Fast diffeomorphic matching to learn globally asymptotically stable nonlinear dynamical systems

Nicolas Perrin[a], Philipp Schlehuber-Caissier[a],

[a] *Sorbonne Universités, UPMC Univ Paris 06, CNRS, UMR 7222, the Institute of Intelligent Systems and Robotics (ISIR), 4 place Jussieu, 75005, Paris, France (e-mail:* {perrin, schlehuber}@isir.upmc.fr*).*

Abstract

We propose a new diffeomorphic matching algorithm and use it to learn nonlinear dynamical systems with the guarantee that the learned systems have global asymptotic stability. For a given set of demonstration trajectories, and a reference globally asymptotically stable time-invariant system, we compute a diffeomorphism that maps forward orbits of the reference system onto the demonstrations. The same diffeomorphism deforms the whole reference system into one that reproduces the demonstrations, and is still globally asymptotically stable.

Keywords:
Nonlinear dynamical systems, Diffeomorphic mapping, Imitation learning, Lyapunov stability, Dynamical movement primitives

1. Introduction

We consider the problem of learning dynamical systems (DS) from demonstrations. More precisely, given a list of trajectories $(\mathbf{x}_i(t))$ observed as timed sequences of points in $\mathbb{R}^d$, the objective is to build a smooth autonomous system $\dot{\mathbf{x}} = f(\mathbf{x})$ (i.e. a vector field) that reproduces the demonstrations as closely as possible.

The ability to construct such DS is an important skill in imitation learning (see for example [1]). The learned systems can be used as dynamical movement primitives generating goal-directed behaviors [2].

Modeling movement primitives with DS is convenient for closed loop implementations, and their generalization to unseen parts of the state space provides robustness to spatial perturbations. Moreover, the choice of autonomous (i.e. time-invariant) systems, while not always suitable or preferable, is interesting in many situations as they are inherently robust to temporal perturbations.

The most common movement primitives consist of motions that converge toward a single targeted configuration. They correspond to globally asymptotically stable DS. But classical learning algorithms cannot provide the guarantee that their output is always globally asymptotically stable. They might produce DS with instabilities or spurious attractors. This issue has recently been studied by Khansari-Zadeh and Billard [3, 4] who proposed several approaches to learn globally asymptotically stable nonlinear DS. One of the main ideas they investigated consists in learning a Lyapunov function candidate (or simply Lyapunov candidate[1]) L that is highly compatible with the demonstrations in the following sense: at almost every point $\mathbf{x}_i(t_j)$, the estimated or measured velocity $\mathbf{v}_i(t_j)$ is such that its scalar product with the gradient of L is negative: $\mathbf{v}_i(t_j) \cdot \nabla L(\mathbf{x}_i(t_j)) < 0$. Once L is found, a learning algorithm optimizes a weighted sum of DS that admit L as a common Lyapunov function, therefore ensuring the global asymptotic stability of the resulting DS. Alternatively, L can be used to modify movement primitives by correcting trajectories whenever they would violate the compatibility condition.

The main limitation of this method comes from the difficulty to find good Lyapunov candidates. In SEDS (*Stable Estimator of Dynamical Systems*), one of the first algorithms proposed by Khansari-Zadeh and Billard, the Lyapunov function is set to be the l^2-norm squared ($\|\cdot\|^2$), which means that all trajectories produced by the learned DS are such that the distance to the target is monotonically decreasing. In their more recent algorithm CLF-DM (*Control Lyapunov Function-based Dynamic Movements*), the search for a Lyapunov candidate is done among a set called Weighted Sums of Asymmetric Quadratic Functions (WSAQF). It highly increases the set of DS that can be learned, but the restrictions remain significant as the search is limited to a small convex subset of the set of Lyapunov candidates.

To go further, Neumann and Steil [6] suggested to initially compute a Lyapunov candidate with the above method, and then apply a simple diffeomorphism (of the form $\mathbf{x} \mapsto \eta(\mathbf{x})\mathbf{x}$, with $\eta(\mathbf{x}) \in \mathbb{R}_{\geq 0}$) that deforms the space and transforms the Lyapunov candidate into the function $\mathbf{x} \mapsto \|\mathbf{x}\|^2$, thus simplifying the trajectories of the demonstrations. In the deformed space, an algorithm like SEDS is then more likely to learn a globally asymptotically stable DS that reproduces faithfully the demonstrations.

In this paper, we propose a more direct diffeomorphism-based approach. Our contribution is twofold.

[1] In this paper (cf. Definition 1), a Lyapunov function candidate is a C^1 function from $\mathbb{R}^d$ to $\mathbb{R}_{\geq 0}$, radially unbounded, taking the value 0 at a target point $\mathbf{x}^*$ and with no other local extremum. We generally assume $\mathbf{x}^* = \mathbf{0}$.

- First, we introduce a new algorithm for diffeomorphic matching (Sections 2 and 3) and show from experimental comparisons that it tends to be one or two orders of magnitude faster than a state-of-the-art algorithm.
- Then, we explain how it can be used to directly map simple trajectories of a DS like $\dot{\mathbf{x}} = -\mathbf{x}$ onto the trajectories of the training data (Section 4). This gives a new way to generate Lyapunov candidates as well as globally asymptotically stable smooth autonomous systems reproducing the demonstrations.

The most direct applications of this work are in motor control and robotics, but we believe that learning globally asymptotically stable nonlinear DS and computing Lyapunov candidates can be useful for various types of systems and control design problems.

2. Diffeomorphic locally weighted translations

Given a smooth (symmetric positive definite) kernel function $k_\rho(\mathbf{x},\mathbf{y})$, depending on some parameter ρ, such that $\forall \mathbf{x}, k_\rho(\mathbf{x},\mathbf{x}) = 1$ and $k_\rho(\mathbf{x},\mathbf{y}) \to 0$ when $\|\mathbf{y}-\mathbf{x}\| \to \infty$, given a "direction" $\mathbf{v} \in \mathbb{R}^d$ and a "center" $\mathbf{c} \in \mathbb{R}^d$, we consider the following *locally weighted translation*:

$$\psi_{\rho,\mathbf{c},\mathbf{v}}(\mathbf{x}) = \mathbf{x} + k_\rho(\mathbf{x},\mathbf{c})\mathbf{v}.$$

Theorem 1. *If* $\forall(\mathbf{x},\mathbf{y}) \in \mathbb{R}^d \times \mathbb{R}^d$, $\frac{\partial k_\rho}{\partial \mathbf{x}}(\mathbf{x},\mathbf{y}) \cdot \mathbf{v} > -1$, *then* $\psi_{\rho,\mathbf{c},\mathbf{v}}$ *is a smooth (C^∞) diffeomorphism.*

Proof. For a given $\mathbf{x} \in \mathbb{R}^d$, let us try to find $\mathbf{y} \in \mathbb{R}^d$ such that $\psi_{\rho,\mathbf{c},\mathbf{v}}(\mathbf{y}) = \mathbf{x}$. This can be rewritten $\mathbf{y} = \mathbf{x} - k_\rho(\mathbf{y},\mathbf{c})\mathbf{v}$, so we know that $\mathbf{y}$ must be of the form $\mathbf{x} + r\mathbf{v}$. The equation becomes $\psi_{\rho,\mathbf{c},\mathbf{v}}(\mathbf{x}+r\mathbf{v}) = \mathbf{x}$, i.e.: $r\mathbf{v}+k_\rho(\mathbf{x}+r\mathbf{v},\mathbf{c})\mathbf{v} = \mathbf{0}$. If $\mathbf{v} = \mathbf{0}$, $\psi_{\rho,\mathbf{c},\mathbf{v}}$ is the identity (and a smooth diffeomorphism), and $\mathbf{y} = \mathbf{x}$. Otherwise, solving $\psi_{\rho,\mathbf{c},\mathbf{v}}(\mathbf{y}) = \mathbf{x}$ amounts to solving $r + k_\rho(\mathbf{x} + r\mathbf{v},\mathbf{c}) = 0$.

Let us define:

$$h_\mathbf{x} : r \in \mathbb{R} \mapsto r + k_\rho(\mathbf{x} + r\mathbf{v},\mathbf{c}) \in \mathbb{R}.$$

If $\frac{\partial k_\rho}{\partial \mathbf{x}}(\mathbf{x},\mathbf{c}) \cdot \mathbf{v} > -1$, we get: $\forall r \in \mathbb{R}$, $\frac{dh_\mathbf{x}}{dr}(r) > 0$. Because of the absolute monotonicity of $h_\mathbf{x}$, and since $h_\mathbf{x}(r)$ tends to $-\infty$ when r tends to $-\infty$, and to $+\infty$ when r tends to $+\infty$, we deduce that there exists a unique $s_{\rho,\mathbf{c},\mathbf{v}}(\mathbf{x}) \in \mathbb{R}$ such that $h_\mathbf{x}(s_{\rho,\mathbf{c},\mathbf{v}}(\mathbf{x})) = 0$. It follows that the equation $\psi_{\rho,\mathbf{c},\mathbf{v}}(\mathbf{y}) = \mathbf{x}$ has a unique solution: $\mathbf{y} = \mathbf{x} + s_{\rho,\mathbf{c},\mathbf{v}}(\mathbf{x})\mathbf{v}$. We conclude that $\psi_{\rho,\mathbf{c},\mathbf{v}}$ is invertible, and:

$$\psi^{-1}_{\rho,\mathbf{c},\mathbf{v}}(\mathbf{x}) = \mathbf{x} + s_{\rho,\mathbf{c},\mathbf{v}}(\mathbf{x})\mathbf{v}.$$

The implicit function theorem can be applied to prove that $s_{\rho,\mathbf{c},\mathbf{v}}$ is smooth, and as a consequence $\psi_{\rho,\mathbf{c},\mathbf{v}}$ is a smooth diffeomorphism. □

With Gaussian Radial Basis Function (RBF) kernel:

We now consider the following kernel (with $\rho \in \mathbb{R}_{>0}$):

$$k_\rho(\mathbf{x},\mathbf{y}) = \exp\left(-\rho^2\|\mathbf{x}-\mathbf{y}\|^2\right).$$

We have:

$$\frac{\partial k_\rho}{\partial \mathbf{x}}(\mathbf{x},\mathbf{y}) \cdot \mathbf{v} = -2\rho^2 \exp\left(-\rho^2\|\mathbf{x}-\mathbf{y}\|^2\right)(\mathbf{x}-\mathbf{y}) \cdot \mathbf{v},$$

with the lower bound:

$$\frac{\partial k_\rho}{\partial \mathbf{x}}(\mathbf{x},\mathbf{y}) \cdot \mathbf{v} \geq -2\rho^2 \exp\left(-\rho^2\|\mathbf{x}-\mathbf{y}\|^2\right)\|\mathbf{x}-\mathbf{y}\|.\|\mathbf{v}\|.$$

The expression on the right takes its minimum for $\|\mathbf{x}-\mathbf{y}\| = \frac{1}{\sqrt{2}\rho}$, which yields:

$$\frac{\partial k_\rho}{\partial \mathbf{x}}(\mathbf{x},\mathbf{y}) \cdot \mathbf{v} \geq -\sqrt{2}\|\mathbf{v}\|\rho \exp\left(-\frac{1}{2}\right).$$

We pose $\rho_{\max}(\mathbf{v}) = \frac{1}{\sqrt{2}\|\mathbf{v}\|}\exp\left(\frac{1}{2}\right)$. Applying Theorem 1, $\mathbf{v} = 0$ or $\rho < \rho_{\max}(\mathbf{v})$ implies that $\psi_{\rho,\mathbf{c},\mathbf{v}}$ is a smooth diffeomorphism. In that case, $s_{\rho,\mathbf{c},\mathbf{v}}(\mathbf{x})$, and as a result $\psi^{-1}_{\rho,\mathbf{c},\mathbf{v}}(\mathbf{x})$, can be very efficiently computed with Newton's method.

3. A diffeomorphic matching algorithm

In this section we are interested in the following problem: given two sequences of distinct points $\mathbf{X} = (\mathbf{x}_i)_{i\in\{0,\dots,N\}}$ and $\mathbf{Y} = (\mathbf{y}_i)_{i\in\{0,\dots,N\}}$, compute a diffeomorphism Φ that maps each $\mathbf{x}_i$ onto $\mathbf{y}_i$, either exactly or approximately. More formally, defining $\mathsf{dist}(\mathbf{A},\mathbf{B}) = \frac{1}{N+1}\sum_i \|\mathbf{a}_i - \mathbf{b}_i\|^2$ for two sequences $\mathbf{A}$ and $\mathbf{B}$ of $N+1$ points, and denoting by $\Phi(\mathbf{X})$ the sequence of points $(\Phi(\mathbf{x}_i))_{i\in\{0,\dots,N\}}$, we want to find a diffeomorphism Φ that minimizes $\mathsf{dist}(\Phi(\mathbf{X}),\mathbf{Y})$.

3.1. State of the art

Since the sequences $\mathbf{X}$ and $\mathbf{Y}$ can be very different in shape, to the best of our knowledge the state-of-the-art existing techniques to solve this problem are based on the Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework introduced in the seminal article by Joshi and Miller [7]. Its core idea is to work with a time dependent vector field $v(\mathbf{x},t) \in \mathbb{R}^d$ ($t \in [0,1]$), and define a flow $\phi(\mathbf{x},t)$ via the transport equation:

$$\frac{d\phi(\mathbf{x},t)}{dt} = v(\phi(\mathbf{x},t),t),$$

with $\phi(\mathbf{x},0) = \mathbf{x}$. With a few regularity conditions on v (see [8] for specific requirements), $\mathbf{x} \mapsto \phi(\mathbf{x},t)$ is a diffeomorphism. The resulting diffeomorphism $\Phi(\mathbf{x}) = \phi(\mathbf{x},1)$ is given by:

$$\Phi(\mathbf{x}) = \mathbf{x} + \int_0^1 v(\phi(\mathbf{x},t),t)dt.$$

Using an appropriate Hilbert space, the vector fields $\mathbf{x} \mapsto v(\mathbf{x},t)$ can be associated to an infinitesimal cost whose integration is interpreted as a deformation energy.

Various gradient descent algorithms have been proposed to optimize v with respect to a cost that depends both on the deformation energy and on the accuracy of the mapping, whether the objective is to map curves [9], surfaces [10], or, as in our case, points [11].

3.2. Our algorithm

The LDDMM framework has several advantages. For example, it tries to minimize the deformation, and allows the computation of similarity measures between diffeomorphic geometrical objects. However, Φ is not in closed-form, so once obtained, evaluating it requires an integration that can be slightly time-costly. In our context, it can be necessary to use Φ inside a control law, so its evaluation (and that of Φ^{-1}) must be very fast.

We propose a completely different and novel approach to diffeomorphic matching, based on the diffeomorphic locally weighted translations presented in the previous section, which are functions that can be evaluated extremely quickly.

We fix a number of iterations K, and two parameters $0 < \mu < 1$ and $0 < \beta \leq 1$. K is defined empirically, as the number of iterations required for a good approximation depends on the intrinsic difficulty of the problem. μ is a kind of "safety margin": strictly less than 1, it ensures that the results cannot be arbitrarily close to non-invertible functions. β is similar to a learning rate: a small value allows only small modifications at every iteration, resulting in a slower but usually more stable convergence. Typically, on the examples presented in this paper, we use $K = 150$, $\mu \approx 0.9$ and $\beta \approx 0.5$.

Initially, we define $\mathbf{Z} := \mathbf{X}$. Every iteration updates $\mathbf{Z}$. The j-th iteration can be briefly described by the following steps:

1. we select the point $\mathbf{p}_j$ in $\mathbf{Z}$ that is the furthest from its corresponding target $\mathbf{q}$ in $\mathbf{Y}$ (see lines 7 to 9 in the pseudo-code below);
2. we consider the locally weighted translation $\psi_{\rho_j,\mathbf{p}_j,\mathbf{v}_j}$ of direction $\mathbf{v}_j = \beta(\mathbf{q} - \mathbf{p}_j)$, center $\mathbf{p}_j$, and Gaussian RBF kernel k_{ρ_j}, optimizing $\rho_j \in [0, \mu\rho_{\max}(\mathbf{v}_j)]$ to minimize the error between $\psi_{\rho_j,\mathbf{p}_j,\mathbf{v}_j}(\mathbf{Z})$ and $\mathbf{Y}$;
3. we perform the update: $\mathbf{Z} := \psi_{\rho_j,\mathbf{p}_j,\mathbf{v}_j}(\mathbf{Z})$.

The resulting (smooth) diffeomorphism is the composition of all the locally weighted translations:

$$\Phi = \psi_{\rho_K,\mathbf{p}_K,\mathbf{v}_K} \circ \cdots \circ \psi_{\rho_2,\mathbf{p}_2,\mathbf{v}_2} \circ \psi_{\rho_1,\mathbf{p}_1,\mathbf{v}_1}$$

Here is a description of the algorithm in pseudo-code:

1: **Input:** $\mathbf{X} = (\mathbf{x}_i)_{i\in\{0,\dots,N\}}$ and $\mathbf{Y} = (\mathbf{y}_i)_{i\in\{0,\dots,N\}}$
2: **Parameters:** $K \in \mathbb{N}_{>0}$, $0 < \mu < 1$, $0 < \beta \leq 1$
3:
4: $\mathbf{Z} = (\mathbf{z}_i)_{i\in\{0,\dots,N\}}$
5: $\mathbf{Z} := \mathbf{X}$
6: **for** $j = 1$ to K **do**
7: $\quad m := \underset{i\in\{0,\dots,N\}}{\arg\max} (\|\mathbf{z}_i - \mathbf{y}_i\|)$
8: $\quad \mathbf{p}_j := \mathbf{z}_m$
9: $\quad \mathbf{q} := \mathbf{y}_m$
10: $\quad \mathbf{v}_j := \beta(\mathbf{q} - \mathbf{p}_j)$
11: $\quad \rho_j := \underset{\rho\in[0,\mu\rho_{\max}(\mathbf{v}_j)]}{\arg\min} (\mathrm{dist}(\psi_{\rho,\mathbf{p}_j,\mathbf{v}_j}(\mathbf{Z}), \mathbf{Y}))$
12: $\quad \mathbf{Z} := \psi_{\rho_j,\mathbf{p}_j,\mathbf{v}_j}(\mathbf{Z})$
13: **end for**
14: **return** $(\rho_j)_{j\in\{1,\dots,K\}}$, $(\mathbf{p}_j)_{j\in\{1,\dots,K\}}$, $(\mathbf{v}_j)_{j\in\{1,\dots,K\}}$

Five remarks:

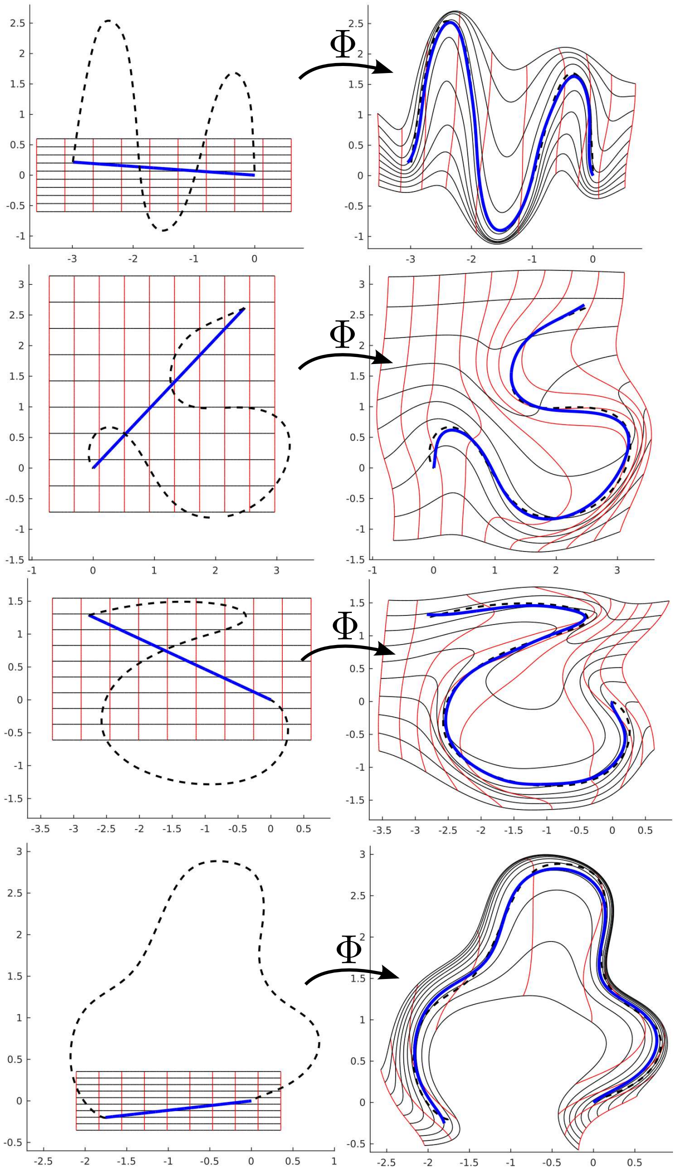

Figure 1: On the left, the dashed curve is a trajectory represented by a sequence of points $\mathbf{Y} = (\mathbf{y}_i)_{i\in\{0,\dots,N\}}$. The solid line is $\mathbf{X} = (\mathbf{y}_0 + \frac{i}{N}(\mathbf{y}_N - \mathbf{y}_0))_{i\in\{0,\dots,N\}}$. The right side shows the result of the application of the diffeomorphism Φ constructed by our algorithm to map $\mathbf{X}$ onto $\mathbf{Y}$.

- Here we have presented a version of the algorithm in which the parameter β is constant, but we can also make it vary iteration after iteration, for example by increasing it towards 1.

- The line 11 of the algorithm performs a nonlinear optimization, but it depends only on one bounded real variable, so a minimum can be found very quickly and precisely.

- We can add a fixed upper bound $\rho_M > 0$ for all ρ_j, and a regularization term in the cost of the optimization problem of line 11, to prevent the diffeomorphism from overly deforming the space to get a perfect matching. Simply using inputs with a dense representation (large value of N) has a similar effect (and it barely slows the algorithm down).

- Again in line 11, dist can be replaced by any distance, e.g. the largest singular value norm of $(\mathbf{X} - \mathbf{Y})$ (with $\mathbf{X}$ and $\mathbf{Y}$ written as $(N + 1)$-by-d matrices).
- Nothing prevents the algorithm from getting stuck in a local minimum, so a general proof of convergence cannot be found. However, as we show in the next sections, experimental results give empirical evidence that the algorithm is efficient and converges quickly in practice, even on difficult matching problems. In future work, we will try to further improve the algorithm and get convergence proofs under realistic assumptions.

3.3. Experimental evaluation

We compared our algorithm to an implementation of diffeomorphic matching in the LDDMM framework developed by J. Glaunès (the "`Matchine`" software [12]).

Given a sequence of points $\mathbf{Y} = (\mathbf{y}_i)_{i\in\{0,\dots,N\}}$ representing a trajectory, we set $\mathbf{X} = (\mathbf{x}_i)_{i\in\{0,\dots,N\}} = \left(\mathbf{y}_0 + \frac{i}{N}(\mathbf{y}_N - \mathbf{y}_0)\right)_{i\in\{0,\dots,N\}}$ and applied our algorithm or the LDDMM algorithm to construct a diffeomorphism Φ such that $\Phi(\mathbf{X})$ and $\mathbf{Y}$ match. Figure 1 displays the result of our algorithm on four 2D trajectories, and Table 1 shows a comparison of the results obtained on these trajectories with our algorithm and the LDDMM algorithm. For each trajectory, we tried with representations as sequences of 21, 51 and 101 points (i.e. $N = 20$, $N = 50$, $N = 100$). For both algorithms, the same parameters were kept across all the trials. In all cases, our algorithm provided a substantial speedup. For example, with $N = 50$, Φ was learned in average 58 times faster and evaluated 240 times faster, while the error $\mathrm{dist}(\Phi(\mathbf{X}), \mathbf{Y})$ was 2.67 times smaller. The tests were made on an Intel(R) Core(TM) i7-4700MQ @ 2.4 GHz with 4GB of RAM.

	N	our algorithm	LDDMM
Learning: average duration of the construction of Φ	20	0.25 s	2.78 s
	50	0.25 s	14.5 s
	100	0.26 s	53.3 s
Forward evaluation: average duration of the computation of $\Phi(\mathbf{X})$	20	3.05 ms	157 ms
	50	3.35 ms	804 ms
	100	3.72 ms	3130 ms
Backward evaluation: average duration of the computation of $\Phi^{-1}(\mathbf{Y})$	20	29.8 ms	145 ms
	50	35 ms	798 ms
	100	38.5 ms	3110 ms
Accuracy of the mapping: average value of $\mathrm{dist}(\Phi(\mathbf{X}), \mathbf{Y})$	20	3.49×10^{-3}	18.2×10^{-3}
	50	8.32×10^{-3}	22.2×10^{-3}
	100	9.51×10^{-3}	22.0×10^{-3}
Generalization: average value of $\mathrm{dist}(\Phi(\mathbf{X}_{1000}), \mathbf{Y}_{1000})$	20	19.8×10^{-3}	20.3×10^{-3}
	50	9.51×10^{-3}	21.6×10^{-3}
	100	11.5×10^{-3}	22.3×10^{-3}

Table 1: Comparison of experimental results for the 4 examples of Figure 1. Remark: standard deviations are negligible for our algorithm: it is deterministic, and the computation times depend almost entirely on the input size (N) and on the fixed number of iterations (K). $\mathbf{Y}$ is obtained by subsampling from an initial recording of 1000 points: $\mathbf{Y}_{1000}$. $\mathbf{X}_{1000}$ is the linear progression from $\mathbf{y}_0$ to $\mathbf{y}_{999}$. To get a sense of how precisely the mapping generalizes around the set of training points, we compute $\mathrm{dist}(\Phi(\mathbf{X}_{1000}), \mathbf{Y}_{1000})$. We observe that for $N = 50$ and $N = 100$, our results are about twice as accurate as the ones obtained with the algorithm based on LDDMM.

4. Learning globally asymptotically stable nonlinear dynamical systems

In this section, we show how a diffeomorphic matching algorithm can be used to learn globally asymptotically stable DS that reproduce demonstration trajectories.

4.1. Definitions and theorems

Remark: we only consider dynamical systems $\dot{\mathbf{x}} = f(\mathbf{x})$ such that $f(\mathbf{x})$ is locally Lipschitz.

Definition 1. *A Lyapunov candidate L is a continuously differentiable function from $\mathbb{R}^d$ to $\mathbb{R}_{\geq 0}$ taking the value 0 at a "target point" $\mathbf{x}^*$, with no other local extremum, and radially unbounded ($||\mathbf{x}|| \to \infty \Rightarrow L(\mathbf{x}) \to \infty$).*

Definition 2. *A Lyapunov candidate L with target point $\mathbf{x}^*$ is said to be compatible with the DS $\dot{\mathbf{x}} = f(\mathbf{x})$ if:*

$$\forall \mathbf{x} \in \mathbb{R}^d,\ \mathbf{x} \neq \mathbf{x}^* \Rightarrow f(\mathbf{x}) \cdot \nabla L(\mathbf{x}) < 0.$$

The following is a classical theorem in Lyapunov stability theory (see for example [13], Chapter 4):

Theorem 2. *If a DS $\dot{\mathbf{x}} = f(\mathbf{x})$ is compatible with some Lyapunov candidate L, then it is globally asymptotically stable.*

Note that Definition 1 is stronger than the usual definition of Lyapunov candidates in that they must have no other local extremum than $\mathbf{x}^*$. This is very important when the objective is to construct globally asymptotically stable DS, as the standard approach is to first compute a good Lyapunov candidate, and then a DS that is compatible with it. But if the gradient of the Lyapunov candidate vanishes at several points (which can be difficult to check), then no DS can be compatible with it. Therefore, it is crucial to ensure *by construction* that the Lyapunov candidate has no undesired extremum.

Definition 3. *We say that L is a Lyapunov function for the DS $\dot{\mathbf{x}} = f(\mathbf{x})$ if it is a Lyapunov candidate compatible with $\dot{\mathbf{x}} = f(\mathbf{x})$.*

Definition 4. *Two DS $\dot{\mathbf{x}} = f(\mathbf{x})$ and $\dot{\mathbf{x}} = g(\mathbf{x})$ are said to be diffeomorphic, or smoothly equivalent, if there exists a diffeomorphism $\Phi : \mathbb{R}^d \to \mathbb{R}^d$ such that:*

$$\forall \mathbf{x} \in \mathbb{R}^d,\ g(\Phi(\mathbf{x})) = J_\Phi(\mathbf{x}) f(\mathbf{x}),$$

where $J_\Phi(x)$ is the Jacobian matrix: $J_\Phi(\mathbf{x}) = \frac{\partial \Phi}{\partial \mathbf{x}}(\mathbf{x})$. If Φ is a C^k-diffeomorphism, then the DS are said to be C^k-diffeomorphic.

Theorem 3. *If two DS $\dot{\mathbf{x}} = f(\mathbf{x})$ and $\dot{\mathbf{x}} = g(\mathbf{x})$ are diffeomorphic, then if one is globally asymptotically stable, both are.*

Proof. Without ambiguity we can call these DS f and g. Let Φ be a diffeomorphism such that $\forall \mathbf{x} \in \mathbb{R}^d,\ g(\Phi(\mathbf{x})) = J_\Phi(\mathbf{x}) f(\mathbf{x})$.

For any forward orbit of f, i.e. any trajectory $(\mathbf{x}(t))_{t\geq 0}$ such that $\dot{\mathbf{x}} = f(\mathbf{x})$, let us consider the trajectory $(\Phi(\mathbf{x}(t)))_{t\geq 0}$. We have:

$$\frac{d}{dt}(\Phi(\mathbf{x}(t))) = J_\Phi(\mathbf{x}(t))\dot{\mathbf{x}}(t) = J_\Phi(\mathbf{x}(t))f(\mathbf{x}(t)) = g(\Phi(\mathbf{x}(t))).$$

This implies that $(\Phi(\mathbf{x}(t)))_{t\geq 0}$ is a forward orbit of g. More generally, any orbit $(\mathbf{y}(t))_{t\geq 0}$ of g can be written $(\Phi(\mathbf{x}(t)))_{t\geq 0}$, with $\mathbf{x}(0) = \Phi^{-1}(\mathbf{y}(0))$, and $(\mathbf{x}(t))_{t\geq 0}$ orbit of f. If f is globally asymptotically stable, then all orbits $(\mathbf{x}(t))_{t\geq 0}$ converge towards some target point $\mathbf{x}^*$, and thus all orbits $(\mathbf{y}(t))_{t\geq 0}$ of g converge towards $\Phi(\mathbf{x}^*)$, which proves that g is globally asymptotically stable. A similar demonstration proves the converse implication. □

Theorem 4. *Let $\dot{\mathbf{x}} = f(\mathbf{x})$ and $\dot{\mathbf{x}} = g(\mathbf{x})$ be two C^1-diffeomorphic DS, and let Φ be a C^1-diffeomorphism such that $\forall \mathbf{x} \in \mathbb{R}^d$, $g(\Phi(\mathbf{x})) = J_\Phi(\mathbf{x})f(\mathbf{x})$. If L is a Lyapunov function for $\dot{\mathbf{x}} = f(\mathbf{x})$, then $L \circ \Phi^{-1}$ is a Lyapunov function for $\dot{\mathbf{x}} = g(\mathbf{x})$.*

Proof. Again, we call these DS f and g. We also pose $M = L \circ \Phi^{-1}$. Using Theorem 2, we know that f is globally asymptotically stable, and by Theorem 3, g is globally asymptotically stable as well. Let $\mathbf{x}^*$ be the target point of f. $\Phi(\mathbf{x}^*)$ is the target point of g. Let us consider a forward orbit $(\mathbf{y}(t))_{t\geq 0}$ of g. It can be written $(\Phi(\mathbf{x}(t)))_{t\geq 0}$, with $(\mathbf{x}(t))_{t\geq 0}$ forward orbit of f (cf. proof of Theorem 3). It follows that $M(\mathbf{y}(t)) = L(\mathbf{x}(t))$, and if $\mathbf{y}(t) \neq \Phi(\mathbf{x}^*)$, i.e. $\mathbf{x}(t) \neq \mathbf{x}^*$, then $\frac{d}{dt}(M(\mathbf{y}(t))) = g(\mathbf{y}(t)) \cdot \nabla M(\mathbf{y}(t)) = \frac{d}{dt}(L(\mathbf{x}(t))) < 0$. Besides, it can be verified that M is a Lyapunov candidate (with target point $\Phi(\mathbf{x}^*)$), so M is a Lyapunov function for g. □

4.2. Overview of the method

The objective of our approach is to learn a smooth diffeomorphism Φ that maps forward orbits of the DS $\dot{\mathbf{x}} = -\mathbf{x}$ (i.e. line segments) onto the observed trajectories (the demonstrations). The mapping obtained is at first purely geometrical, but the initial DS can be transformed into $\dot{\mathbf{x}} = -\gamma(\mathbf{x})\mathbf{x}$, with $\gamma : \mathbb{R}^d \to \mathbb{R}_{>0}$, to adjust the velocities without modifying the forward orbits. If the matching is accurate, the result is that Φ deforms the whole DS $\dot{\mathbf{x}} = -\gamma(\mathbf{x})\mathbf{x}$ into the globally asymptotically stable DS $\dot{\mathbf{x}} = -\gamma(\Phi^{-1}(\mathbf{x}))J_\Phi(\Phi^{-1}(\mathbf{x}))\Phi^{-1}(\mathbf{x})$ that reproduces well the demonstrations. Additionally, since $\mathbf{x} \mapsto \|\mathbf{x}\|$ is a Lyapunov function for $\dot{\mathbf{x}} = -\gamma(\mathbf{x})\mathbf{x}$, $\mathbf{x} \mapsto \|\Phi^{-1}(\mathbf{x})\|$ is a Lyapunov function for $\dot{\mathbf{x}} = -\gamma(\Phi^{-1}(\mathbf{x}))J_\Phi(\Phi^{-1}(\mathbf{x}))\Phi^{-1}(\mathbf{x})$ (cf. Theorem 4), and therefore a Lyapunov candidate highly compatible with the demonstrations. This transformation of a globally asymptotically stable DS into another one via diffeomorphism has strong similarities with the construction of navigation functions proposed in [14], which is based on the fact that navigation properties are invariant under C^k-diffeomorphisms for $k \geq 1$.

Trajectories being represented as sequences of points, the problem of forward orbits mapping can be cast as diffeomorphic matching. In the case of a unique demonstration $\mathbf{Y} = (\mathbf{y}(t_i))_{i\in\{0,\dots,N\}}$, with $t_i = i\Delta t$, we want to find a diffeomorphism that maps $\mathbf{X} = \left(\mathbf{y}(0) + \frac{i}{N}(\mathbf{0} - \mathbf{y}(0))\right)_{i\in\{0,\dots,N\}}$ onto $\mathbf{Y}$ (the trajectory is assumed to arrive at the target: $\mathbf{y}(t_N) = \mathbf{0}$). To do so, we simply use the algorithm presented in Section 3.2. The diffeomorphism Φ_K obtained after K iterations can be such that $\Phi_K(\mathbf{0}) \neq \mathbf{0}$, so we add an extra iteration that picks $\mathbf{p}_{K+1} = \Phi_K(\mathbf{0})$ and $\mathbf{v}_{K+1} = \mathbf{0} - \Phi_K(\mathbf{0})$. This ensures that the final diffeomorphism Φ verifies $\Phi_K(\mathbf{0}) = \mathbf{0}$. Remark: the structure of Φ makes it easy to efficiently compute $J_\Phi(\mathbf{x})$ at any given point.

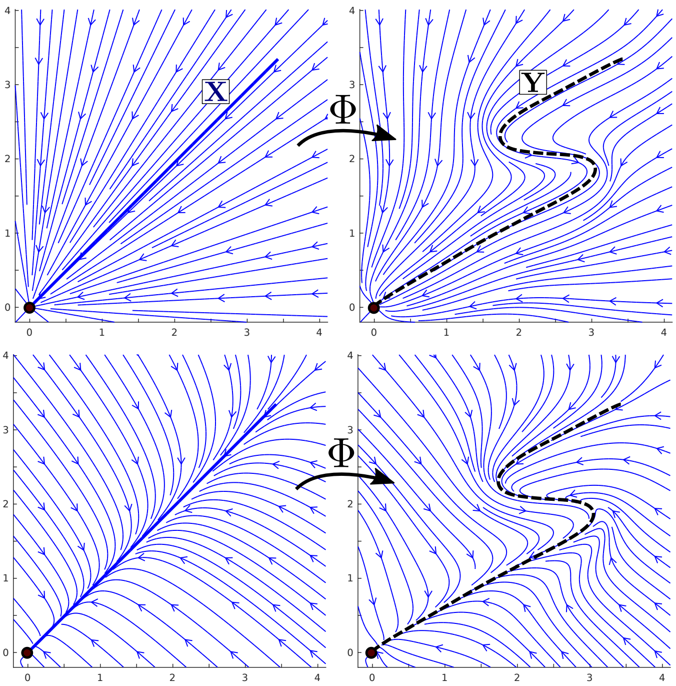

Figure 2: The diffeomorphism Φ, that maps the trajectory $\mathbf{X}$ onto $\mathbf{Y}$, transforms a globally asymptotically stable DS with $\mathbf{X}$ as a forward orbit into a globally asymptotically stable DS with $\mathbf{Y}$ as a forward orbit (*top row*). Using this property, we can modulate the initial vector field while keeping $\mathbf{X}$ unchanged to obtain systems with different behaviors that all reproduce the demonstration $\mathbf{Y}$.

4.3. Results

The top row of Figure 2 shows the result of mapping the straight trajectory $\mathbf{X}$ (on the left) onto the trajectory $\mathbf{Y}$ (on the right). The diffeomorphism Φ that realizes this matching also transforms the entire DS $\dot{\mathbf{x}} = -\mathbf{x}$ into a nonlinear globally asymptotically stable DS that reproduces the trajectory $\mathbf{Y}$ (as the forward orbit of $\mathbf{y}(0)$).

Modifying the initial DS without changing the forward orbit of $\mathbf{x}(0)$ leads, by application of Φ, to another DS that still reproduces $\mathbf{Y}$. On the bottom row of Figure 2, we use a linear system that keeps $\mathbf{x}(0)$ as an eigenvector associated with eigenvalue -1 (ensuring that its forward orbit is not modified), but has a negative eigenvalue of absolute value greater than 1 in the orthogonal direction (unlike the DS $\dot{\mathbf{x}} = -\mathbf{x}$). This results in a transformed DS that “tracks” more agressively the trajectory $\mathbf{Y}$, bringing robustness in the sense that, after a perturbation, the systems goes back quickly to the reference trajectory $\mathbf{Y}$. The DS of the top row corresponds to another notion of robustness, in which the reproduction of the pattern has more importance.

We evaluated our approach on the LASA Handwriting Dataset [15], similarly to [3, 4, 6]. On all cases shown in Figure 4, a set of 7 trajectories ending at the same point demon-

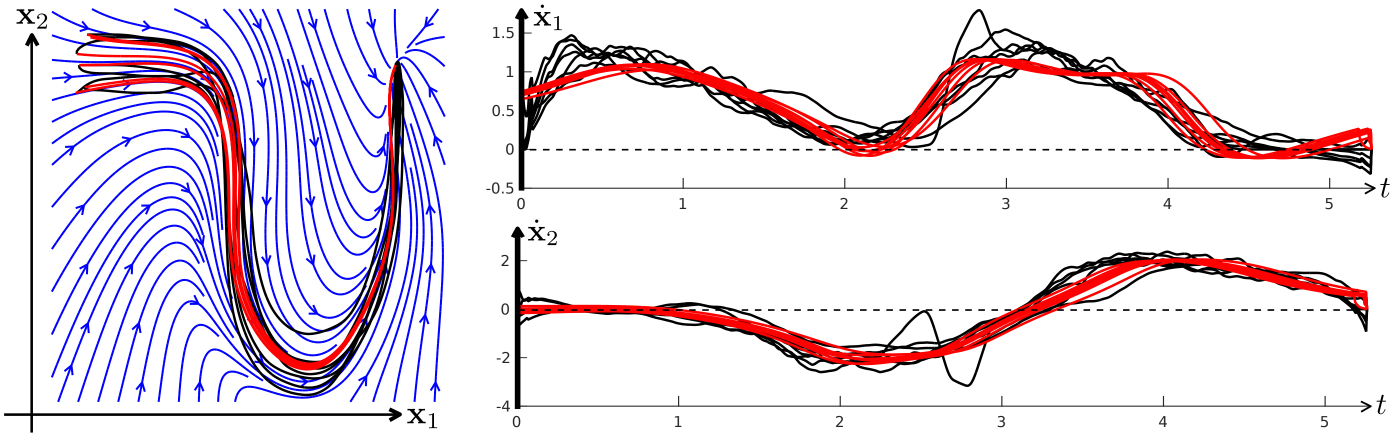

Figure 3: On the left: a smooth autonomous systems, learned with our method, that reproduces a motion pattern (demonstrations are in black, reproduced trajectories in red). The trajectories on the right show that the velocity profiles are quite accurately reproduced as well (again, demonstrations in black and reproductions in red).

strate a single pattern of handwriting motion. For each of these patterns, we create an average timed sequence of points $\mathbf{Y} = (\mathbf{y}(i\Delta t))_{i\in\{0,\ldots,N\}}$ based on the 7 trajectories, and apply our matching algorithm to construct a diffeomorphism Φ that maps $\mathbf{X} = (\frac{N-i}{N}\mathbf{y}(0))_{i\in\{0,\ldots,N\}}$ onto $\mathbf{Y}$. This gives a Lyapunov candidate $\mathbf{x} \mapsto \|\Phi^{-1}(\mathbf{x})\|$. We compare it to the optimized WSAQF Lyapunov candidates obtained with the method of Khansari-Zadeh and Billard [4] also used in [6]. On the *1st column* of Figure 4 are displayed level sets of the WSAQF Lyapunov candidates, and on the *2nd column* level sets of our Lyapunov candidates. We can observe that the level sets of the Lyapunov candidates produced by our method have a richer geometry and exhibit variations that are more suitably adapted to the training data.

The CLF-DM method of Khansari-Zadeh and Billard [4] could be used with these Lyapunov candidates to correct any learned DS and ensure global asymptotic stability. But as mentioned above, the diffeomorphism also provides a way to directly get a globally asymptotically stable DS that reproduces the motion pattern. We define a function $\gamma : \mathbb{R}^d \to \mathbb{R}_{>0}$ such that, starting at $\mathbf{x}(0) = \mathbf{y}(0)$ with $t = 0$, the DS $\dot{\mathbf{x}} = -\gamma(\mathbf{x})\mathbf{x}$ produces a trajectory that passes by the points $\frac{N-i}{N}\mathbf{y}(0)$ at times $i\Delta t$, for $i \in \{1,\ldots,N-1\}$, and converges asymptotically towards $\mathbf{0}$ for $t > (N-1)\Delta t$. A simple choice for γ is $\gamma(\mathbf{x}) = \frac{\|\mathbf{y}(0)\|}{N\Delta t\|\mathbf{x}\|}$ for $\|x\| \geq \frac{\|\mathbf{y}(0)\|}{N}$ and $\gamma(\mathbf{x}) = \frac{\|\mathbf{y}(0)\|}{N}$ otherwise (but it is easy to design a smoother function with the same desired properties). Φ transforms the DS $\dot{\mathbf{x}} = -\gamma(\mathbf{x})\mathbf{x}$ into one that reproduces the demonstrations and their velocity profiles, as shown in Figure 3. The eigenvalue in the direction orthogonal to $\mathbf{y}(0)$ can be adjusted according to the variability of the 7 demonstrations, or to get a better rate of convergence towards the demonstrations (cf. Figure 2). The vector fields obtained with our method are shown on the *4th column* of Figure 4, and the vector fields obtained with the τ-SEDS method of Neumann and Steil [6] based on WSAQF are shown on the *3rd column*.

4.4. Comparison with previous approaches

The existing approaches follow a two-step process:

1. Compute a Lyapunov candidate L, highly compatible with the demonstrations.
2. Compute a DS compatible with L that reproduces the demonstrations.

Neumann and Steil [6] add a diffeomorphic deformation between step 1 and step 2 to simplify the construction of the DS, and Khansari-Zadeh and Billard [4] separate step 1 and step 2 completely, noticing that *any* DS reproducing the demonstrations can be corrected into a globally asymptotically stable DS once the Lyapunov candidate L is known. In both approaches, step 1 is crucial because the Lyapunov candidate restricts the possibilities of the DS of step 2. But a major difficulty is that the set of Lyapunov candidates (Definition 1) is rather ill-behaved in the sense that it is non-convex and not closed under addition or multiplication: the sum or product of two Lyapunov candidates is not necessarily a Lyapunov candidate, as local extrema might appear. To circumvent this difficulty, a solution is to restrict the search to a convex subset of the set of Lyapunov candidates. The computation of the Lyapunov candidates (step 1) is based on WSAQF [4, 6]. Another method could be mentioned: NILC [5, 6], but it does not result in globally valid Lyapunov candidates. As mentioned by Neumann and Steil (Lemma 1 in [6]), any WSAQF Lyapunov candidate L (with target point $\mathbf{0}$) is compatible with the DS $\dot{\mathbf{x}} = -\mathbf{x}$:

$$\forall \mathbf{x} \in \mathbb{R}^d, \mathbf{x} \neq \mathbf{0} \Rightarrow -\mathbf{x} \cdot \nabla L(\mathbf{x}) < 0.$$

Interestingly, the set of Lyapunov candidates that are compatible with a fixed globally asymptotically stable DS (in this case $\dot{\mathbf{x}} = -\mathbf{x}$) is a convex cone. Thanks to this property, searching for a WSAQF Lyapunov candidate can be done efficiently. But the compatibility to $\dot{\mathbf{x}} = -\mathbf{x}$ is a serious restriction (note that NILC has exactly the same restriction, and that it is this restriction that allows the simple diffeomorphic deformation proposed in [6]), and our method does not have it. For example, in Figure 4, *2nd column*, the Lyapunov candidates of the *1st*, *2nd*, *5th* and *6th rows* are not compatible with $\dot{\mathbf{x}} = -\mathbf{x}$. Moreover, the WSAQF Lyapunov candidates are constructed as sums of convex functions, and as such their level sets define convex regions: for any WSAQF Lyapunov candidate L, for any $\lambda > 0$, the set

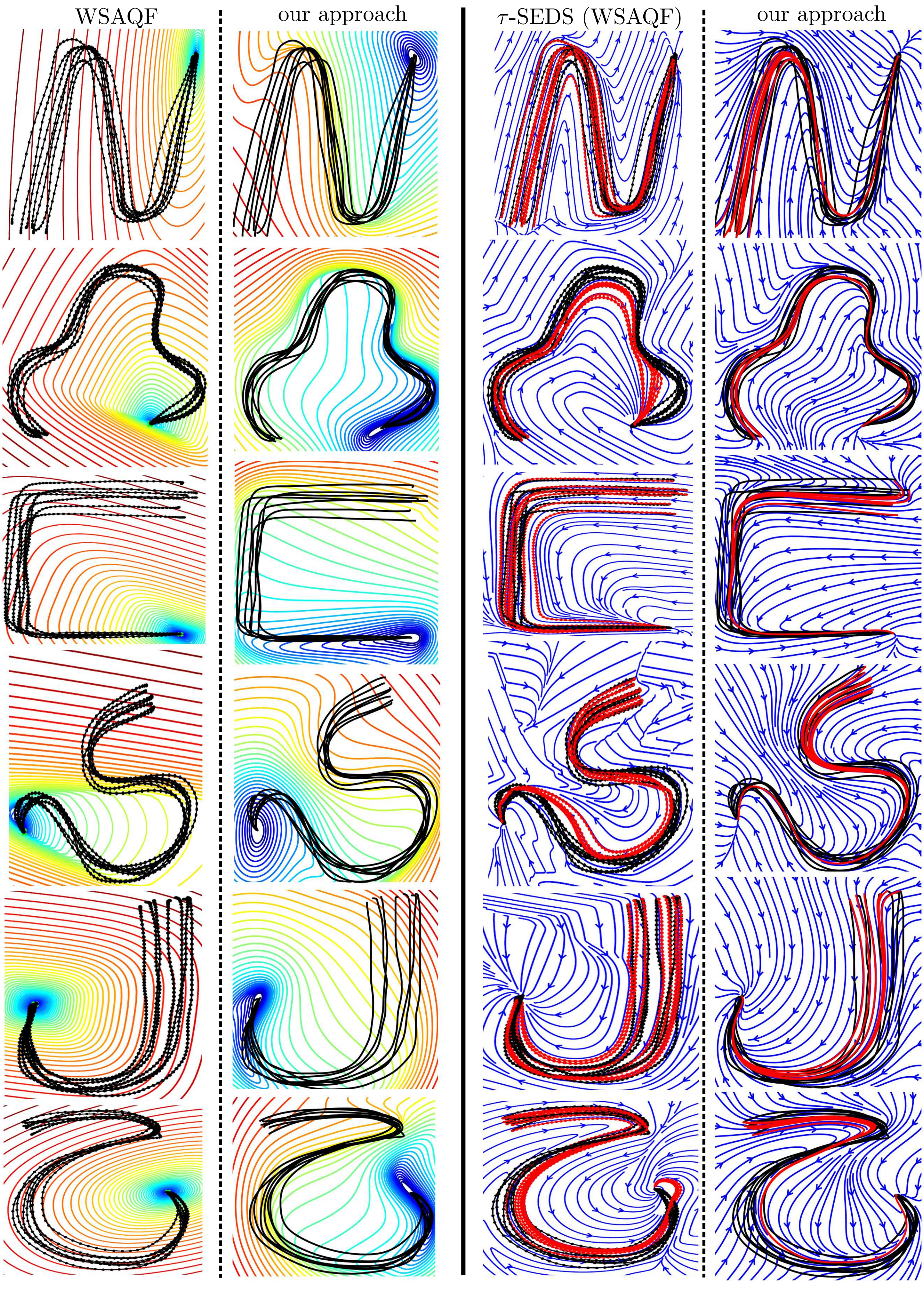

Figure 4: Demonstrations are in black, and reproduced trajectories in red. On the left: level sets of the Lyapunov candidates obtained with WSAQF (*1st column*) and with our approach (*2nd column*). On the right: streamlines of the DS produced by τ-SEDS (WSAQF) [6] (*3rd column*) and with our approach (*4th column*).

$\{\mathbf{x} \in \mathbb{R}^d \mid L(\mathbf{x}) \leq \lambda\}$ is convex. This is an even stronger restriction. In Figure 4, *2nd column*, all the Lyapunov candidates found with our method have level sets that define non-convex regions.

Our method can potentially learn more complex Lyapunov candidates because it finds them *indirectly* (via diffeomorphisms), and, instead of being based on good properties of a subset of the set of Lyapunov candidates, it is based on the stability under composition of diffeomorphisms. On top of that, the heuristic proposed in our algorithm performs a fast diffeomorphic matching in a way that is dimension-independent, which is not true for the optimization problems used in previous approaches. It potentially leads to a speed-up that can be critical in contexts in which the ability to learn and re-learn quickly is important.

5. Conclusion

In this paper, we presented a new algorithm for diffeomorphic matching and a way to use it to learn globally asymptotically stable nonlinear autonomous dynamical systems from demonstrations. While the demonstrations were 2D single motion patterns in the results we presented, our algorithm scales well to higher dimensions (because all its parameters are dimension-independent) and can be extended to handle multiple motion patterns, although in some cases topological issues may prevent the convergence of the matching. It should also be noted that we can only produce vector fields that are diffeomorphic to the DS $\dot{\mathbf{x}} = -\mathbf{x}$, which is not true for all globally asymptotically stable smooth autonomous systems. A related limitation concerns 2D spiral trajectories, which cannot be reproduced. In future work, we will try to combine our approach with existing methods to extend its possibilities. For instance, Kronander et al. [16] suggest to iteratively reshape DS by locally applying full-rank modulations such as scalings and rotations, and Khansari-Zadeh and Khatib [17] compute stable time-invariant control policies expressed as the negative gradient of a scalar potential function minus a dissipative field. These works do not guarantee the global asymptotic stability of the resulting controllers, but we see in them approaches that could be complementary to ours.

The main advantages of our method are:

1. *speed*: unlike [4] and [6], we do not need a second learning phase once the Lyapunov candidate has been found, and we do not rely on numerical optimization of parameters whose number rapidly increases with the dimensionality; instead, the simple iterative algorithm we use has a constant number of parameters, and we empirically verified its quick convergence for many difficult matching problems.
2. *simplicity*: of implementation because the algorithm is very short, but also of use thanks to the small number of parameters to adjust.

For these reasons, we believe it can be applied with ease to efficiently learn a large variety of globally asymptotically stable autonomous systems, with applications in dynamic movement primitives construction or more generally in control design. If instead of $\dot{\mathbf{x}} = -\mathbf{x}$, we start with an initial DS that has a limit cycle, our approach can be adapted to learn limit cycle systems.

Finally, being significantly faster than a state-of-the-art algorithm, our diffeomorphic matching algorithm itself might be of interest for completely different applications, such as for example image registration.

Acknowledgements

We would like to thank Aude Billard and the reviewers for their constructive comments, which helped us improve the manuscript. This work has been partly supported by the DREAM project, which received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 640891.

References

[1] S. Schaal, A. Ijspeert, A. Billard, Computational approaches to motor learning by imitation, Philosophical Transactions of the Royal Society B: Biological Sciences 358 (1431) (2003) 537–547.

[2] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, S. Schaal, Dynamical movement primitives: learning attractor models for motor behaviors, Neural computation 25 (2) (2013) 328–373.

[3] S. M. Khansari-Zadeh, A. Billard, Learning stable nonlinear dynamical systems with gaussian mixture models, IEEE Transactions on Robotics 27 (5) (2011) 943–957.

[4] S. M. Khansari-Zadeh, A. Billard, Learning control lyapunov function to ensure stability of dynamical system-based robot reaching motions, Robotics and Autonomous Systems 62 (6) (2014) 752–765.

[5] K. Neumann, A. Lemme, 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), 2013, pp. 1216–1222.

[6] K. Neumann, J. J. Steil, Learning robot motions with stable dynamical systems under diffeomorphic transformations, Robotics and Autonomous Systems 70 (2015) 1–15.

[7] S. C. Joshi, M. Miller, et al., Landmark matching via large deformation diffeomorphisms, IEEE Transactions on Image Processing 9 (8) (2000) 1357–1370.

[8] P. Dupuis, U. Grenander, M. I. Miller, Variational problems on flows of diffeomorphisms for image matching, Quarterly of applied mathematics 56 (3) (1998) 587–600.

[9] J. Glaunès, A. Qiu, M. I. Miller, L. Younes, Large deformation diffeomorphic metric curve mapping., International Journal of Computer Vision 80 (3) (2008) 317–336.

[10] M. Vaillant, J. Glaunès, Surface matching via currents, in: Information Processing in Medical Imaging, 2005, pp. 381–392.

[11] H. Guo, A. Rangarajan, S. Joshi, Diffeomorphic point matching, in: Handbook of Mathematical Models in Computer Vision, Springer, 2006, pp. 205–219.

[12] "Matchine" software by J. A. Glaunès, Copyright (C) Université Paris Descartes, http://www.mi.parisdescartes.fr/~glaunes/matchine.zip.

[13] H. K. Khalil, Nonlinear Systems, 2002.

[14] E. Rimon, D. E. Koditschek, The construction of analytic diffeomorphisms for exact robot navigation on star worlds. Transactions of the American Mathematical Society 327 (1) (1991) 71–116.

[15] "LASA Handwriting Dataset", version 2.0, Copyright (C) 2010 S. Mohammad Khansari-Zadeh, LASA Laboratory, EPFL, https://bitbucket.org/khansari/lasahandwritingdataset.

[16] K. Kronander, M. Khansari, A. Billard, Incremental motion learning with locally modulated dynamical systems, Robotics and Autonomous Systems 70 (2015) 52–62.

[17] M. Khansari-Zadeh, O. Khatib, Learning potential functions from human demonstrations with encapsulated dynamic and compliant behaviors, Autonomous Robots (2015) 1–25.