Learning Stable Dynamical Systems using Contraction Theory

Caroline Blocher[2], Matteo Saveriano[1] and Dongheui Lee[1]
[1] Human-centered Assistive Robotics, Technical University of Munich, Munich, Germany
(Tel: +49-89-289-26900; E-mail: {matteo.saveriano,dhlee}@tum.de)
[2] Department of Civil and Environmental Engineering, Imperial College, London, UK (E-mail: c.blocher16@imperial.ac.uk)

***Abstract*—This paper discusses the learning of robot point-to-point motions via non-linear dynamical systems and Gaussian Mixture Regression (GMR). The novelty of the proposed approach consists in guaranteeing the stability of a learned dynamical system via Contraction theory. A contraction analysis is performed to derive sufficient conditions for the global stability of a dynamical system represented by GMR. The results of this analysis are exploited to automatically compute a control input which stabilizes the learned system on-line. Simple and effective solutions are proposed to generate motion trajectories close to the demonstrated ones, without affecting the stability of the overall system. The proposed approach is evaluated on a public benchmark of point-to-point motions and compared with state-of-the-art algorithms based on Lyapunov stability theory.**

***Keywords*—Learning contracting systems. Stable discrete movements. Learning from demonstration. Contraction theory.**

1. Introduction

Programming by Demonstration (PbD) [1] is a useful tool to rapidly increase robot skills. Instead of explicitly programming a robot, the user teaches the robot how to perform a certain task by demonstrating the correct task execution [2], [3]. Learned skills are represented in a compact form which reduces memory requirements [4].

A recent trend in PbD research [5]–[11] suggests to represent robotic skills via stable dynamical systems (DS). Stable DS are well suited to represent point-to-point motions, since they are guaranteed to converge to a specified target. Moreover, dynamical systems are robust to changes in the initial/target location, and can be used in cluttered environments to generate collision-free paths [12]–[14]. Dynamic Movement Primitives (DMP) [5] are one of the first example of DS learned by demonstrations. DMP superimpose a linear DS and a non-linear force term learned from a single demonstration.

The generation of stable motions from non-linear DS, represented by GMR, is considered in [6], where asymptotic convergence to the target is guaranteed for trajectories that remain inside the demonstration area. The global convergence of a DS, represented by GMR, is guaranteed by the stable estimator of dynamical systems (SEDS) [7]. In SEDS, GMM parameters are learned by solving a constrained optimization problem, where stability constraints are derived from a quadratic Lyapunov function. The main advantage of SEDS is that the learned DS is proven to be globally stable [7]. For complex motions, however, contradictions may occur between demonstrations and quadratic stability constraints, which prevents SEDS to accurately learn the motion [9].

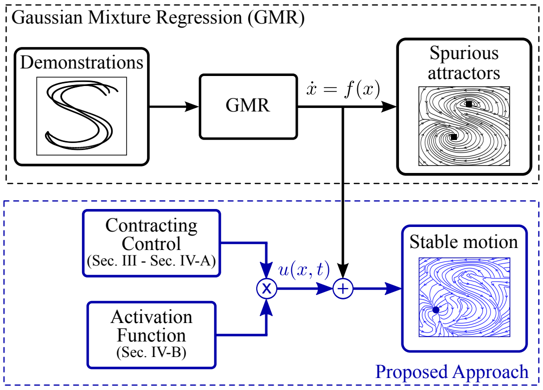


Fig. 1. System overview. A dynamical system represented by GMR generates spurious attractors (top). The dynamical systems reaches the desired target by applying an automatically computed control input (bottom).

The accurate motion reproduction is considered in [8] and in the SEDSII approach [9]. A common idea between [8] and [9] is to learn a Lyapunov function, which minimizes the contradictions between the stability constraints and the training data, from demonstrations. In [8], the DS is represented by an extreme learning machine (ELM) [15]. ELM parameters are learned by solving a constrained optimization problem, where stability constraints are derived from the learned Lyapunov function. In SEDSII the Lyapunov function is used to compute a stabilizing control input. The main advantage of SEDSII is its applicability to any regression technique, while [8] is limited to ELM. Alternative approaches [10], [11] propose to learn a diffeomorphic transformation that maps the trajectories of the DS into a space where demonstrations are accurately reproduced. The approach in [11] is significantly faster than [10], but it works only for linear DS.

Contraction theory [16] is effectively applied in [17] to stabilize a neural network DS. Authors perform the contraction analysis and use the resulting constraints to learn the neural network paramters. The parameter learning is formulated as a constrained optimization problem.

The contribution of this paper is to investigate the applicability of Contraction theory to stable motion generation via DS and GMR. To this end, we perform the contraction analysis of a DS represented by GMR and derive sufficient stability conditions. These conditions are leveraged to automatically

TABLE 1
DEFINITIONS

***Definition** 1:* GMR parameters
$m^k = \begin{bmatrix} m_x^k \\ m_{\dot{x}}^k \end{bmatrix}$, $\Sigma^k = \begin{bmatrix} \Sigma_x^k & \Sigma_{x\dot{x}}^k \\ \Sigma_{\dot{x}x}^k & \Sigma_{\dot{x}}^k \end{bmatrix}$, $A^k = \Sigma_{\dot{x}x}^k (\Sigma_x^k)^{-1}$,
$b^k = m_{\dot{x}}^k - A^k m_x^k$, $h^k(x) = \frac{\pi^k \mathcal{N}(x|m_x^k, \Sigma_x^k)}{\sum_{i=1}^K \pi^i \mathcal{N}(x|m_x^i, \Sigma_x^i)}$

***Definition** 2:* Let $||A||_i$ be an induced matrix norm of A on $\mathbb{R}^{m\times m}$.
The corresponding **matrix measure** is defined by
$\mu(A) = \lim_{\epsilon\rightarrow 0+} \frac{||I+\epsilon A||_i - 1}{\epsilon}$. Useful properties of $\mu(A)$ are:
1) $\mu(\alpha A) = \alpha\, \mu(A)$, $\forall A \in \mathbb{R}^{m\times m}, \alpha \geq 0$
2) $\mu(A + B) \leq \mu(A) + \mu(B)$, $\forall A, B \in \mathbb{R}^{m\times m}$

***Definition** 3:* Jacobian $F(x,t) = \partial f/\partial x$ is uniformly negative definite
if $\exists \beta > 0 | \forall x \in \mathcal{C} \subseteq \mathbb{R}^m, \forall t \geq 0, \frac{1}{2}\left(F(x,t)^{\mathrm{T}} + F(x,t)\right) \leq -\beta I < 0$

compute a stabilizing control input given the GMR parameters. The control action is smoothly activated or deactivated using an activation function (see Fig. 1). In this way, we improve the reproduction accuracy without affecting the stability of the controlled DS. Compared to [17], we perform the contraction analysis of a GMR system, and use the resulting constraints to compute, at run time, a stabilizing control input. Our approach shares with SEDSII the idea of stabilizing the DS at run time. In contrast to SEDSII, the proposed control law does not require the learning of a suitable Lyapunov function. The performance of the proposed approach, called *Contracting GMR* (C-GMR), are evaluated on a public dataset and compared with Lyapunov based approaches in [7], [9].

2. PROPOSED APPROACH

2.1. Problem Definition

In this section, we recap on DS represented by GMR from [7]. Let us consider N demonstrations $\mathcal{D} = \{x_{t,i}, \dot{x}_{t,i}\}_{t=1,i=1}^{T,N}$ of a point-to-point motion, where $x_{t,i}, \dot{x}_{t,i} \in \mathbb{R}^m$ are respectively the position and the velocity. We assume that the demonstrations are drawn from a non-linear and smooth DS $\dot{x} = f(x, \theta)$, where $\theta \in \mathbb{R}^d$ is a vector of parameters depending on the regression technique. GMR is used in this work to model $f(x, \theta)$. Therefore, the non-linear function $f(x, \theta)$ is parametrized by $\theta = \{\pi^k, m^k, \Sigma^k\}_{k=1}^K$, i.e. by the priors π^k, the means m^k and the covariance matrices Σ^k (see Tab. 1). $f(x, \theta)$ can be written as [7]:

$$\dot{x} = f(x, \theta) = \sum_{k=1}^{K} h^k(x)(A^k x + b^k) \tag{1}$$

where A^k, b^k and $h^k(x)$ are defined in Tab. 1. As shown in [7], the DS in (1) can generate spurious equilibria, which limit the applicability of GMR in point-to-point motion learning.

2.2. Contraction Analysis

Contraction theory [16] is a novel approach to analyze the stability of non-linear DS. Contraction theory is based on the idea that moving along a trajectory of the contracting system, the pointwise distance to its neighboring trajectories exponentially decreases. In particular, a DS is contracting if the generalized Jacobian $F(x, t)$ is uniformly negative definite (see *Definition 3* in Tab. 1). Notably, contraction of a DS can be analyzed by using the matrix measure $\mu(F)$ of the Jacobian.

In order to show that (1) is contracting, we have to compute the Jacobian matrix. This computation is not trivial, due to the term $\partial h^k(x)/\partial x$. Hence, we exploit Partial Contraction theory [18] to simplify the problem. We define an auxiliary DS:

$$\dot{y} = \sum_{k=1}^{K} h^k(x)(A^k y + b^k) \tag{2}$$

where y is the auxiliary state variable. Proving that the auxiliary system has an equilibrium point $x^\star$ and that it is globally contracting allows us to conclude that the original DS globally exponentially converges to $x^\star$ [18]. The Jacobian of the auxiliary system is simply $J_a = \sum_{k=1}^K h^k(x) A^k$.

Theorem 1. *The dynamical system in (1) globally exponentially converges to $x^\star$ if for all $k = 1, \ldots, K$:*

- $b^k = -A^k x^\star$ (3)
- $\exists \mu(\cdot)$ *such that* $\mu(A^k) < 0$ (4)

Proof. Condition (3) guarantees that the auxiliary system (2) has an equilibrium point at $x^\star$. Indeed, by substituting (3) into (2), we have $\sum_{k=1}^K h^k(x) A^k (y - x^\star) = 0$ if $y = x^\star$. Condition (4) guarantees that the auxiliary system is globally contracting. Indeed, considering (4), *Definition 2*, and that $0 \leq h^x \leq 1$ from *Definition 1*, we have $\mu(J_a) \leq \sum_{k=1}^K h^k(x)\mu(A^k) < 0$ for all $y \in \mathbb{R}^m$. As the auxiliary system is autonomous and globally contracting, it has a unique equilibrium at $x^\star$ [16]. Partial Contraction theory allows us to conclude that the DS (1) globally exponentially converges to $x^\star$. □

2.3. Global Exponential Stabilizer

Stability conditions in *Theorem 1* allow the usage of different matrix measures, Lyapunov theory requires the usage of the Euclidean norm. We exploit this advantage of Contraction theory to automatically compute a stabilizing controller.

The DS in (1) can be stabilized with the control input:

$$\dot{x} = \underbrace{\sum_{k=1}^{K} h^k(x)\left(A^k x + b^k\right)}_{System} + \underbrace{\sum_{k=1}^{K} h^k(x)\left(U^k x - A^k x^\star\right)}_{Control} \tag{5}$$

To prove that (5) globally exponentially converges to $x^\star$, we consider the auxiliary DS $\dot{y} = \sum_{k=1}^K h^k(x)(A^k y + b^k) + \sum_{k=1}^K h^k(x)(U^k y - A^k x^\star)$. The term $-\sum_{k=1}^K h^k(x) A^k x^\star$ is introduced to satisfy condition (3). The auxiliary DS can be re-written as $\dot{y} = \sum_{k=1}^K h^k(x)\left(A^k + U^k\right)(y - x^\star)$, which underlines that the control matrices U^k have to guarantee $\mu(A^k + U^k) < 0,\ k = 1, .., K$ (condition (4)).

To compute the control matrices U^k, we propose an automatic procedure based on the measure $\mu_1(\cdot)$ associated to the l_1-norm. The procedure is summarized in Algorithm 1. Given a square matrix $A \in \mathbb{R}^{m\times m}$, Algorithm 1 finds a diagonal matrix $U \in \mathbb{R}^{m\times m}$ such that $C = A + U$ has negative diagonal elements $c_{ii} < 0,\ i = 1, .., m$ and is diagonally dominant $|c_{ii}| > \sum_{j=1}^m |c_{ij}|,\ i = 1, .., m$. In other words, Algorithm

Algorithm 1 Find a matrix U such that $\mu_1(A+U) < 0$

```
Require: $A \in \mathbb{R}^{m \times m}$, $p > 1$, $m$        // dimension of state space
  $U \leftarrow$ Matrix of zeros in $\mathbb{R}^{m \times m}$
  for $d = 1$ to $m$ do
    $s = \sum_{i \neq d} |a_{di}|$
    if $a_{dd} > 0$ and $a_{dd} < s$ then
      $u_{dd} \leftarrow -\ s - p * a_{dd}$
    else if $a_{dd} > 0$ and $a_{dd} > s$ then
      $u_{dd} \leftarrow -\ 2 * a_{dd}$
    else if $a_{dd} < 0$ and $|a_{dd}| < s$ then
      $u_{dd} \leftarrow -\ s$
    else
      $u_{dd} \leftarrow 0$                    // no need to modify $a_{dd}$
    end if
  end for
  return $U$
```

1 guarantees that $\mu_1(C) \triangleq \max_j(c_{jj} + \sum_{i \neq j} |c_{ij}|) < 0$. In details, Algorithm 1 inspects A row by row. For each row, the algorithm computes the sum of the off-diagonal elements $s = \sum_{i \neq d}^{m} |a_{di}|$ and considers four cases: i) if the d-th diagonal element of A is positive ($a_{dd} > 0$) and $a_{dd} < c$ then $u_{dd} = -s - p * a_{dd}$. ii) If $a_{dd} > 0$ and $a_{dd} < s$ then $u_{dd} = -\ 2 * a_{dd}$. iii) If $a_{dd} < 0$ and $|a_{dd}| < s$ then $u_{dd} = -s$. iv) In all the other cases there is no need to modify A. It is easy to verify that $a_{dd} + u_{dd} < 0$ and that $|a_{dd} + u_{dd}| > c, d = 1, .., m$, i.e. $\mu_1(A+U) < 0$.

The controller in (5) guarantees the convergence towards a unique equilibrium. The control gains are automatically computed given the GMM parameters and the DS is stabilized at run time. Figure 2 shows qualitative results of the proposed approach when applied to stabilize a GMR model. The control input affects the reproduced point-to-point motions and do not guarantee an accurate reproduction (see Fig. 2). The reason is that the controlled DS (5) converges with an exponentially stable dynamics which contradicts the demonstrations. More formally, $\exists \beta > 0$ such that $\|x(t) - x^\star\|_1 \leq \beta \|x(0) - x^\star\|_1 e^{-ct}$, where $\|\cdot\|_1$ is the l_1-norm and $\mu_1(A^k + U^k) \leq -c < 0, \forall k$. An approach to alleviate this issue is presented in the next section.

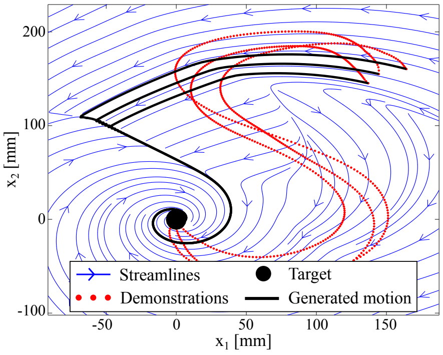


Fig. 2. A globally contracting motion generated by the controlled system (5). The control action guarantees global convergence to $x^\star = [0, 0]^{\mathrm{T}}$, but it affects the accurate reproduction of the demonstrated data.

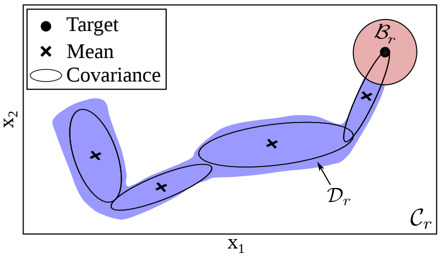


Fig. 3. The state space partitioned into three regions. $\mathcal{D}_r$ is the demonstration area, while the union $\mathcal{C}_r \cup \mathcal{B}_r$ is a contracting region.

2.4. *Accurate Reproduction of Stable Motions*

We propose a modification of the control law in (5) which improves the reproduction accuracy and preserves the convergence properties of the controlled DS. As shown in Fig. 3, the state space is divided into three regions: the contraction region $\mathcal{C}_r$, a ball of radius r centered at the target $\mathcal{B}_r$ and the demonstration area $\mathcal{D}_r$. Given $\mathcal{C}_r$, $\mathcal{B}_r$ and $\mathcal{D}_r$, we propose a stabilizing control input:

$$\dot{x} = \underbrace{\sum_{k=1}^{K} h^k(x)\left(A^k x + b^k\right)}_{System} + \underbrace{\omega(x,t)\left(\sum_{k=1}^{K} h^k(x)\left(U^k x - b^k\right)\right)}_{Control} \tag{6}$$

The activation function $\omega(x,t) \in \mathbb{R}$ in (6) is computed as:

$$\begin{cases} \dot{\omega}(x,t) = -\gamma(\omega(x,t) - c(x)) & t < t_{max} \\ \omega(x,t) = 1 & t \geq t_{max} \end{cases} \tag{7}$$

where $c(x) = 1$ if $x \in \mathcal{C}_r \cup \mathcal{B}_r$, $c(x) = 0$ if $x \in \mathcal{D}_r$ and $\omega(x_0, t_0) = c(x_0)$, where x_0 is a given initial state.

Theorem 2. *The dynamical system in (6), with activation function (7), globally asymptotically converges to $x^\star$*

Proof. Being interested in the asymptotic stability, we have to analyze the convergence properties of (6) for $t \to \infty$. We can notice that, for $\omega(x,t) = 1$, the controlled DS in (6) and (5) are the same. Recalling that (5) is globally exponentially stable, and considering that $\omega(x,t) = 1$ for $t \geq t_{max}$, we can conclude that (6) globally asymptotically converges to $x^\star$. □

The value of $\gamma > 0$ in (7) can be chosen considering that, in practice, $\omega(x,t) = c(x)$ after $5/\gamma$ s. We activate the controller by setting $c(x) = 1$ inside the region $\mathcal{C}_r \cup \mathcal{B}_r$. Trajectories that start in $\mathcal{C}_r \cup \mathcal{B}_r$ follow an exponentially convergent path unless they reach the equilibrium $x^\star$ or they enter $\mathcal{D}_r$. The control input is smoothly deactivated ($c(x) = 0$) inside the demonstration area $\mathcal{D}_r$ to allow an accurate reproduction. The control action is activated if the system has not converged within t_{max} seconds. This guarantees that the generated motion trajectories reach the target from any initial state. In our experiments, we set t_{max} to twice the maximum time duration of the demonstrations. This prevents the DS to converge to spurious equilibria inside $\mathcal{D}_r$ for badly initialized GMR. As empirically shown in Sec. 3, a GMR with a proper number of components in general does not generate spurious attractors within $\mathcal{D}_r$. For each $x \in \mathcal{D}_r$, in fact, the GMR generates a velocity close to the demonstrated one(s). If demonstrated velocities are zero only at the target, the generated velocity will not drop to zero far from the given target. It is worth noticing that eventual spurious attractors will only affect the accuracy of the proposed approach, but not its stability.

The demonstration area $\mathcal{D}_r$ is constructed off-line using the approach in [6]. Each training point $\mathcal{D} = \{x_{t,i}\}_{t=1,i=1}^{T,N}$ is clustered into K clusters $\mathcal{D}^k$, where K is the number of Gaussian components. A training point $x_{t,i}$ belongs to $\mathcal{D}^k$ if $\mathcal{N}\left(x_{t,i}|\mu_x^k, \Sigma_x^k\right) > \mathcal{N}\left(x_{t,i}|\mu_x^j, \Sigma_x^j\right)$, $\forall j \neq k$, where $\mathcal{N}(\cdot)$ is the probability that $x_{t,i}$ is generated from the k-th component. Given the K clusters, a set of K scalars is computed as:

$$\delta^k = \alpha \min_{x_{t,i} \in \mathcal{D}^k} \mathcal{N}\left(x_{t,i}|\mu_x^k, \Sigma_x^k\right) \quad (8)$$

where $0 < \alpha \leq 1$ is a constant. δ^k is the probability of the point in $\mathcal{D}^k$ located at the maximum distance from μ^k. Hence, the region $\mathcal{D}_{tot} = \{x_{t,i} : \mathcal{N}\left(x_{t,i}|\mu_x^k, \Sigma_x^k\right) \geq \delta^k\}$ contains all training data points. The demonstration area $\mathcal{D}_r$ is then $\mathcal{D}_r = \mathcal{D}_{tot} \backslash \mathcal{B}_r$. The scalar α in (8) defines the area of $\mathcal{D}_r$. Bigger values of α result in tighter $\mathcal{D}_r$ around the demonstrations. In all the experiments, we found that $\alpha = 0.1$ prevents trajectories starting in $\mathcal{D}_r$ to exit from the demonstration area. It is worth noticing that the construction of $\mathcal{D}_r$ is computationally efficient. Indeed, the probabilities $\mathcal{N}\left(x_{t,i}|\mu_x^k, \Sigma_x^k\right)$ in (8) are used to train the GMR and come at no extra cost. Therefore, we just have to find the minimum value for K vectors, multiply the obtained K values by a scalar α, and store the results in $\delta^k, k = 1, .., K$.

At run time, we firstly check if a state x belongs to $\mathcal{B}_r$, i.e. if $\|x - x^\star\| \leq r$. If x is not part of $\mathcal{B}_r$, we have to determine whether x belongs to $\mathcal{D}_r$ or $\mathcal{C}_r$. The state $x \in \mathcal{D}_r$ if there exists a k such that $\mathcal{N}\left(x|\mu_x^k, \Sigma_x^k\right) \geq \delta^k, k = 1, .., K$, otherwise $x \in \mathcal{C}_r$. This procedure is computationally efficient. Indeed, the probabilities $\mathcal{N}\left(x|\mu_x^k, \Sigma_x^k\right), k = 1, .., K$, are used to compute $h^k(x)$ (see Tab. 1) and come at no extra cost. Therefore, one has only to check if the K values $\mathcal{N}\left(x|\mu_x^k, \Sigma_x^k\right) \geq \delta^k$. In case $x \in \mathcal{C}_r \cup \mathcal{B}_r$ or $t > t_{max}$, the controller is activated, which only requires to sum up the K matrices A^k and U^k.

3. Experimental Results

In this section, we evaluate the performance of Contracting GMR (C-GMR), in terms of reproduction accuracy and training time. In order to underline basic differences between C-GMR and the Lyapunov based approaches, we perform a comparison with SEDS [7] and SEDSII [9].

3.1. Benchmark and metrics

The LASA Handwritten[1] dataset is used as a benchmark to test stable motion generation with DS. The dataset contains 20 motions in 2D. Each motion ends in $x^\star = [0, 0]^\mathrm{T}$ and is demonstrated three times. Data are collected at 50 Hz.

Two metrics are used to measure the reproduction accuracy. One is the area between each demonstration $\mathcal{D}_i \in \mathcal{D}$ and the trajectory $\mathcal{T}$ generated by C-GMR (SEDS or SEDSII) starting from the first point in $\mathcal{D}_i$. Both $\mathcal{D}_i$ and $\mathcal{T}$ end at the same target. Hence, the area among $\mathcal{D}_i$ and $\mathcal{T}$ is a suitable measure of the reproduction error. Since $\mathcal{D}_i$ and $\mathcal{T}$ have a different number of points, we first leverage multi-dimensional dynamic time warping (DTW) [19] to find the optimal non-linear match between the two trajectories and then compute the area among them. The area metric measures how well the learned DS preserves the shape of the motion. To measure how the DS preserves the kinematics of the demonstrations, we use the velocity error [8] $V_e = \sqrt{\frac{1}{NT}\sum_{t,i=1}^{T,N} \|\dot{x}_{t,i} - f(x_{t,i})\|^2}$.

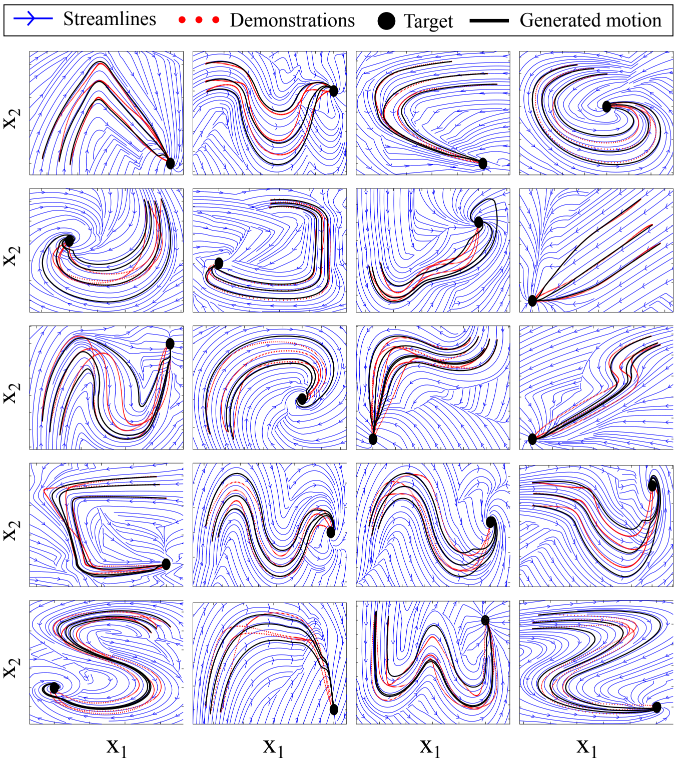


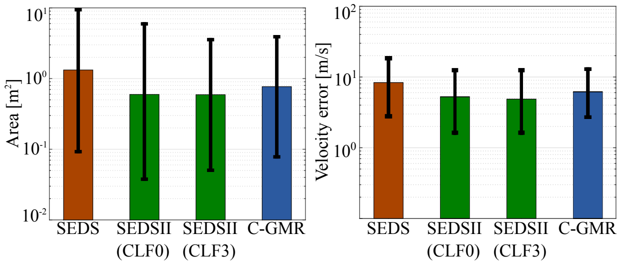
Fig. 4. Stable motions learned by C-GMR on the LASA dataset.

3.2. Reproduction accuracy

We compare C-GMR with SEDS and SEDSII in terms of reproduction error. The sampling time for reproduction is set to 2ms. SEDSII requires to predefine the form of the control Lyapunov function (CLF) depending on the form of the motion. As we chose not to adapt any parameter to each motion separately but to run the experiments unsupervised, we tested two different parametrization of CLF, namely CLF0 and CLF3. CLF0 has no asymmetric components, i.e $CLF0 = x^\mathrm{T} P^0 x$. CLF3 has three asymmetric components, i.e. $CLF3 = CLF0 + \sum_{l=1}^{3} \beta^l(x)\left(x^\mathrm{T} P^l (x - \mu^l)\right)^2$. As detailed in [9], β^l, P^l and μ^l are learned from demonstrations by solving a constrained optimization problem.

A DS for each motion in the dataset is learned by GMR considering all available demonstrations. The dataset contains motions with different complexity. Complex motions usually require many components to be accurately represented. To maximize the accuracy, the number of components is computed by means of Bayesian information criterion (BIC) [20]. C-GMR is used to stabilize $f(x)$ at run time. As qualitatively

[1] The source code for SEDS, SEDSII and the LASA dataset are available on-line: lasa.epfl.ch/sourcecode/



(a) (b)

Fig. 5. Overall accuracy for SEDS, SEDII and C-GMR on the LASA dataset. Black error bars are 10% and 90% quantiles of the median value.

shown in Fig. 4, C-GMR accurately learns complex point-to-point motions while guaranteeing global convergence.

GMR, SEDS and SEDSII require the numerical solution of optimization problems, for which different initial conditions lead to different results. In order to evaluate the typical performance, we tested each approach 10 times for each motion. Overall accuracy is shown in Fig. 5. For a better visualization, we use a logarithmic scale for the ordinate axis. As the reproduction error is not normally distributed, we consider the median instead of the mean. To indicate the maximal and minimal deviation from the typical performance, we provide the location of the 10% and the 90% quantiles (black error bars in Fig. 5). Results for C-GMR are obtained by setting the radius of $\mathcal{B}_r$ equal to the 15% of the distance between the starting point and the target.

Quantitative results in Fig. 5 show that SEDS has median error area of 1.3m^2 and a median velocity error of 8.3m/s respectively. This means that most of the motions in the dataset cannot be learned with the Lyapunov function $x^{\mathrm{T}}x$ used in SEDS. SEDSII CLF0 and CLF3 have almost the same (best) median accuracy ($0.596\,\mathrm{m}^2$ and $0.591\,\mathrm{m}^2$ for the area error, $5.2\,\mathrm{m/s}$ and $4.9\,\mathrm{m/s}$ for V_e), meaning that most of the considered motions can be learned with the Lyapunov function $x^{\mathrm{T}}P^0x$. The proposed C-GMR has a smaller median accuracy ($0.768\,\mathrm{m}^2$ and $6.1\,\mathrm{m/s}$) compared to SEDSII, but higher accuracy than SEDS. C-GMR loses accuracy when the state reaches the ball $\mathcal{B}_r$ around the target. Inside $\mathcal{B}_r$, in fact, the motion follows an exponentially converging dynamics.

3.3. Training time

We investigate the extra training time introduced by C-GMR, i.e. the time required to execute Algorithm 1 and to compute $\mathcal{D}_r$. The execution time of Algorithm 1 depends on the number K of Gaussians and on the dimension m of the state space ($A, U \in \mathbb{R}^{m\times m}$). The execution time of our unoptimized Matlab implementation of Algorithm 1 is shown in Fig. 6. The left graph is obtained by fixing $m = 2$ (as in the LASA dataset) and varying K, the right graph by fixing $K = 10$ and varying m. The execution time is below $0.6\,\mathrm{ms}$.

The time required to construct $\mathcal{D}_r$ depends on K and on the number of training points T. The execution time of our unoptimized Matlab implementation is shown in Fig. 7. The left graph is obtained by keeping $T = 300$ fixed and varying K, the right graph by fixing $K = 10$ and varying T. The execution time is always below $0.1\,\mathrm{ms}$.

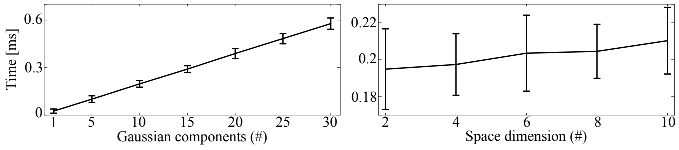


Fig. 6. Execution time of Algorithm 1, averaged over 100 executions.

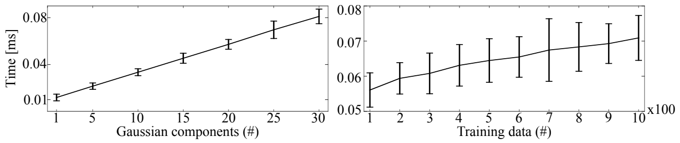


Fig. 7. Time required to compute $\mathcal{D}_r$, averaged over 100 executions.

We use the same configuration of the previous experiment to compare the training times of C-GMR, SEDS and SEDSII. Overall training time is shown in Fig. 8. For a better visualization, we use a logarithmic scale for the ordinate axis. SEDS training step requires about $6.5\,\mathrm{s}$ (median value) for each motion, while SEDSII CLF0 and CLF3 spend $0.76\,\mathrm{s}$ and $2.7\,\mathrm{s}$ respectively. C-GMR has a (best) median training time of $0.58\,\mathrm{s}$. It is worth noticing that C-GMR time in Fig. 8 includes the time needed to compute the region $\mathcal{D}_r$ and to run Algorithm 1. As shown in the previous experiment, this time is at least three orders of magnitude smaller than GMM time.

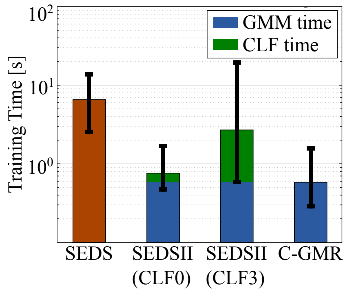
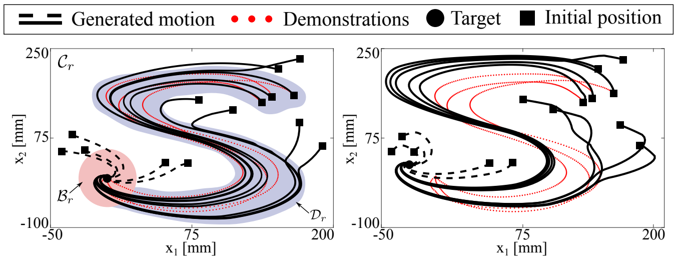


Fig. 8. Overall training time for SEDS, SEDII and C-GMR on the LASA dataset. Black error bars are 10% and 90% quantiles of the median value.

3.4. Generalization to different initial/target positions

Motion generation based on DS has the advantage to be robust to changes in the initial and target positions. In particular, stable DS are able to generate convergent trajectories for any initial/target position. The stability of C-GMR is mathematically proven in *Theorem 2*. Nevertheless, it is interesting to qualitatively show how the learned DS adapts to different initial/target positions. As a proof of concept, we train C-GMR on the S-shape motion in the LASA dataset. Figure 9(a) shows the trajectories generated from different initial positions. As expected, all the trajectories converge towards the target. Black solid lines are trajectories that start or enter in $\mathcal{D}_r$. Figure 9(a) shows how the system smoothly transits between the different region. This is due to the smooth activation function in (7). Black dashed lines are trajectories that start and remain in



(a)

(b)

Fig. 9. Generalization of the S-shape motion. (a) Trajectories generated from different initial positions. (b) Trajectories converging to a different goal.

$\mathcal{C}_r \cup \mathcal{B}_r$ and converge exponentially. Figure 9(b) shows DS trajectories converging to the new goal $x^\star = [-25, 25]^{\mathrm{T}}$.

3.5. *Discussion*

The experimental evaluation shows that the proposed C-GMR is less accurate than SEDSII. This is because SEDSII is optimized along the entire motion, while C-GMR loses accuracy when the state reaches the ball $\mathcal{B}_r$ (see Fig. 3) around the target. In general, contraction theory allows to consider a coordinate transformation $\Theta(x)$. In our future research, we will focus on learning from demonstration a suitable coordinate transformation $\Theta(x)$ that maps the input space into a space where demonstrations can be more accurately reproduced.

Regarding the training time, C-GMR outperforms SEDSII. C-GMR, in fact, stabilizes the dynamical system at run time, without learning steps. The control input is automatically computed given GMR parameters, and (as shown in Sec. 3.3), the time to compute the control input is practically negligible. A possible application of C-GMR is incremental learning, where GMR parameters are continuously updated as new demonstrations of the task are provided. On the contrary, SEDSII is preferable in applications which require high accuracy and for which the training time is not a limitation.

A general assumption behind stable motion generation with autonomous DS is that the demonstrations do not explicitly show spurious equilibria. In case demonstrated velocities drop to zero far from the given target, the desired behavior is probably to stop for a certain time and then to reach the goal. A unique autonomous DS is not suitable to execute this task. A solution is to use one autonomous DS for each attractor and to switch among them as soon as one equilibrium is reached.

4. Conclusion

This work investigated the possibility of applying Contraction theory to learn point-to-point motions via dynamical systems and Gaussian mixture regression. We derived sufficient stability conditions for a dynamical system represented by GMR by applying contraction analysis. An automatic and computationally efficient procedure is developed to compute a stabilizing control input given the GMR parameters. At run time, the control action is smoothly activated or deactivated using an activation function. The combination of the control action and the activation function significantly improves the reproduction accuracy without affecting the stability of the overall system. The resulting algorithm, called Contracting GMR (C-GMR), has been tested on a public dataset consisting of complex 2D motion, showing promising results in terms of accuracy and training time. We also performed an experimental comparison with Lyapunov based approaches, useful to understand the main differences between the different approaches. Our next research will focus on extending C-GMR to other regression techniques, and on testing C-GMR on real robots.

Acknowledgements

This work has been supported by the Technical University of Munich, International Graduate School of Science and Engineering (IGSSE).

References

[1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer Handbook of Robotics*. Springer Berlin Heidelberg, 2008, pp. 1371–1394.

[2] D. Lee and C. Ott, "Incremental kinesthetic teaching of motion primitives using the motion refinement tube," *Autonomous Robots*, vol. 31, no. 2, pp. 115–131, 2011.

[3] M. Saveriano, S. An, and L. Lee, "Incremental kinesthetic teaching of end-effector and null-space motion primitives," in *International Conference on Robotics and Automation*, 2015, pp. 3570–3575.

[4] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *Transactions on Systems, Man, and Cybernetics. Part B*, vol. 37, no. 2, pp. 286–298, 2007.

[5] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, and S. Schaal, "Dynamical Movement Primitives: learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.

[6] S. M. Khansari-Zadeh and A. Billard, "BM: An iterative algorithm to learn stable non-linear dynamical systems with Gaussian Mixture Models," *Int. Conf. on Rob. and Aut.*, pp. 2381–2388, 2010.

[7] ——, "Learning stable non-linear dynamical systems with Gaussian Mixture Models," *Trans. on Rob.*, vol. 27, no. 5, pp. 943–957, 2011.

[8] A. Lemme, F. Reinhart, K. Neumann, and J. J. Steil, "Neural learning of vector fields for encoding stable dynamical systems," *Neurocomputing*, vol. 141, pp. 3–14, 2014.

[9] S. M. Khansari-Zadeh and A. Billard, "Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions," *Rob. And Auton. Systems*, vol. 62, no. 6, pp. 752–765, 2014.

[10] K. Neumann and J. J. Steil, "Learning robot motions with stable dynamical systems under diffeomorphic transformations," *Robotics and Autonomous Systems*, vol. 70, pp. 1–15, 2015.

[11] P. Perrin and P. Schlehuber-Caissier, "Fast diffeomorphic matching to learn globally asymptotically stable nonlinear dynamical systems," *Systems & Control Letters*, vol. 96, pp. 51–59, 2016.

[12] M. Saveriano and L. Lee, "Point cloud based dynamical system modulation for reactive avoidance of convex and concave obstacles," in *Intl Conf. on Intelligent Robots and Systems*, 2013, pp. 5380–5387.

[13] M. Saveriano, F. Hirt, and L. Lee, "Human-aware motion reshaping using dynamical systems," *Pattern Recognition Letters*, 2017.

[14] M. Saveriano and L. Lee, "Distance based dynamical system modulation for reactive avoidance of moving obstacles," in *Intl Conf. on Robotics and Automation*, 2014, pp. 5618–5623.

[15] Z. Huang and C. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[16] W. Lohmiller and J. Slotine, "On Contraction analysis for nonlinear systems," *Automatica*, vol. 34, no. 6, pp. 683–696, 1998.

[17] H. Ravichandar and A. Dani, "Learning contracting nonlinear dynamics from human demonstrations for robot motion planning," in *Dynamics, Systems and Control Conference*, 2015.

[18] W. Wang and J. Slotine, "On Partial Contraction analysis for coupled nonlinear oscillators," *Biol. Cybern.*, vol. 92, no. 1, pp. 38–53, 2005.

[19] P. Sanguansat, "Multiple multidimensional sequence alignment using generalized dynamic time warping," *Transactions on Mathematics*, vol. 11, no. 8, pp. 668–678, 2012.

[20] G. Schwarz, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.