}$ are the mean and the covariance of the k th Gaussian, respectively. Note that in $f(\cdot)$, $x(t)$ is written twice to represent the dependency of $x(t)$ at multiple places in $f(\cdot)$. Given a set of N demonstrations, this paper addresses the problem of learning the function $f(\cdot)$, which is modeled using a GMM, under constraints derived through partial contraction analysis. Now, consider an auxiliary system, with the state variable $y \in \mathbb{R}^n$,

$$\dot{y}(t) = f(y(t), x(t)) = \sum_{k=1}^K h_k(x(t)) (A_k y(t) + b_k) \quad (3)$$

Theorem 1 *If the constraints in (4) and (5) are satisfied for the auxiliary system in (3),*

$$(A_k)^T M(y) + \dot{M}_k(y) + M(y) A_k \preceq -\gamma M(y), \quad k = 1, \dots, K, \quad \forall y \quad (4)$$

$$A_k x^* + b_k = 0, \quad k = 1, 2, \dots, K \quad (5)$$

where γ is a strictly positive scalar constant, $M(y) \in \mathbb{R}^{n \times n}$ represents a positive definite symmetric matrix, and the ij th element of the matrix $\dot{M}_k(y)$ is given by $\dot{M}_{k,ij}(y) \triangleq \frac{dM_{ij}(y)}{dy} (A_k y + b_k)$, then the system in (3) is said to be contracting in y , and (2) is said to be partially contracting and all the trajectories of (2) will converge to the goal location x^* .

Proof: If the constraints in (5) are satisfied, then $y = x^*$ is the equilibrium point and a particular solution of the auxiliary system in (3). Based on contraction analysis [11], for the auxiliary system in (3) to be contracting, the following constraint has to be satisfied

$$\delta y^T \left(\frac{\partial f^T}{\partial y} M(y) + \dot{M}(y) + M(y) \frac{\partial f}{\partial y} \right) \delta y \leq -\gamma \delta y^T M(y) \delta y \quad (6)$$

where $M(y) \in \mathbb{R}^{n \times n}$ is a uniformly positive definite symmetric matrix, the ij th element of the matrix $\dot{M}(y)$ is given by $\dot{M}_{ij}(y) \triangleq \frac{dM_{ij}(y)}{dy} f(y) = \frac{dM_{ij}(y)}{dy} \left(\sum_{k=1}^K \{h_k(x) (A_k y + b_k)\} \right)$. On defining the matrix $\dot{M}_k(y)$ whose ij th element is given by $\dot{M}_{k,ij}(y) \triangleq \frac{dM_{ij}(y)}{dy} (A_k y + b_k)$, the matrix $\dot{M}(y)$ can be decomposed as $\dot{M}(y) = \sum_{k=1}^K \{h_k(x) \dot{M}_k(y)\}$. The Jacobian $\frac{\partial f}{\partial y}$ of the auxiliary system is given by $\frac{\partial f}{\partial y} \triangleq \sum_{k=1}^K \{h_k(x) A_k\}$. Thus, the expression on left hand side of (6) can be rewritten as $\delta y^T \left(\frac{\partial f^T}{\partial y} M(y) + \dot{M}(y) + M(y) \frac{\partial f}{\partial y} \right) \delta y = \delta y^T \left(\sum_{k=1}^K \{h_k(x) A_k^T\} M(y) + \sum_{k=1}^K \{h_k(x) \dot{M}_k(y)\} + M(y) \sum_{k=1}^K \{h_k(x) A_k\} \right) \delta y$. Further, on pulling $h_k(x)$ out of the sums and using (4), the following bound can be derived

$$\delta y^T \left(\frac{\partial f^T}{\partial y} M(y) + \dot{M}(y) + M(y) \frac{\partial f}{\partial y} \right) \delta y \leq -\delta y^T \left(\sum_{k=1}^K \{h_k(x) [\gamma M(y)]\} \right) \delta y \quad (7)$$

Since $0 \leq h_k(x) \leq 1$, $\forall k = 1, 2, \dots, K$ and $\sum_{k=1}^K h_k(x) = 1$, the inequality in (7) can be rewritten as $\delta y^T \left(\frac{\partial f^T}{\partial y} M(y) + \dot{M}(y) + M(y) \frac{\partial f}{\partial y} \right) \delta y \leq -\gamma \delta y^T M(y) \delta y$ and consequently, the auxiliary system in (3) is said to be contracting with respect to y [11].

According to partial contraction theory [10], if the auxiliary y -system in (3) is contracting and the trajectories $y = x^*$ (equilibrium point), and $y(t) = x(t)$, $\forall t \geq 0$ are particular solutions of (3), then the trajectories $x(t)$ of (2) will globally exponentially converge to the goal location x^* . ■

Remark 2 During implementation, the constraints in (4) are evaluated at x since Theorem 1 shows that the trajectories $y(t)$ and $x(t)$ converge to each other exponentially.

The constrained optimization problem to be solved in order to train the GMM model with the demonstrations can be written as

$$\{\theta_G^*, \theta_M^*\} = \arg \min_{\theta_G, \theta_M} \frac{1}{2\mathcal{T}} \sum_{n=1}^N \sum_{t=0}^{T_n} \|\hat{x}_n(t) - \dot{x}_n(t)\|^2 \quad (8)$$

$$\text{s.t. } (A_k)^T M(x) + \dot{M}_k(x) + M(x) A_k \preceq -\gamma M(x), \quad k = 1, \dots, K, \quad \forall x \quad (9)$$

$$A_k x^* + b_k = 0, \quad k = 1, \dots, K, \quad (10)$$

$$M(x, \theta_M) \succ 0, \quad \forall x \quad (11)$$

$$\Sigma_k \succ 0, \quad \sum_k \pi_k = 1, \quad 0 \leq \pi_k \leq 1 \quad k = 1, \dots, K, \quad (12)$$

where $\theta_G = \{\mu_1 \dots \mu_K, \Sigma_k \dots \Sigma_K, \pi_1 \dots \pi_K\}$ is a vector containing the parameters of the GMM model, $\theta_M \in \mathbb{R}^m$ is the set of all the unknown constant coefficients in the matrix $M(x)$, $\mathcal{T} = \sum_{n=1}^N T_n$ is the total number of data points in the demonstrations, and $\hat{x}_n(t) = \hat{f}(x_n(t))$ is the predicted state derivative computed based on (2). Note that the parameters of contraction metric are also learned from the data as opposed to being manually designed. The constraints (9)-(11) ensure the global attraction of the goal location x^* and the constraints in (12) are a result of using a GMM.

Note that the constraints in (9) and (11) must be satisfied for all points in the state space, which makes the optimization problem intractable. To circumvent this issue, a two-fold approach is proposed. First, the elements of the contraction metric $M(x)$ are assumed to be polynomial functions in x .

Assumption 1 The contraction metric is defined as $M(x) \triangleq \Theta(x)^T \Theta(x)$, where $\Theta(x) \in \mathbb{R}^{n \times n}$ and the ij th element of $\Theta(x)$ is given by $\Theta_{ij}(x) = r_{ij}(x)$ where $r_{ij}(x)$ is a polynomial function in x up to a maximum order d_{max} .

Second, the constraint in (9) is reformulated such that the state dependency is eliminated, resulting in a constraint that is a function of the parameters to be learned. On defining the matrices

$$G_k(\theta_G, \theta_M, x) \triangleq (A_k)^T M(x) + \dot{M}_k(x) + M(x) A_k + \gamma M(x), \quad k = 1, \dots, K, \quad (13)$$

the constraints in (9) can be rewritten as the scalar constraints $y^T G_k(\theta_G, \theta_M, x) y \leq 0, \forall x, y, k = 1, \dots, K$, where $y \in \mathbb{R}^n$ is a vector of indeterminates [34]. Based on Assumption 1, the expression $y^T G_k(\theta_G, \theta_M, x) y$ can be further decomposed as

$$y^T G_k(\theta_G, \theta_M, x) y = m(x, y)^T \bar{G}_k(\theta_G, \theta_M) m(x, y) \quad (14)$$

where $m(x, y) \in \mathbb{R}^{\bar{n}}$ is a vector of monomials in the elements of x and y [34] and the elements of the matrix $\bar{G}_k(\theta_G, \theta_M)$ can be obtained by coefficient matching. Similarly, the constraint $M(x, \theta_M) \succ 0, \forall x$ can be rewritten as the scalar constraint, $y^T M(x, \theta_M) y > 0, \forall x, y$. Since the elements of $M(x, \theta_M)$ are polynomial functions in x , similar to the expression $y^T G_k(\theta_G, \theta_M, x) y$, the expression $y^T M(x) y$ can be decomposed as

$$y^T M(x, \theta_M) y = q(x, y)^T \bar{M}(\theta_M) q(x, y) \quad (15)$$

where $q(x, y) \in \mathbb{R}^{\bar{m}}$ is a vector of monomials in the elements of x and y and the elements of the matrix $\bar{M}(\theta_M)$ are polynomials in the elements of θ_M .

Theorem 2 The autonomous dynamical system in (2) is said to be partially contracting and all its solutions will converge to the goal location x^* , if $\bar{M}(\theta_M) \succ 0$, and for $k = 1, 2, \dots, K$, $\bar{G}_k(\theta_G, \theta_M) \preceq 0$, and $A_k x^* + b_k = 0$.

Proof: If $\bar{G}_k(\theta_G, \theta_M) \preceq 0, k = 1, 2, \dots, K$, it directly follows that

$$m(x, y)^T \bar{G}_k(\theta_G, \theta_M) m(x, y) \leq 0, \forall x, y, k = 1, \dots, K \quad (16)$$

where $m(x, y)$ is a vector of monomials in the elements of x and y . Further, based on (14), the inequality in (16) implies that $y^T G_k(\theta_G, \theta_M, x) y \leq 0, \forall x, y, k = 1, 2, \dots, K$ and consequently

$$G(\theta_G, \theta_M, x) \preceq 0, \forall x, k = 1, 2, \dots, K \quad (17)$$

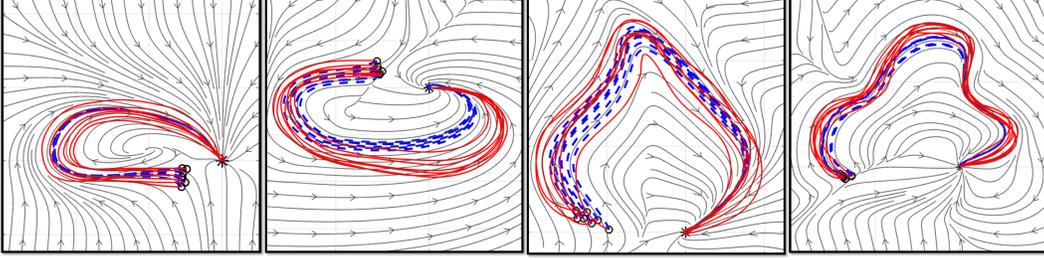


Figure 1: Models of four shapes (*BendedLine*, *DoubleBendedLine*, *Leaf_1*, and *Leaf_2*), from the LASA handwriting dataset, learned using the CDSP algorithm. The streamlines of the models are shown (in gray) along with the demonstrations (in solid red lines) and the reproductions (in broken blue lines) generated using the learned models.

Similarly, based on (15), if $\bar{M}(\theta_M) \succ 0$, then $M(x) \succ 0, \forall x$. Thus, based on (13) and (17), along with the fact that $A_k x^* + b_k = 0, k = 1, 2, \dots, K$, it can be seen that the constraints in (4) and (5) are satisfied. Hence, according to Theorem 1, the system in (2) is partially contracting. ■

The modified constrained optimization problem is given by

$$\{\theta_G^*, \theta_M^*\} = \arg \min_{\theta_G, \theta_M} \frac{1}{2T} \sum_{n=1}^N \sum_{t=0}^{T_n} \|\hat{x}_n(t) - \dot{x}_n(t)\|^2 \quad (18)$$

$$\text{s.t. } \bar{G}_k(\theta_G, \theta_M) \leq 0, k = 1, \dots, K, \bar{M}(\theta_M) \succ 0, \quad (19)$$

$$A_k x^* + b_k = 0, \Sigma_k \succ 0, \sum_k \pi_k = 1, 0 \leq \pi_k \leq 1 k = 1, \dots, K, \quad (20)$$

3 Experimental Validation

Performance on the LASA Handwriting Dataset: The experiments are conducted using a desktop computer running Intel i3 processor and 8 Gigabytes of memory. The CDSP algorithm is coded using MATLAB 2016a and the *fmincon* function is used to solve the optimization problem. Initial estimates of the parameters of the GMM are obtained using the Expectation Maximization (E-M) algorithm. The CDSP algorithm is tested on four shapes (*BendedLine*, *DoubleBendedLine*, *Leaf_1*, and *Leaf_2*) from the LASA human handwriting library, introduced in [3], that consists of handwriting motions collected from pen input using a Tablet PC. These four shapes represent complex dynamics where the 2-norm distances to the target do not decrease monotonically. Hence, the use of a state-dependent contraction metric is crucial for accurately encoding the dynamics. The qualitative performance of the CDSP algorithm on these shapes is shown in Fig. 1. The CDSP algorithm is then compared with the DMP [2] and CLF-DM [13] algorithms. The source codes of the CLF-DM and DMP algorithms were obtained from the respective authors' websites. The number of basis functions for the DMP and CLF-DM algorithms for each shape were empirically chosen as follows: (i) *BendedLine*: DMP: 25, CLF-DM: 6; (ii) *DoubleBendedLine*: DMP: 35, CLF-DM: 7; (iii) *Leaf_1*: DMP: 25, CLF-DM: 6; (iv) *Leaf_2*: DMP: 35, CLF-DM: 6. Other design parameters of the DMP algorithm are chosen according to the guidelines provided in [2]. To carry out a fair comparison, for each shape, the number of Gaussians of the CDSP algorithm was chosen to be equal to that of the CLF-DM algorithm. Additionally, the number of asymmetric functions of the CLF-DM algorithm was chosen based on the guidelines in [13]. Quantitative comparisons are carried out in terms of the reproduction accuracy as measured by swept error area (SEA) [13] and dynamic time warping distance (DTWD) between the demonstrations and the reproductions. The average SEA for a method is given by $SEA = \frac{1}{N_d} \sum_{i=1}^{N_d} \sum_{t=0}^{T_i-1} \mathcal{A}(\hat{x}_i(t), \hat{x}_i(t+1), x_i(t), x_i(t), x_i(t+1))$ where $\hat{x}_i(t), \forall t = 0, \dots, T_i$ is the equidistantly re-sampled reproduction of the i th demonstration with T_i samples and $\mathcal{A}(\cdot)$ denotes the area of enclosed tetrahedron with the points $\hat{x}_i(t), \hat{x}_i(t+1), x_i(t)$, and $x_i(t+1)$ as corners. Note that DTWD measures the dissimilarity between the shapes of the demonstrations and the corresponding reproductions without taking time alignment into account. On the other hand, SEA penalizes both spatial and temporal misalignment between the demonstrations and the corresponding reproductions. The results of the comparisons are illustrated in Fig. 2.

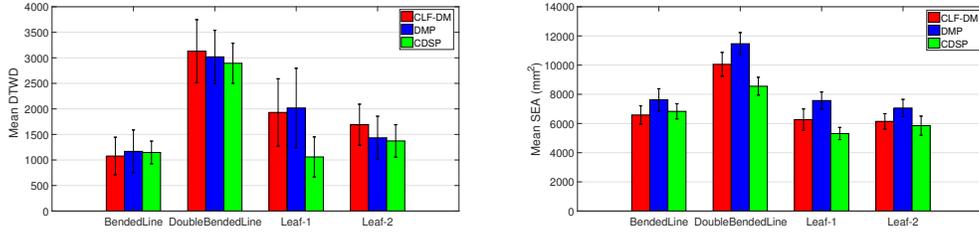


Figure 2: The average dynamic time warping distance (DTWD) and swept error area (SEA) between seven demonstrations and the corresponding reproductions of each algorithm computed for each shape is shown along with one standard deviation.

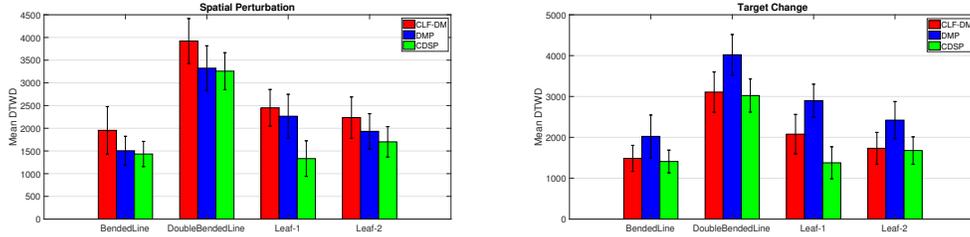


Figure 3: The average dynamic time warping distance (DTWD) between seven demonstrations and the corresponding reproductions of each algorithm after spatial perturbations and target changes. The averages are shown for each shape along with one standard deviation.

To compare the robustness of the algorithms, learned using the three algorithms, to sudden spatial perturbations, a sudden push to the current position is applied during reproductions of the shapes *BendedLine* and *Leaf-1*. These perturbations simulate disturbances to the robot’s end-effector during reproductions. Similarly, a test was conducted to illustrate and compare the algorithms’ abilities to adapt to sudden changes in goal location. The times, the amplitudes, and the directions of the pushes and the changes of goal location were chosen to be the same for all the algorithms and were determined according to the guidelines presented in [35]. It was seen that the reproductions of the CDSP algorithm consistently converge to the correct goal location after sudden perturbations and target changes. Exemplar reproductions of the models learned using different algorithms are shown in Fig. 6. Furthermore, due to the scaling of the acceleration profiles after the perturbations, some of the reproductions of the DMP algorithm either overshoot the target before convergence, or seem to have excessive curvature. Indeed, as noted in [2], a more careful choice of coordinate system would help circumvent the issue of excess curvature during generalization. However, such a choice has to be hand-crafted for each model. The robustness of the learned models are quantified by evaluating the DTWD between the reproductions after perturbations or target changes and the respective demonstrations. Since, for evaluating robustness, temporal alignment of the reproductions and the trajectories is not as important as shape reconstruction, SEA is not evaluated in this experiment. The results of the comparisons are illustrated in Fig. 2. Based on the observed statistics, the reproductions of the CDSP algorithm, on average, are more robust (in terms of DTWD) to sudden perturbations and target changes when compared to the CLF-DM and DMP algorithms.

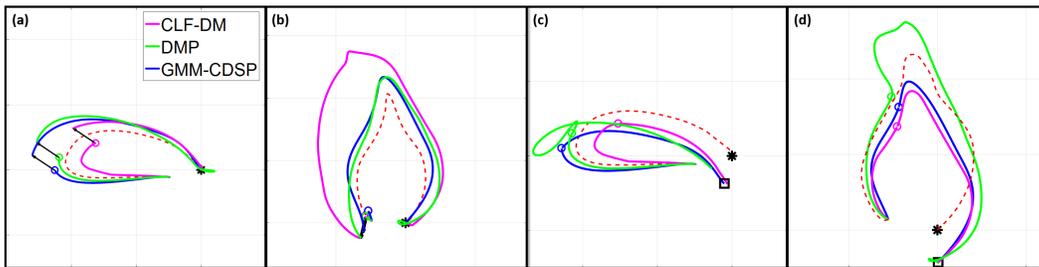


Figure 4: Exemplar trajectories illustrating the robustness of the learned models to sudden spatial perturbations (plots a and b) and target changes (plots c and d) during reproductions of the shapes *BendedLine* and *Leaf-1*.

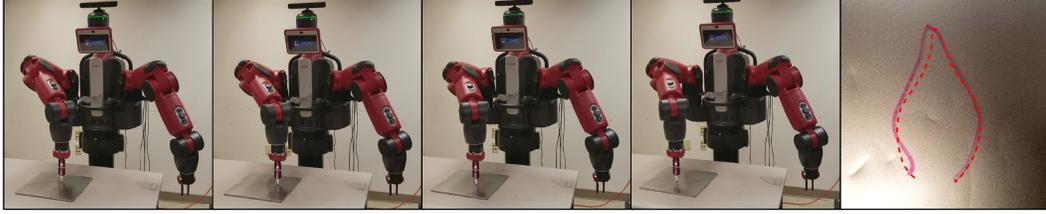


Figure 5: Sequence of images showing the Baxter robot autonomously drawing the *Leaf_1* shape learned from the LASA handwriting dataset.

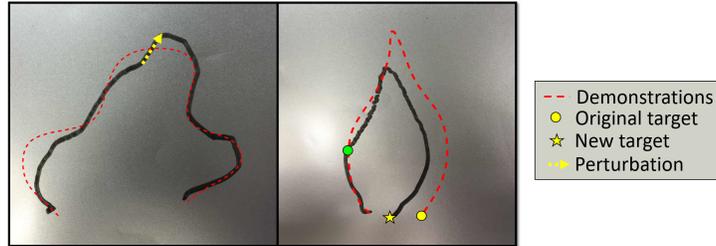


Figure 6: Baxter robot adapting its motion to sudden spatial perturbations (left) and target changes (right) during reproductions.

Robot Implementation: In order to demonstrate the utility of our approach in robot motion generation, the GMM, trained on the demonstrations of the shape “*Leaf_1*”, is used to generate reference trajectories for a seven degree-of-freedom Baxter research robot to draw the leaf shape autonomously. To execute the trajectories on the Baxter robot, the in-built low-level controller and the inverse kinematics engine IKFast are used. A sequence of images showing the Baxter robot drawing the *Leaf_1* shape is shown in Fig. 5. Furthermore, as shown in Fig. 6, the learned model’s ability to adapt the robot’s motion to sudden spatial perturbations was validated. The perturbations were chosen according to the guidelines in [35] and were applied using the control software. Finally, to showcase the CDSP algorithm’s ability to directly learn motions from real-world data, a set of four demonstrations for a new shape were collected by providing kinesthetic demonstrations to the Baxter robot. As shown in Fig. 7, the CDSP algorithm was able to successfully learn the motion dynamics of the new shape while guaranteeing global convergence to the goal location.

4 Conclusion

The CDSP algorithm for learning arbitrary point-to-point motions using dynamical systems is presented. The experimental evaluations on four shapes of the LASA human handwriting library suggest that the CDSP algorithm is able to successfully learn stable complex point-to-point motions that are robust to spatial perturbations and can adapt to sudden changes in goal location. Comparisons against the DMP and CLF-DM algorithms show that the CDSP algorithm resulted in the highest reproduction accuracy for three of the four shapes, measured by mean swept error area (SEA) and dynamic time warping distance (DTWD). Furthermore, the CDSP algorithm’s reproductions are shown to be robust to sudden spatial perturbations and target changes, measured in terms of DTWD. Finally, the CDSP algorithm’s ability to learn motions directly from real-world demonstrations given to a robot is validated.

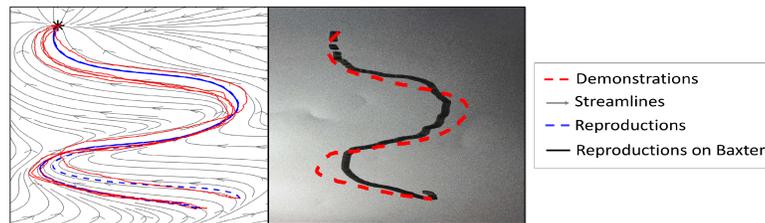


Figure 7: Learning from noisy kinesthetic demonstrations given to the Baxter robot.

Acknowledgments

The authors would like to acknowledge the support from the UTC Institute for Advanced Systems Engineering (UTC-IASE) of the University of Connecticut and the United Technologies Corporation. Any opinions expressed herein are those of the authors and do not represent those of the sponsor.

References

- [1] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242, 1999.
- [2] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural Computation*, 25(2):328–373, 2013.
- [3] S. M. Khansari-Zadeh and A. Billard. Learning stable nonlinear dynamical systems with Gaussian mixture models. *IEEE Transactions on Robotics*, 27(5):943–957, 2011.
- [4] J. Rey, K. Kronander, F. Farshidian, J. Buchli, and A. Billard. Learning motions from demonstrations and rewards with time-invariant dynamical systems based policies. *Autonomous Robots*, 2017. doi:10.1007/s10514-017-9636-y.
- [5] C. Wang, Y. Zhao, C.-Y. Lin, and M. Tomizuka. Fast planning of well conditioned trajectories for model learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1460–1465, 2014.
- [6] G. F. Rossano, C. Martinez, M. Hedelind, S. Murphy, and T. A. Fuhlbrigge. Easy robot programming concepts: An industrial perspective. In *IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1119–1126, 2013.
- [7] H. Ravichandar, P. k. Thota, and A. P. Dani. Learning periodic motions from human demonstrations using transverse contraction analysis. In *American Control Conference (ACC)*, pages 4853–4858, 2016.
- [8] H. Ravichandar and A. P. Dani. Human intention inference using expectation-maximization algorithm with online model learning. *IEEE Transactions on Automation Science and Engineering*, 14(2):855–868, 2017.
- [9] N. Koenig and M. J. Matarić. Robot life-long task learning from human demonstrations: a bayesian approach. *Autonomous Robots*, pages 1–16, 2016. ISSN 1573-7527.
- [10] W. Wang and J.-J. E. Slotine. On partial contraction analysis for coupled nonlinear oscillators. *Biological Cybernetics*, 92(1), 2005.
- [11] W. Lohmiller and J.-J. E. Slotine. On contraction analysis for nonlinear systems. *Automatica*, 34(6):683–696, 1998.
- [12] A. P. Dani, S.-J. Chung, and S. Hutchinson. Observer design for stochastic nonlinear systems via contraction-based incremental stability. *IEEE Transactions on Automatic Control*, 60(3): 700–714, 2015.
- [13] S. M. Khansari-Zadeh and A. Billard. Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions. *Robotics and Autonomous Systems*, 62(6): 752–765, 2014.
- [14] K. Neumann, A. Lemme, and J. J. Steil. Neural learning of stable dynamical systems based on data-driven lyapunov candidates. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1216–1222, 2013.
- [15] H. Ravichandar and A. P. Dani. Learning contracting nonlinear dynamics from human demonstration for robot motion planning. In *ASME Dynamic Systems and Control Conference (DSCC)*, 2015.

- [16] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.
- [17] A. D. Dragan, K. Muelling, J. A. Bagnell, and S. S. Srinivasa. Movement primitives via optimization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2339–2346, 2015.
- [18] S. Calinon, D. Bruno, and D. G. Caldwell. A task-parameterized probabilistic model with minimal intervention control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3339–3344, 2014.
- [19] A. Lemme, K. Neumann, F. Reinhart, and J. J. Steil. Neurally imprinted stable vector fields. In *European Symposium on Artificial Neural Networks*, pages 327–332, 2013.
- [20] B. Akgun, M. Cakmak, K. Jiang, and A. L. Thomaz. Keyframe-based learning from demonstration. *International Journal of Social Robotics*, 4(4):343–355, 2012.
- [21] C. Bowen and R. Alterovitz. Closed-loop global motion planning for reactive execution of learned tasks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1754–1760, 2014.
- [22] E. Todorov and M. I. Jordan. Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, 5(11):1226–1235, 2002.
- [23] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *ACM International Conference on Machine Learning*, 2004.
- [24] J.-P. Laumond, N. Mansard, and J.-B. Lasserre. Optimality in robot motion: optimal versus optimized motion. *Communications of the ACM*, 57(9):82–89, 2014.
- [25] M. C. Priess, J. Choi, and C. Radcliffe. The inverse problem of continuous-time linear quadratic Gaussian control with application to biological systems analysis. In *ASME Dynamic Systems and Control Conference*, 2014.
- [26] J.-P. Laumond, N. Mansard, and J. B. Lasserre. Optimization as motion selection principle in robot action. *Communications of the ACM*, 58(5):64–74, 2015.
- [27] S. Schaal, P. Mohajerin, and A. Ijspeert. Dynamics systems vs. optimal control—a unifying view. *Progress in Brain Research*, 165:425–445, 2007.
- [28] A. Rai, F. Meier, A. Ijspeert, and S. Schaal. Learning coupling terms for obstacle avoidance. In *International Conference on Humanoid Robotics*, pages 512–518, 2014.
- [29] A. Ude, A. Gams, T. Asfour, and J. Morimoto. Task-specific generalization of discrete and periodic dynamic movement primitives. *IEEE Transactions on Robotics*, 26(5):800–815, 2010.
- [30] B. Nemeč, M. Tamošiūnaitė, F. Wörgötter, and A. Ude. Task adaptation through exploration and action sequencing. In *IEEE-RAS International Conference on Humanoid Robots*, pages 610–616, 2009.
- [31] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann. Probabilistic movement primitives. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2616–2624, 2013.
- [32] K. Neumann and J. J. Steil. Learning robot motions with stable dynamical systems under diffeomorphic transformations. *Robotics and Autonomous Systems*, 70:1–15, 2015.
- [33] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4(1):129–145, 1996.
- [34] D. Henrion and A. Garulli. *Positive polynomials in control*, volume 312. Springer Science & Business Media, 2005.
- [35] A. Lemme, Y. Meirovitch, S. M. Khansari-Zadeh, T. Flash, A. Billard, and J. J. Steil. Open-source benchmarking for learned reaching motion generation in robotics. *Paladyn Journal of Behavioral Robotics*, 6(1):30–41, 2015.