

Learning Robot Motions with Stable Dynamical Systems under Diffeomorphic Transformations

Klaus Neumann^{1,*}, Jochen J. Steil^{2,*}

*Research Institute for Cognition and Robotics (CoR-Lab)
Bielefeld University - Germany*

Abstract

Accuracy and stability have in recent studies been emphasized as the two major ingredients to learn robot motions from demonstrations with dynamical systems. Several approaches yield stable dynamical systems but are also limited to specific dynamics that can potentially result in a poor reproduction performance. The current work addresses this accuracy-stability dilemma through a new diffeomorphic transformation approach that serves as a framework generalizing the class of demonstrations that are learnable by means of provably stable dynamical systems. We apply the proposed framework to extend the application domain of the stable estimator of dynamical systems (SEDS) by generalizing the class of learnable demonstrations by means of diffeomorphic transformations τ . The resulting approach is named τ -SEDS and analyzed with rigorous theoretical investigations and robot experiments.

Keywords: SEDS, imitation learning, programming by demonstration, robotics, dynamical system, stability

1. Introduction

Nonlinear dynamical systems have been utilized recently as flexible computational basis to model motor capabilities featured by modern robots [1, 2, 3]. Point-to-point movements are an important subclass that can be modeled by autonomous dynamical systems. They are often used to provide a library of basic components called movement primitives (MP) [4, 5], which are applied very successfully to generate movements in a variety of manipulation tasks [6]. The main advantage of the dynamical systems approach over standard path planning algorithms is its inherent robustness to perturbations that result from encoding the endpoint or goal as a stable attractor, whereas the movement itself can be learned from demonstrations. Naturally, the generalization and robustness then depends on the stability properties of the underlying dynamical systems, as has been emphasized in several recent studies [7, 8, 9, 10].

The most widely known approach is the dynamic movement primitives (DMP) approach [7], which generates motions by means of a non-autonomous dynamical system. Essentially, a DMP is a linear dynamical system with nonlinear perturbation, which can be learned from demonstration to model a desired movement behavior. The stability is enforced by suppressing the nonlinear perturbation at the end of the motion, where the smooth switch from nonlinear to stable linear dynamics is controlled by a phase

variable. The phase variable can be seen as external stabilizer which in return distorts the temporal pattern of the dynamics. This leads to the inherent inability of DMP to generalize well outside the demonstrated trajectory [11].

This shortcoming motivates methods which are capable of generalizing to unseen areas in case of spatio-temporal perturbations. Such methods are time-independent and thus preserve the spatio-temporal pattern. They became of special interest by focusing on the “what to imitate” problem [12, 13].

An appealing approach that aims at ensuring robustness to temporal perturbations by learning dynamical systems from demonstrations is the stable estimator of dynamical systems (SEDS) [13]. It is based on a mixture of Gaussian functions and respects correlation across several dimensions. In [13], it is rigorously shown that SEDS is globally asymptotically stable in a fixed point attractor which marks the end of the encoded point-to-point movement. However, the proof also reveals that movements learned by SEDS are restricted to contractive dynamics corresponding to a quadratic Lyapunov function, i.e. that the distance of the current state to the attractor will decrease in time when integrating the system’s dynamics. This results in dynamical systems with globally asymptotically stable fixpoint attractor but also potentially poor reproduction performance, if the demonstrated trajectories are not contractive.

This *stability vs. accuracy dilemma* was acknowledged by Khansari-Zadeh et al. in the original work on SEDS. They remark that “the stability conditions at the basis of SEDS are sufficient conditions to ensure global asymp-

*Corresponding author

¹kneumann@cor-lab.uni-bielefeld.de

²jsteil@cor-lab.uni-bielefeld.de

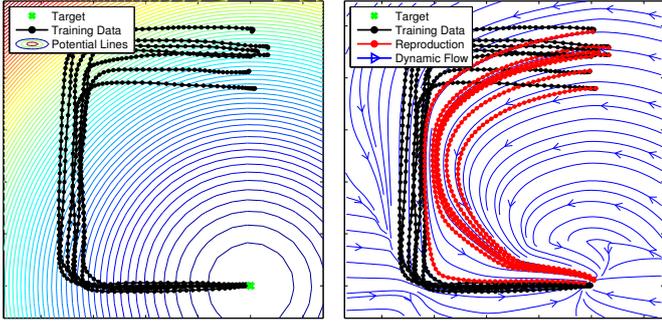


Figure 1: Learning of demonstrations with a dynamical system admitting a quadratic Lyapunov function. Contradicting quadratic Lyapunov function and demonstrations (left). The resulting dynamical system is inaccurately approximating the demonstrations when using a learner which is bound to a quadratic Lyapunov function (right).

otic stability of non-linear motions when modeled with a mixture of Gaussian functions. Although our experiments showed that a large library of robot motions can be modeled while satisfying these conditions, these global stability conditions might be too stringent to accurately model some complex motions.” ([13], p. 956).

An example illustrating this trade-off in learning from demonstrations while simultaneously satisfying a quadratic Lyapunov function is shown in Fig. 1. The sharp-C-shaped demonstrations are part of the LASA data set [14]. The equipotential lines (colored) of the quadratic Lyapunov function together with the demonstrations (black) are depicted in the left plot of the figure. The resulting flow (blue arrows) of the dynamical system, the demonstrations (black), and its reproductions (red) are depicted in Fig. 1 (right). The reproductions by means of the dynamical system are obviously stable but also inaccurate in approximating the demonstrations.

One way to overcome this problem is to separate concerns, e.g. Reinhart et al. [10] used a neural network approach to generate movements for the humanoid robot iCub. Accuracy and stability are addressed in their approach by two separately trained but superpositioned networks, which is feasible but also very complex and yields no stability guarantees.

Another approach that allows learning larger sets of demonstrations accurately is the recently developed control Lyapunov function - dynamic movements (CLF-DM) approach³ which was published in [15, 16]. CLF-DM is “in spirit identical to the control Lyapunov function - based control scheme in control engineering” ([15], p. 88) and implements less conservative stability conditions compared to SEDS but relies on an online correction signal which potentially interferes with the dynamical system. We show an example of such interference in Sec. 7.3. The construction of an appropriate control Lyapunov function (CLF)

is achieved by using an approach called weighted sum of asymmetric quadratic function (WSAQF), which learns from the set of demonstrations.

The technique of control Lyapunov functions was developed mainly by Artstein and Sontag [17, 18] who generalized Lyapunov’s theory of stability. Such control functions are used to stabilize non-linear dynamical systems (that are typically known beforehand and not learned) through corrections at runtime and interfere with the dynamical system whenever unstable behavior is detected. The detection and elimination of unstable tendencies of the dynamical system without distorting the dynamics too strongly remains difficult nevertheless.

A further approach to accurately learn a larger class of dynamics than SEDS was developed by Lemme et al. and called neurally imprinted stable vector fields (NIVF) [8]. It is based on neural networks and stability is addressed through a Lyapunov candidate that shapes the dynamical system during learning by means of quadratic programming. Lyapunov candidates are constructed by the neurally imprinted Lyapunov candidate (NILC) approach introduced recently in [19]. While this approach leads to accurate results, stability is restricted to finite regions in the workspace. Also the mathematical guarantees are obtained by an ex-post verification process which is computationally costly.

The current work therefore addresses the accuracy-stability dilemma through a new framework that generalizes the class of learnable demonstrations, provides provably stable dynamical systems and integrates the Lyapunov candidate into the learning process.

Assume that a data set $(\mathbf{x}(k), \mathbf{v}(k))$ encoding demonstrations is given, where $\mathbf{x}(k)$ refers to the position and $\mathbf{v}(k)$ to the velocity of a robot end-effector. As a first step, a so-called Lyapunov candidate $L \in \mathcal{L}$ with $L : \Omega \rightarrow \mathbb{R}$ that is consistent with the demonstrations (i.e. $\nabla L(\mathbf{x}(k)) \cdot \mathbf{v}(k) < 0 : \forall k$) is learned. Second, diffeomorphic transformations $\tau : \mathcal{L} \times \Omega \rightarrow \tilde{\Omega}$ are defined that transform those candidates from the original space to a new space in which they appear as a fixed and simple function.

These transformations (parameterized with the learned Lyapunov candidate L) are then used to map the demonstrations from Ω to $\tilde{\Omega}$ where they are consistent with the underlying fixed Lyapunov function of the learner - in the special case of SEDS, a quadratic function. That is, in the new space a provably globally stable dynamical system (i.e. $\Omega = \tilde{\Omega} = \mathbb{R}^d$) can be learned with respect to the transformed data, which is then back-transformed into the original space with the inverse mapping $\tau_L^{-1} : \tilde{\Omega} \rightarrow \Omega$ which exists because of the diffeomorphic properties of τ_L . It is then shown that in the original space, the Lyapunov candidate indeed can be used to prove stability of the transformed dynamical system, which accurately models the demonstrations and resolves the dilemma.

We evaluate the new approach - named τ -SEDS - in detail, provide rigorous theoretical investigations, and experiments that substantiate the effectiveness and applica-

³This approach was originally called SEDS-II [15].

bility of the proposed theoretical framework to enhance the class of learnable stable dynamical systems to generate robot motions.

2. Programming Autonomous Dynamical Systems by Demonstration

Assume that a data set $D = (\mathbf{x}^i(k), \mathbf{v}^i(k))$ with the indices $i = 1 \dots N_{\text{traj}}$ and $k = 1 \dots N^i$ consisting of N_{traj} demonstrations is given. $N = \sum_i N^i$ denotes the number of samples in the data set. The demonstrations considered in this paper encode point-to-point motions that share the same end point, i.e. $\mathbf{x}^i(N^i) = \mathbf{x}^j(N^j) = \mathbf{x}^* : \forall i, j = 1 \dots N_{\text{traj}}$ and $\mathbf{v}^i(N^i) = 0 : \forall i = 1 \dots N_{\text{traj}}$. These demonstration could be a sequence of the robot's joint angles or the position of the arm's end-effector possibly obtained by kinesthetic teaching.

We assume that such demonstrations can be modeled by autonomous dynamical systems which can be learned by using a set of parameters that are optimized by means of the set of demonstrations.

$$\dot{\mathbf{x}}(t) = \mathbf{y}(\mathbf{x}(t)) : \mathbf{x} \in \Omega, \quad (1)$$

where $\Omega \subseteq \mathbb{R}^d$ might be the joint or workspace of the robot. It is of particular interest that $\mathbf{y}(\mathbf{x}) : \Omega \rightarrow \Omega$ has a single asymptotically stable point attractor $\mathbf{x}^* = \mathbf{v}(\mathbf{x}^*) = 0$ in Ω , besides that \mathbf{y} is nonlinear, continuous, and continuously differentiable. The limit of each trajectory has to satisfy:

$$\lim_{t \rightarrow \infty} \mathbf{x}(t) = \mathbf{x}^* : \forall \mathbf{x} \in \Omega. \quad (2)$$

New trajectories can be obtained by numerical integration of Eq. (1) when starting from a given initial point in Ω . They are called reproductions and denoted by $\hat{\mathbf{x}}^i(\cdot)$ if they start from the demonstrations' initial points $\mathbf{x}^i(0)$.

In order to analyze the stability of a dynamical system, we recall the conditions of asymptotic stability found by Lyapunov:

Theorem 1. *A dynamical system is **locally asymptotically stable** at fixed-point $\mathbf{x}^* \in \Omega$ in the positive invariant neighborhood $\Omega \subset \mathbb{R}^d$ of \mathbf{x}^* , if and only if there exists a continuous and continuously differentiable function $L : \Omega \rightarrow \mathbb{R}$ which satisfies the following conditions:*

$$\begin{aligned} \text{(i)} \quad L(\mathbf{x}^*) &= 0 & \text{(ii)} \quad L(\mathbf{x}) > 0 : \forall \mathbf{x} \in \Omega \setminus \mathbf{x}^* \\ \text{(iii)} \quad \dot{L}(\mathbf{x}^*) &= 0 & \text{(iv)} \quad \dot{L}(\mathbf{x}) < 0 : \forall \mathbf{x} \in \Omega \setminus \mathbf{x}^*. \end{aligned} \quad (3)$$

*The dynamical system is **globally asymptotically stable** at fixed-point \mathbf{x}^* if $\Omega = \mathbb{R}^d$ and L is radially unbounded, i.e. $\|\mathbf{x}\| \rightarrow \infty \Rightarrow L(\mathbf{x}) \rightarrow \infty$. The function $L : \Omega \rightarrow \mathbb{R}$ is called **Lyapunov function**.*

It is usually easier to search for the existence of a Lyapunov function than to proof asymptotic stability of a dynamical system directly. Typically, previously defined Lyapunov candidates are used as a starting point for stability verification and conditions (i)-(iv) of theorem 1 are verified

in a stepwise fashion to promote the candidate to become an actual Lyapunov function. We thus first define what kind of Lyapunov candidates are in principle applicable for investigation.

Definition 1. A **Lyapunov candidate** is a continuous and continuously differentiable function $L : \Omega \rightarrow \mathbb{R}$ that satisfies the following conditions

$$\begin{aligned} \text{(i)} \quad L(\mathbf{x}^*) &= 0 & \text{(ii)} \quad L(\mathbf{x}) > 0 : \forall \mathbf{x} \in \Omega \setminus \mathbf{x}^* \\ \text{(iii)} \quad \nabla L(\mathbf{x}^*) &= 0 & \text{(iv)} \quad \nabla L(\mathbf{x}) \neq 0 : \forall \mathbf{x} \in \Omega \setminus \mathbf{x}^*, \end{aligned} \quad (4)$$

where $\mathbf{x}^* \in \Omega$ is the asymptotically stable fixed-point attractor and Ω is a positive invariant neighborhood of \mathbf{x}^* . L is a globally defined Lyapunov candidate if $\Omega = \mathbb{R}^d$ and L is radially unbounded in addition to the previous conditions, i.e. $\|\mathbf{x}\| \rightarrow \infty \Rightarrow L(\mathbf{x}) \rightarrow \infty$.

We use the term Lyapunov candidate whenever the function is used to enforce asymptotic stability of a dynamical system during learning and control Lyapunov function if the dynamical system is stabilized during runtime. In the following, we will learn such candidate functions and evaluate their quality. To this aim we define what it means that a Lyapunov candidate contradicts a given demonstration or reference trajectory.

Definition 2. A Lyapunov candidate $L : \Omega \rightarrow \mathbb{R}$ **violates/contradicts** a dynamical system $\mathbf{v} : \Omega \rightarrow \Omega$ or a demonstration $(\mathbf{x}^i(k), \mathbf{v}^i(k)) : k = 1 \dots N^i$, if and only if

$$\begin{aligned} \exists \mathbf{x} \in \Omega : \nabla^T L(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) > 0 \quad \text{or} \\ \exists k : 1 \leq k \leq N^i : \nabla^T L(\mathbf{x}^i(k)) \cdot \mathbf{v}^i(k) > 0. \end{aligned} \quad (5)$$

The dynamical system or the given demonstration is said to be **consistent with** or **satisfying/fulfilling** the Lyapunov candidate if and only if there is no violation.

3. Related Work

Several different approaches for movement generation and learning of autonomous dynamical systems have been introduced so far. This section introduces the most important developments among those methods as related work and embeds them in the previously defined formalism necessary to achieve stable movement generation.

3.1. Stable Estimator of Dynamical Systems (SEDS)

The stable estimator of dynamical systems (SEDS) [13] is a advanced version of the binary merging algorithm [9] and learns a dynamical system by means of a Gaussian mixture model

$$\dot{\mathbf{x}} = \sum_{k=1}^K \frac{\mathcal{P}(k)\mathcal{P}(\mathbf{x}|k)}{\sum_i \mathcal{P}(k)\mathcal{P}(\mathbf{x}|i)} (\mu_{\dot{\mathbf{x}}}^k + \Sigma_{\dot{\mathbf{x}}\mathbf{x}}^k (\Sigma_{\mathbf{x}}^k)^{-1} (\mathbf{x} - \mu_{\mathbf{x}}^k)) \quad (6)$$

where $\mathcal{P}(k)$, μ^k , and Σ^k yield the prior probability, the mean, and the covariance matrix of the K Gaussian functions, respectively. Note that this model can be expressed

as a space varying sum of linear dynamical systems according to the definition of the matrix $A^k = \Sigma_{\mathbf{x}\mathbf{x}}^k (\Sigma_{\mathbf{x}}^k)^{-1}$, the bias $b^k = \mu_{\mathbf{x}}^k - A^k \mu_{\mathbf{x}^*}$, and the nonlinear weighting terms $h(k) = \frac{\mathcal{P}^k(k)\mathcal{P}(\mathbf{x}|k)}{\sum_i \mathcal{P}^k(k)\mathcal{P}(\mathbf{x}|i)}$. The reformulation according to this definition leads to

$$\dot{\mathbf{x}}(t) = \mathbf{y}(\mathbf{x}(t)) = \sum_{k=1}^K h^k(\mathbf{x}(t)) (A^k \mathbf{x}(t) + \mathbf{b}^k) . \quad (7)$$

Learning can be done by minimization of different objective functions by a non-linear program subject to a set of non-linear constraints. A possible objective function can be the mean square error functional

$$\min \sum_{ik}^N \|\mathbf{v}^i(k) - \mathbf{y}(\mathbf{x}^i(k))\|^2 , \quad (8)$$

which is minimized in the parameters of the Gaussian mixture model and at the same time subject to the following non-linear constraints [13]

$$\begin{aligned} (i) \quad & \mathbf{b}^k = -A^k \mathbf{x}^* \\ (ii) \quad & A^k + A^{kT} \prec 0 : \forall k = 1, \dots, K , \end{aligned} \quad (9)$$

where $\prec 0$ denotes the negative definiteness of a matrix. Note, that it is also necessary to add the constraints for the requirements on the covariance matrices Σ^k and priors \mathcal{P} to the non-linear program⁴. It is shown that these constraints are sufficient conditions for the learned dynamical system to be globally asymptotically stable.

The stability analysis in the original contribution considers a quadratic Lyapunov candidate

$$L(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^T (\mathbf{x} - \mathbf{x}^*) : \forall \mathbf{x} \in \mathbb{R}^d , \quad (10)$$

which is used for stability analysis. In detail, the theorem states that this scalar function is indeed a Lyapunov function of the autonomous dynamical system defined by SEDS

$$\begin{aligned} \dot{L}(\mathbf{x}) &= \frac{d}{dt} L(\mathbf{x}(t)) \\ &= \nabla L(\mathbf{x}(t)) \cdot \frac{d}{dt} \mathbf{x}(t) = \nabla L(\mathbf{x}(t)) \cdot \dot{\mathbf{x}}(t) \\ &= \sum_{k=1}^K \underbrace{h^k(\mathbf{x})}_{>0} \underbrace{(\mathbf{x} - \mathbf{x}^*)^T A^k (\mathbf{x} - \mathbf{x}^*)}_{<0} \\ &< 0 : \forall \mathbf{x} \in \mathbb{R}^d \setminus \mathbf{x}^* . \end{aligned} \quad (11)$$

This directly suggests that the dynamical system learned by SEDS are restricted to contractive behavior irrespective of the used demonstrations, i.e. that the distance $\|\hat{\mathbf{x}}(t) - \mathbf{x}^*\|$ of the dynamic state $\hat{\mathbf{x}}(t)$ to the attractor \mathbf{x}^* decreases in time when integrating $\dot{\hat{\mathbf{x}}}(t) = \mathbf{y}(\hat{\mathbf{x}}(t))$. This

means that if the Lyapunov function is in contradiction to the demonstrations, learning the data by means of this method will result in a dynamical system with asymptotically stable fixpoint attractor but also a potentially poor reproduction performance.

3.2. Control Lyapunov Function-Based Dynamics Movements (CLF-DM)

In order to overcome the limitations of SEDS, a different approach called control Lyapunov function-based dynamics movements (CLF-DM) that is based on control Lyapunov functions is proposed in [16]. This approach is separated into three steps in order to obtain a stable dynamical system. The first step is to learn an appropriate Lyapunov candidate that is suited towards the data. The second step is to learn the demonstrations with a regression method. Typically, Gaussian process regression, Gaussian mixture regression, locally weighted projection regression, or support vector regression techniques are used. However, this dynamical system is very likely unstable and will be stabilized in the third step. The idea is to stabilize the trajectory by means of online corrections during runtime that are performed whenever the Lyapunov candidate is violated by the motion. The resulting dynamics are given by

$$\dot{\mathbf{x}}(t) = \mathbf{y}(\mathbf{x}(t)) = \hat{\mathbf{y}}(\mathbf{x}(t)) + \mathbf{u}(\mathbf{x}(t)) , \quad (12)$$

where $\hat{\mathbf{y}}$ is the unstable system obtained by the regression method and \mathbf{u} is the correction signal applied during runtime. This correction signal \mathbf{u} needs an appropriate control Lyapunov function that is consistent with the data in order to control the signal accurately. The signal itself is obtained by solving a constrained optimization program analytically and strongly related to Sontag's universal formula [18]

$$\mathbf{u}(\mathbf{x}) = \begin{cases} -\hat{\mathbf{y}}(\mathbf{x}^*) : \mathbf{x} = \mathbf{x}^* \\ \mathbf{0} : \nabla L(\mathbf{x})^T \hat{\mathbf{y}}(\mathbf{x}) \leq -\rho(\mathbf{x}) \\ -\frac{\nabla L(\mathbf{x})^T \hat{\mathbf{y}}(\mathbf{x}) + \rho(\mathbf{x})}{\nabla L(\mathbf{x})^T \nabla L(\mathbf{x})} : \text{otherwise} \end{cases} . \quad (13)$$

The parameterized function $\rho(\mathbf{x}) = \rho_0 (1 - e^{-\kappa_0 \|\mathbf{x}\|})$ defines the threshold for activation of the online corrections while the scalar function L defines the actual control Lyapunov function. One method to obtain a suitable CLF from demonstrations is the so-called weighted sum of asymmetric quadratic functions (WSAQF) [16] which was specifically developed for the CLF-DM approach and serves as a potential control Lyapunov function. A detailed description of the approach is given in Sec. 5.1.

The CLF-DM approach allows learning a larger set of robot motions compared to the SEDS approach. However, the learning of the Lyapunov candidate function and the dynamical system are two separate processes, which is different from the unified learning of the SEDS approach. The online corrections can thus lead to a performance that can be outperformed by SEDS in specific cases.

⁴See the paper by Khansari-Zadeh et al. [13] for further details.

3.3. Neurally Imprinted Stable Vector Fields (NIVF)

Another successful approach to represent robotic movements by means of autonomous dynamical systems is based on neural networks and called the neurally imprinted vector fields approach [8]. It features efficient supervised learning and incorporates stability constraints via quadratic programming (QP). The constraints are derived from a parameterized or learned Lyapunov function which enforces local stability.

The approach considers feed-forward neural networks that comprise three different layers of neurons: $\mathbf{x} \in \mathbb{R}^I$ denotes the input, $\mathbf{h} \in \mathbb{R}^R$ the hidden, and $\mathbf{y} \in \mathbb{R}^I$ the output neurons. These neurons are connected via input matrix $W^{\text{inp}} \in \mathbb{R}^{R \times I}$, which remains fixed after random initialization and are not subject to supervised learning. The read-out matrix given by $W^{\text{out}} \in \mathbb{R}^{I \times R}$ which is subject to supervised learning. For input \mathbf{x} the output of the i th read-out neuron is thus given by

$$y_i(\mathbf{x}) = \sum_{j=1}^R W_{ij}^{\text{out}} f\left(\sum_{n=1}^I W_{jn}^{\text{inp}} x_n + b_j\right), \quad (14)$$

where the biases b_j parameterize the component-wise Fermi function $f(x) = \frac{1}{1+e^{-x}}$ of the j th neuron in the hidden layer.

It is assumed that a Lyapunov candidate L is given. In order to obtain a learning algorithm for W^{out} that also respects condition (iv) of Lyapunov's theorem, this condition is analyzed by taking the time derivative of L :

$$\begin{aligned} \dot{L}(\mathbf{x}) &= (\nabla_{\mathbf{x}} L(\mathbf{x}))^T \cdot \frac{d}{dt} \mathbf{x} = (\nabla_{\mathbf{x}} L(\mathbf{x}))^T \cdot \dot{\mathbf{v}} \\ &= \sum_{i=1}^I (\nabla_{\mathbf{x}} L(\mathbf{x}))_i \sum_{k=1}^R W_{ik}^{\text{out}} f_j(W^{\text{inp}} x + b) < 0. \end{aligned} \quad (15)$$

Interestingly, \dot{L} is linear in the output parameters W^{out} and irrespective of the form of the Lyapunov function L . For a given point $\mathbf{u} \in \Omega$, Eq. (15) defines a linear constraint on the read-out parameters W^{out} , which is implemented by solving the quadratic program with weight regularization [20]:

$$\begin{aligned} W^{\text{out}} &= \arg \min_W (\|W \cdot H(X) - V\|^2 + \varepsilon \|W\|^2) \\ &\text{subject to: } \dot{L}(U) < 0, \end{aligned} \quad (16)$$

where the matrix $H(X) = (\mathbf{h}(\mathbf{x}(1)), \dots, \mathbf{h}(\mathbf{x}(N_{\text{tr}})))$ collects the hidden layer states obtained from a given data set $D = (X, V) = (\mathbf{x}^i(k), \mathbf{v}^i(k))$ for inputs X and the corresponding output vectors V and where ε is a regularization parameter. It is shown in [20] that a well-chosen sampling of points U is sufficient to generalize the incorporated discrete constraints to continuous regions in a reliable way. The independence of Eq. (15) from the specific form of L motivates the use of methods to learn highly flexible Lyapunov candidates from data. The neurally imprinted Lyapunov candidate (NILC) [19] is such a method that

enables the NIVF approach to generate robust and flexible movements for robotics. Details of the approach are stated in Sec. 5.2.

4. Learning Stable Dynamics under Diffeomorphic Transformations

This section describes how to link a Lyapunov candidate with respect to given demonstrations in one space Ω and the learning of a stable dynamical system with quadratic Lyapunov function with respect to transformed data in a second space $\tilde{\Omega}$ by means of a diffeomorphism τ . The latter is described on an abstract level and by an illustrative example. Also the main algorithmic steps are introduced. The procedure undergoes a rigorous stability analysis that substantiates the underlying principles.

4.1. Overview

Assume that a Lyapunov candidate $L : \Omega \rightarrow \mathbb{R}$ with $L \in \mathcal{L}$, which is consistent with the demonstrations in D , is given or can be constructed automatically. The main goal is to find a mapping $\tau : \mathcal{L} \times \Omega \rightarrow \tilde{\Omega}$ that transforms the Lyapunov function candidate L into a fixed and simple function $\tilde{L} : \tilde{\Omega} \rightarrow \mathbb{R}$ in the new space $\tilde{\Omega}$ such that the parameterized mapping $\tau_L : \Omega \rightarrow \tilde{\Omega}$ is a diffeomorphism. The transformation is defined according to the following

Definition 3. A **diffeomorphic candidate transformation** $\tau : \mathcal{L} \times \Omega \rightarrow \tilde{\Omega}$ with $(L, \mathbf{x}) \mapsto \tilde{\mathbf{x}}$ transforms all Lyapunov candidates $L : \Omega \rightarrow \mathbb{R}$ with $L \in \mathcal{L}$ to a fixed function $\tilde{L} : \tilde{\Omega} \rightarrow \mathbb{R}$ such that the parameterized mapping $\tau_L : \Omega \rightarrow \tilde{\Omega}$ is a diffeomorphism, i.e. $\tau_L : \Omega \rightarrow \tilde{\Omega}$ is bijective, continuous, continuously differentiable, and the inverse mapping $\tau_L^{-1} : \tilde{\Omega} \rightarrow \Omega$ is also continuous and continuously differentiable. We say τ **corresponds to** \mathcal{L} .

The main example and standard case used in this work is to target a quadratic function $\tilde{L}(\tilde{\mathbf{x}}) = L(\tau_L^{-1}(\tilde{\mathbf{x}})) = \tilde{\mathbf{x}}^2$ after transformation.

The idea is then to use τ_L in order to transform the data set D into the new space. The obtained data set

$$\tilde{D} = (\tilde{\mathbf{x}}^i(k), \tilde{\mathbf{v}}^i(k)) = (\tau_L(\mathbf{x}^i(k)), J_{\tau}^T(\mathbf{x}^i(k)) \cdot \mathbf{v}^i(k)) \quad (17)$$

is consistent with this Lyapunov candidate \tilde{L} if the initial data D is consistent with the Lyapunov function candidate L . The term $(J_{\tau}(\mathbf{x}^i(k)))_{mn} = \frac{\partial}{\partial x_m} \tau_n(\mathbf{x}^i(k))$ denotes the Jacobian matrix for τ_L at point $\mathbf{x}^i(k)$.

Also assume that a learner is given which is able to guarantee asymptotic stability by means of a quadratic Lyapunov function $\tilde{L}(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}}^2$ (e.g. the SEDS approach). The dynamical system $\tilde{\mathbf{y}} : \tilde{\Omega} \rightarrow \tilde{\Omega}$ trained with the data \tilde{D} in $\tilde{\Omega}$ is then expected to be accurate. The inverse of the diffeomorphism $\tau_L^{-1} : \tilde{\Omega} \rightarrow \Omega$ is used to map the dynamical system back to the original space. The back transformation $\mathbf{y} : \Omega \rightarrow \Omega$ of $\tilde{\mathbf{y}}$ is formally given by

$$\mathbf{y}(\mathbf{x}) := J_{\tau}^{-T}(\tau_L(\mathbf{x})) \cdot \tilde{\mathbf{y}}(\tau_L(\mathbf{x})), \quad (18)$$

Algorithm 1 Diffeomorphic Transformation Approach

Require: Data set $D = (\mathbf{x}^i(k), \mathbf{v}^i(k)) : i = 1 \dots N_{\text{traj}}, k = 1 \dots N^i$ is given

- 1) Construct Lyapunov candidate $L : \Omega \rightarrow \mathbb{R}$ that is consistent with data D
 - 2) Define a diffeomorphism $\tau : \mathcal{L} \times \Omega \rightarrow \tilde{\Omega}$ where \tilde{L} takes a quadratic form
 - 3) Transform D to $\tilde{D} = (\tilde{\mathbf{x}}^i(k), \tilde{\mathbf{v}}^i(k)) = (\tau_L(\mathbf{x}^i(k)), J_\tau^T(\mathbf{x}^i(k)) \cdot \mathbf{v}^i(k))$
 - 4) Learn a dynamical system $\tilde{\mathbf{y}} : \tilde{\Omega} \rightarrow \tilde{\Omega}$ of data \tilde{D} in $\tilde{\Omega}$ with stability according to \tilde{L}
 - 5) Apply the back transformation $\mathbf{y}(\mathbf{x}) := J_\tau^{-T}(\tau_L(\mathbf{x})) \cdot \tilde{\mathbf{y}}(\tau_L(\mathbf{x}))$ in Ω to obtain a stable dynamical system
-

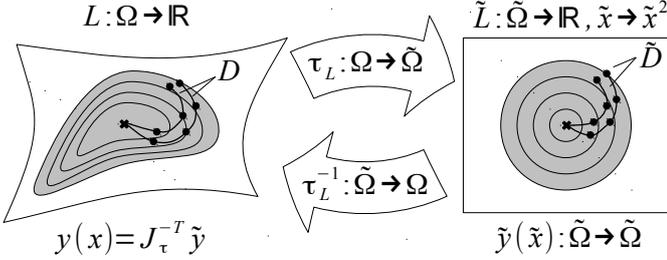


Figure 2: Schematic illustration of the proposed transformation approach. The left part of the figure shows the original space Ω , the demonstrations D , and the complex Lyapunov candidate L . The right side visualizes the transformed space $\tilde{\Omega}$ equipped with a quadratic Lyapunov function \tilde{L} and the corresponding data \tilde{D} . The transformation τ between those space is visualized by the arrows in the center part of the plot.

where $(J_\tau^{-T}(\tau(\mathbf{x})))_{ij} = \frac{\partial}{\partial \tilde{x}_i} \tau_{L_j}^{-T}(\tilde{\mathbf{x}})$ denotes the transpose of the inverse Jacobian matrix for τ_L at point \mathbf{x} . This transformation behavior is rigorously investigated in the following sections regarding the stability analysis of the underlying dynamical systems. This procedure is summarized in Alg. 1 and schematically illustrated in Fig. 2.

4.2. The Diffeomorphic Transformation Approach: A Simple Illustrative Example

Fig. 3 illustrates the intermediate steps of the diffeomorphic candidate transformation and learning of the SEDS approach shown in Alg.1. The movement obviously violates a quadratic Lyapunov candidate (as shown in Fig. 1) and is thus well suited for the transformation approach.

First, we manually construct an elliptic Lyapunov candidate that is more or less consistent with the training data D (step 1). It is directly clear that an elliptic Lyapunov candidate is too restricted for learning complex motions, but it is good enough to serve as an example. We define the Lyapunov candidate as

$$L(\mathbf{x}) = \mathbf{x}^T P \mathbf{x} , \quad (19)$$

with the diagonal matrix $P = \text{diag}(1,5)$. The set of possible candidates \mathcal{L} is given by

$$\mathcal{L} = \{ \mathbf{x}^T P \mathbf{x} : P \text{ diag. matrix and pos. def.} \} , \quad (20)$$

The visualization of this scalar function that serves as Lyapunov candidate is shown in Fig. 3 (second). Note, that this function still violates the training data but relaxes the

violation to a satisfactory degree. A diffeomorphic candidate transformation τ that corresponds to \mathcal{L} is given (step 2) by the following mapping

$$\tau_L(\mathbf{x}) = \sqrt{P} , \quad (21)$$

which is the component-wise square root of the matrix P and particularly constructed for the elliptic candidate functions defined by different diagonal matrices P . It is important to understand that this function τ maps any elliptic Lyapunov candidate in \mathcal{L} onto a quadratic function.

$$\begin{aligned} \tilde{L}(\tilde{\mathbf{x}}) &= L(\tau^{-1}(\tilde{\mathbf{x}})) = L(\sqrt{P}^{-1} \tilde{\mathbf{x}}) \\ &= (\sqrt{P}^{-1} \tilde{\mathbf{x}})^T P \sqrt{P}^{-1} \tilde{\mathbf{x}} \\ &= \tilde{\mathbf{x}}^T \sqrt{P}^{-T} P \sqrt{P}^{-1} \tilde{\mathbf{x}} \\ &= \tilde{\mathbf{x}}^T \sqrt{P}^{-1} P \sqrt{P}^{-1} \tilde{\mathbf{x}} \\ &= \tilde{\mathbf{x}}^T \tilde{\mathbf{x}} = \|\tilde{\mathbf{x}}\|^2 . \end{aligned} \quad (22)$$

The respective Jacobian matrix is analytically given and calculated as

$$J_\tau(\mathbf{x}) = \sqrt{P} , \quad (23)$$

where we used the symmetric definition of the P matrix: $\sqrt{P}^T = \sqrt{P}$.

The training data set D is then prepared for learning by transforming the data set into \tilde{D} that is defined in the transformed space (step 3) which is consistent with a quadratic Lyapunov candidate $\tilde{L}(\tilde{\mathbf{x}})$. The result of the data transformation and the Lyapunov candidate is illustrated in Fig. 3 (second). We then apply a learning approach (here: SEDS) to obtain $\tilde{\mathbf{y}}$ (step 4) which is stable according to a quadratic Lyapunov function \tilde{L} in $\tilde{\Omega}$ after learning the data \tilde{D} . The result of the learning is depicted by the dynamic flow after learning the transformed demonstrations \tilde{D} in Fig. 3 (third).

Finally, the inverse transformation $\tau_L^{-1} = \sqrt{P}^{-1}$ is used to obtain the dynamics \mathbf{y} for the original data D in the original space Ω (step 5). Eq. (18) was used for back transformation. It is illustrated that the transformed data set is still violating the quadratic function, however, less strongly such that more accurate modeling of the original demonstrations is enabled, see Fig. 3 (fourth). Note that the resulting dynamical systems and their generalization can potentially differ in many aspects. It is important to understand that the origin of such effects are caused by

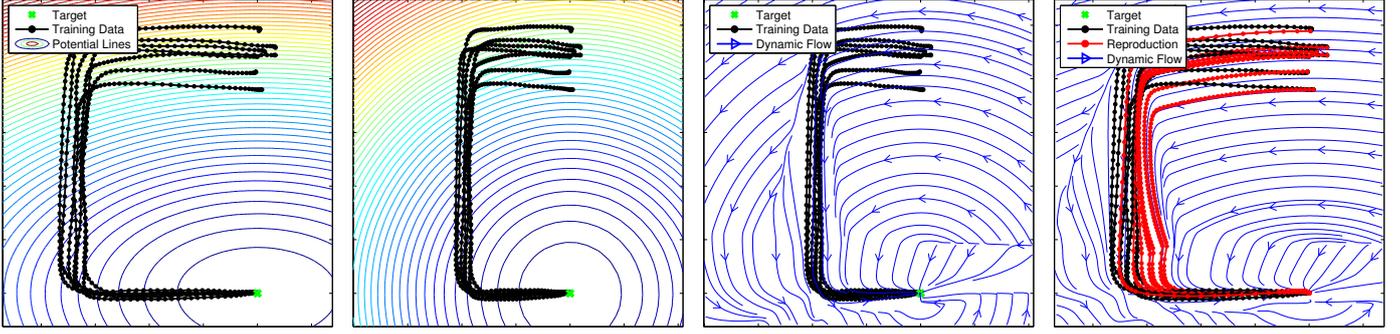


Figure 3: Demonstrations D and the respective Lyapunov candidate L in Ω (first). Transformed Lyapunov function \tilde{L} and transformed demonstrations \tilde{D} in $\tilde{\Omega}$ (second). The dynamical system $\tilde{\mathbf{y}}$ learned by SEDS using the data set \tilde{D} which fulfills a quadratic Lyapunov function in the transformed space $\tilde{\Omega}$ (third). The result \mathbf{y} in Ω after applying the inverse transformation τ_L^{-1} of $\tilde{\mathbf{y}}$ (fourth).

many different features of the algorithm, e.g. the selection of the Lyapunov candidate or the randomness of the SEDS algorithm. The exact definition of the term “generalization capability” of the dynamical system and its measurement remains difficult. Systematic approaches to answer this question were rigorously discussed in [21].

4.3. General Stability Analysis

The main question raised is regarding the stability of the transformed system \mathbf{y} in the original space Ω . It is also of fundamental interest how the generalization capability of the learner in $\tilde{\Omega}$ transfers into the original space Ω .

The following proposition indicates the necessary conditions for implementation.

Proposition 1. Let $D = (\mathbf{x}^i(k), \mathbf{v}^i(k))$ be a data set with $i = 1 \dots N_{\text{traj}}$ and $k = 1 \dots N^i$ consisting of N_{traj} demonstrations and $L : \Omega \rightarrow \mathbb{R}$ be a Lyapunov candidate from the set \mathcal{L} . Let $\tau : \mathcal{L} \times \Omega \rightarrow \tilde{\Omega}$ be a diffeomorphic candidate transformation that corresponds to \mathcal{L} .

Then, it holds for all $L \in \mathcal{L}$ that the dynamical system $\mathbf{y} : \Omega \rightarrow \Omega$ with $\mathbf{y}(\mathbf{x}) := J_\tau^{-T}(\tau_L(\mathbf{x})) \cdot \tilde{\mathbf{y}}(\tau_L(\mathbf{x}))$ is asymptotically stable at target \mathbf{x}^* with Lyapunov function L if and only if the dynamical system $\tilde{\mathbf{y}} : \tilde{\Omega} \rightarrow \tilde{\Omega}$ is asymptotically stable at target $\tilde{\mathbf{x}}^*$ with $\tau_L(\mathbf{x}^*) = \tilde{\mathbf{x}}^*$ and Lyapunov function \tilde{L} .

PROOF. We first derive the transformation properties for the Lyapunov candidate. Note, that the dependence on the Lyapunov candidate L will be omitted in the following for notational simplicity, i.e. $\tau = \tau_L$. Scalar functions such as Lyapunov candidates show the following forward and backward transformation behavior

$$L(\mathbf{x}) = \tilde{L}(\tau(\mathbf{x})) \text{ and } \tilde{L}(\tilde{\mathbf{x}}) = L(\tau^{-1}(\tilde{\mathbf{x}})) , \quad (24)$$

while these equations hold for $\mathbf{x} \in \Omega$ and $\tilde{\mathbf{x}} \in \tilde{\Omega}$. This transformation behavior is important for the investigation of the differential information of the Lyapunov candidates

in the different spaces. The gradient of the Lyapunov candidate thus transforms according to

$$\nabla L(\mathbf{x}) = J_\tau(\mathbf{x}) \cdot \tilde{\nabla} \tilde{L}(\tilde{\mathbf{x}}) , \quad (25)$$

where $(J_\tau(\mathbf{x}))_{ij} = \frac{\partial}{\partial x_i} \tau_j(\mathbf{x})$ is the Jacobian matrix for the diffeomorphism τ at point \mathbf{x} . A vector field $\mathbf{y}(\mathbf{x})$ can also be represented in both spaces. The transformation behavior of the dynamical system is the following

$$\begin{aligned} \mathbf{y}(\mathbf{x}) &= \sum_{k=1}^d \sum_{j=1}^d (J_\tau^{-T}(\tau(\mathbf{x})))_{ji} \cdot \tilde{y}_j(\tau(\mathbf{x})) \\ &= J_\tau^{-T}(\tilde{\mathbf{x}}) \cdot \tilde{\mathbf{y}}(\tilde{\mathbf{x}}) . \end{aligned} \quad (26)$$

where $(J_\tau^{-T}(\tau(\mathbf{x})))_{ij}$ is the transpose of the inverse Jacobian matrix $\frac{\partial}{\partial \tilde{x}_i} \tau_j^{-1}(\tilde{\mathbf{x}})$ for the function τ at point \mathbf{x} . These identities hold because of the diffeomorphic properties of τ . The mathematical justification is given by the inverse function theorem, which states that the inverse of the Jacobian matrix equals the Jacobian of the inverse function.

The following equations show that L is an actual Lyapunov function for the dynamical system $\mathbf{y}(\mathbf{x})$. Per definition, L satisfies (i) and (ii) of Lyapunov’s conditions for asymptotic stability stated in theorem 1. We thus focus on condition (iii)

$$\begin{aligned} \dot{L}(\mathbf{x}^*) &= (\mathbf{y}(\mathbf{x}^*))^T \nabla L(\mathbf{x}^*) \\ &= (J_\tau^{-T}(\tilde{\mathbf{x}}^*) \cdot \underbrace{\tilde{\mathbf{y}}(\tilde{\mathbf{x}}^*)^T}_{=0 \text{ FP}} \cdot \underbrace{\nabla L(\mathbf{x}^*)}_{=0 \text{ (iii)}}) = 0 , \end{aligned} \quad (27)$$

which is also satisfied. The main requirement for the proof of the proposition is that condition (iv) is fulfilled. It states that the dynamical system, which is stable with a quadratic Lyapunov function in the transformation space, becomes asymptotically stable according to the previously defined Lyapunov candidate function L in the original space Ω after back transformation. The Lyapunov candidate L

thus becomes a Lyapunov function.

$$\begin{aligned}
\dot{L}(\mathbf{x}) &= (\mathbf{y}(\mathbf{x}))^T \cdot \nabla L(\mathbf{x}) \\
&= (J_\tau^{-T}(\tilde{\mathbf{x}}) \cdot \tilde{\mathbf{y}}(\tilde{\mathbf{x}}))^T \cdot J_\tau(\mathbf{x}) \cdot \tilde{\nabla} \tilde{L}(\tilde{\mathbf{x}}) \\
&= \tilde{\mathbf{y}}(\tilde{\mathbf{x}})^T \cdot J_\tau^{-1}(\tilde{\mathbf{x}}) \cdot J_\tau(\mathbf{x}) \cdot \tilde{\nabla} \tilde{L}(\tilde{\mathbf{x}}) \\
&= \tilde{\mathbf{y}}(\tilde{\mathbf{x}})^T \cdot \tilde{\nabla} \tilde{L}(\tilde{\mathbf{x}}) \\
&= \dot{\tilde{L}}(\tilde{\mathbf{x}}) < 0 : \forall \tilde{\mathbf{x}} \in \tilde{\Omega}, \tilde{\mathbf{x}} \neq \tilde{\mathbf{x}}^* \\
&\Rightarrow \dot{L}(\mathbf{x}) < 0 : \forall \mathbf{x} \in \Omega, \mathbf{x} \neq \mathbf{x}^* ,
\end{aligned} \tag{28}$$

where Eq. (25) and Eq. (26) were used for derivation.

It is of great interest how this framework affects the approximation capabilities of the underlying approach during transformation. It can be shown that the approximation is optimal in least squares sense which is summarized in the following proposition.

Proposition 2. Assume that the same prerequisites as in Prop. 1 are given. Then, it holds for all $L \in \mathcal{L}$ that the dynamical system $\mathbf{y} : \Omega \rightarrow \Omega$ approximates the data D in least squares sense if and only if $\tilde{\mathbf{y}}(\mathbf{x})$ approximates the transformed data set \tilde{D} in least squares sense.

PROOF. We assume that the mapping in the transformed space according to the learner $\tilde{\mathbf{y}} : \tilde{\Omega} \rightarrow \tilde{\Omega}$ approximates the data set $\tilde{D} = (\tilde{\mathbf{x}}(k), \tilde{\mathbf{v}}(k)) = (\tau_L(\mathbf{x}(k)), J_\tau^T(\mathbf{x}(k)) \cdot \mathbf{v}(k))$, i.e. that the learner itself is continuous and minimizes the following error

$$\tilde{E} = \sum_{k=1}^N \|\tilde{\mathbf{y}}(\tilde{\mathbf{x}}(k)) - \tilde{\mathbf{v}}(k)\|^2 \rightarrow \min . \tag{29}$$

The error in the original space Ω for a given data set D and a back-transformed dynamical system \mathbf{y} and the corresponding data set \tilde{D} in the transformed space $\tilde{\Omega}$ learned by $\tilde{\mathbf{y}}$ is given by

$$\begin{aligned}
E &= \sum_{k=1}^N \|\mathbf{y}(\mathbf{x}^i(k)) - \mathbf{v}^i(k)\|^2 \\
&= \sum_{k=1}^N \|J_\tau^{-T}(\tau(\mathbf{x}^i(k))) [\tilde{\mathbf{y}}(\tau(\mathbf{x}^i(k))) - \tilde{\mathbf{v}}^i(k)]\|^2 \\
&= \sum_{k=1}^N \underbrace{\|J_\tau^{-T}(\tilde{\mathbf{x}}^i(k))\|}_{\text{fixed}} \cdot \underbrace{\|\tilde{\mathbf{y}}(\tilde{\mathbf{x}}^i(k)) - \tilde{\mathbf{v}}^i(k)\|}_{\text{minimized in Eq. (29)}}^2 \rightarrow \min
\end{aligned} \tag{30}$$

This shows that the error E is decreasing for a given fixed transformation τ if the error in the transformed space \tilde{E} is minimized because \tilde{L} and \tilde{D} are consistent.

Note, that the proposition gives no specific information about the construction of the Lyapunov candidate and the diffeomorphism. The following sections introduce and rigorously analyze possible Lyapunov candidates and a corresponding diffeomorphism.

5. Learning Complex Lyapunov Candidates

This section investigates step 1) in Alg.1, the construction or learning of Lyapunov candidates from demonstrations.

5.1. Weighted Sum of Asymmetric Quadratic Functions (WSAQF)

The construction of valid Lyapunov candidates can be done in various ways. One option is to model the candidate function manually. However, this is potentially difficult and time consuming. We therefore suggest to apply automatic methods to learn valid Lyapunov candidate functions from data. A method that constructs Lyapunov candidates in a data-driven manner is the already mentioned weighted sum of asymmetric quadratic functions (WSAQF) [16]. The following equations describe the respective parametrization.

$$L(\mathbf{x}) = \mathbf{x}^T P^0 \mathbf{x} + \sum_{l=1}^{\mathcal{L}} \beta^l(\mathbf{x}) (\mathbf{x}^T P^l (\mathbf{x} - \boldsymbol{\mu}^l))^2 , \tag{31}$$

where we set $\mathbf{x}^* := 0$ for convenience. \mathcal{L} is the number of used asymmetric quadratic functions, $\boldsymbol{\mu}^l$ are mean vectors to shape the asymmetry of the functions, and $P^l \in \mathbb{R}^{d \times d}$ are positive definite matrices. The coefficients β are defined according to the following

$$\beta^l(\mathbf{x}) = \begin{cases} 1 : \mathbf{x}^T P^l (\mathbf{x} - \boldsymbol{\mu}^l) \geq 0 \\ 0 : \mathbf{x}^T P^l (\mathbf{x} - \boldsymbol{\mu}^l) < 0 \end{cases} , \tag{32}$$

Khansari-Zadeh et al. state that this scalar function is continuous and continuously differentiable. Furthermore, the function has a unique global minimum and therefore serves a potential control Lyapunov function. Learning is done by adaptation of the components of the matrices P^l and the vectors $\boldsymbol{\mu}^l$ in order to minimize the following constrained objective function

$$\begin{aligned}
\min \quad & \sum_{i=1}^{N_{\text{traj}}} \sum_{k=1}^{N^i} \frac{1 + \bar{w}}{2} \text{sign}(\psi^{ik}) \psi^{ik2} + \frac{1 - \bar{w}}{2} \psi^{ik2} , \tag{33} \\
\text{subject to} \quad & P^l \succ 0 : l = 0, \dots, \mathcal{L}
\end{aligned}$$

where \succ denotes the positive definiteness of a matrix and \bar{w} is a small positive scalar. The function ψ is defined according to the following

$$\psi^{ik} = \frac{\nabla L(\mathbf{x}^i(k))^T \mathbf{v}^i(k)}{\|\nabla L(\mathbf{x}^i(k))^T\| \cdot \|\mathbf{v}^i(k)\|} , \tag{34}$$

We show that this scalar function is a valid Lyapunov candidate.

Lemma 1. *The WSAQF approach $L : \Omega \rightarrow \mathbb{R}$ is a (global) Lyapunov candidate function according to Def. 2 and it holds that $(\mathbf{x} - \mathbf{x}^*)^T \cdot \nabla L > 0$.*

PROOF. Obviously, condition (i), (ii), and (iii) in Def. 2 are fulfilled. The function is also continuous and continuously differentiable despite the switches of β from zero to one or vice versa. In order to analyze condition (iv), the gradient is calculated.

$$\nabla L = (P^0 + P^{0T})\mathbf{x} + \sum_{l=1}^{\mathcal{L}} 2\beta^l(\mathbf{x})\mathbf{x}^T P^l (\mathbf{x} - \boldsymbol{\mu}^l) \cdot \left[(P^l + P^{lT})\mathbf{x} - P^l \boldsymbol{\mu}^l \right], \quad (35)$$

Condition (iv) holds because of the following inequality that demonstrates that L becomes a valid Lyapunov candidate according to Def. 2. Note that we still set $\mathbf{x}^* := 0$ for convenience without losing generality.

$$\begin{aligned} \mathbf{x}^T \cdot \nabla L &= \mathbf{x}^T (P^0 + P^{0T})\mathbf{x} + \sum_{l=1}^{\mathcal{L}} 2\beta^l(\mathbf{x})\mathbf{x}^T P^l (\mathbf{x} - \boldsymbol{\mu}^l) \cdot \underbrace{\left[\mathbf{x}^T (P^l + P^{lT})\mathbf{x} - \mathbf{x}^T P^l \boldsymbol{\mu}^l \right]}_{\mathbf{x}^T P^l (\mathbf{x} - \boldsymbol{\mu}^l) + \mathbf{x}^T P^{lT} \mathbf{x}} \\ &= \underbrace{\mathbf{x}^T (P^0 + P^{0T})\mathbf{x}}_{>0} + \sum_{l=1}^{\mathcal{L}} \left(\underbrace{2\beta^l(\mathbf{x})}_{\geq 0} \underbrace{(\mathbf{x}^T P^l (\mathbf{x} - \boldsymbol{\mu}^l))^2}_{\geq 0} \right. \\ &\quad \left. + \underbrace{2\beta^l(\mathbf{x})\mathbf{x}^T P^l (\mathbf{x} - \boldsymbol{\mu}^l)}_{\geq 0} \underbrace{\mathbf{x}^T P^{lT} \mathbf{x}}_{>0} \right) > 0 : \forall \mathbf{x} \in \Omega, \end{aligned} \quad (36)$$

where P^l are positive definite matrices and $\Omega = \mathbf{R}^d$. Please note that the transpose of a positive definite matrix is also positive definite. The WSAQF approach indeed constructs Lyapunov candidates that are radially unbounded because of its specific structure.

$$L(\mathbf{x}) = \underbrace{\mathbf{x}^T P^0 \mathbf{x}}_{\|\mathbf{x}\| \rightarrow \infty \Rightarrow \infty} + \sum_{l=1}^{\mathcal{L}} \underbrace{\beta^l(\mathbf{x}) (\mathbf{x}^T P^l (\mathbf{x} - \boldsymbol{\mu}^l))^2}_{\geq 0}. \quad (37)$$

such that the WSAQF approach becomes a globally defined Lyapunov candidate.

Also different methods to learn Lyapunov candidates are potentially applicable as long as the learned function satisfies the conditions in Def. 2.

5.2. Neurally-Imprinted Lyapunov Candidates (NILC)

The learning or construction of appropriate Lyapunov candidate functions from data is challenging. In previous work [19], we have already introduced a neural network approach called neurally imprinted Lyapunov candidate (NILC). This approach learns Lyapunov candidates $L : \Omega \rightarrow \mathbb{R}$ that are smooth and well suited to shape dynamical systems that in earlier work have been learned with neural networks as well [8]. We briefly introduce the method from [19].

Consider a neural network architecture which defines a scalar function $L : \mathbb{R}^d \rightarrow \mathbb{R}$. This network comprises three layers of neurons: $\mathbf{x} \in \mathbb{R}^d$ denotes the input, $\mathbf{h} \in \mathbb{R}^R$ the hidden, and $L \in \mathbb{R}$ the output neuron. The input is connected to the hidden layer through the input matrix $W^{\text{inp}} \in \mathbb{R}^{R \times d}$ which is randomly initialized and stays fixed during learning. The read-out matrix comprises the parameters subject to learning which is denoted by $W^{\text{out}} \in \mathbb{R}^R$. For input \mathbf{x} the output neuron is thus given by

$$L(\mathbf{x}) = \sum_{j=1}^R W_j^{\text{out}} f\left(\sum_{n=1}^d W_{jn}^{\text{inp}} x_n + b_j\right), \quad (38)$$

The main goal is to minimize the violation of the training data and the candidate function by making the negative gradient of this function follow the training data closely. A quadratic program is defined

$$\begin{aligned} \frac{1}{N_{\text{ds}}} \sum_{i=1}^{N_{\text{traj}}} \sum_{k=1}^{N^i} (\| -\nabla L(\mathbf{x}^i(k)) - \mathbf{v}^i(k) \|^2 + \dots + \epsilon_{\text{RR}} \|W^{\text{out}}\|^2) \rightarrow \min_{W^{\text{out}}}, \end{aligned} \quad (39)$$

subject to the following equality and inequality constraints corresponding to Lyapunov's conditions (i)-(iv) in theorem 1 such that L becomes a valid Lyapunov candidate function

$$\begin{aligned} \text{(a)} \quad & L(\mathbf{x}^*) = 0 & \text{(b)} \quad & L(\mathbf{x}) > 0 : \mathbf{x} \neq \mathbf{x}^* \\ \text{(c)} \quad & \nabla L(\mathbf{x}^*) = 0 & \text{(d)} \quad & \mathbf{x}^T \nabla L(\mathbf{x}) > 0 : \mathbf{x} \neq \mathbf{x}^* \end{aligned} \quad (40)$$

where the constraints (b) and (c) define inequality constraints which are implemented by sampling these constraints. The gradient of the scalar function defined by the network in Eq. (38) is linear in W^{out} and given by

$$(\nabla L(\mathbf{x}))_i = \sum_{j=1}^R W_j^{\text{out}} f' \left(\sum_{k=1}^d W_{jk}^{\text{inp}} x_k + b_j \right) \cdot W_{ji}^{\text{inp}}, \quad (41)$$

where f' denotes the first derivative of the Fermi function. The disadvantage of this approach is that the Lyapunov candidate is not globally valid. It can be extended towards predefined but finite regions. Interestingly, this candidate also fulfills the following condition: $(\mathbf{x} - \mathbf{x}^*)^T \cdot \nabla L > 0$, which is important for the diffeomorphic transformation that is defined in the following section. The result of this constructive approach is summarized in the following lemma:

Lemma 2. *The NILC approach $L : \Omega \rightarrow \mathbb{R}$ is a (local) Lyapunov candidate function according to Def. 2 and it holds that $(\mathbf{x} - \mathbf{x}^*)^T \cdot \nabla L > 0$.*

The previous section revealed that arbitrary Lyapunov function candidates are applicable for the learning of stable dynamical systems, if a diffeomorphism is given that transforms this candidate into a quadratic function. The following section defines and investigates a corresponding diffeomorphism for the NILC and the WSAQF Lyapunov candidate approaches.

6. Coping with Complex Lyapunov Candidates: The Diffeomorphic Candidate Transformation

This section defines step **2**) of Alg.1 in detail. In order to allow an implementation of flexible Lyapunov candidates $L : \Omega \rightarrow \mathbb{R}$, a diffeomorphic candidate transformation $\tau : \mathcal{L} \times \Omega \rightarrow \tilde{\Omega}$, $\mathbf{x} \mapsto \tilde{\mathbf{x}}$ is defined as follows

$$\tau_L(\mathbf{x}) = \begin{cases} \sqrt{L(\mathbf{x})} \cdot \frac{\mathbf{x}-\mathbf{x}^*}{\|\mathbf{x}-\mathbf{x}^*\|} & \text{if } \mathbf{x} \neq \mathbf{x}^* \\ \mathbf{x}^* & \text{if } \mathbf{x} = \mathbf{x}^* \end{cases} . \quad (42)$$

This mapping transforms each Lyapunov candidate L according to Def. 2 into a quadratic function $\tilde{L} : \tilde{\Omega} \rightarrow \mathbb{R}$, $\tilde{\mathbf{x}} \mapsto \tilde{\mathbf{x}}^2$ stated by the following lemma.

Lemma 3. *The mapping $\tau : \mathcal{L} \times \Omega \rightarrow \tilde{\Omega}$ is a diffeomorphic candidate transformation according to Def. 3 that corresponds to the set of Lyapunov candidates \mathcal{L} where each element $L \in \mathcal{L}$ fulfills $(\mathbf{x} - \mathbf{x}^*)^T \cdot \nabla L > 0 : \mathbf{x} \in \Omega$, i.e. $\tau_L : \Omega \rightarrow \tilde{\Omega}$ is bijective, continuous, continuously differentiable, and the inverse mapping $\tau_L^{-1} : \tilde{\Omega} \rightarrow \Omega$ is also continuous and continuously differentiable. Further, τ transforms functions $L \in \mathcal{L}$ to the fixed quadratic function $\tilde{L} : \tilde{\Omega} \rightarrow \mathbb{R}$, $\tilde{\mathbf{x}} \mapsto \tilde{\mathbf{x}}^2$.*

PROOF. We again define $\tau := \tau_L$ and set $\mathbf{x}^* = \mathbf{0}$ for convenience. At first, it is obvious that $\tau : \Omega \rightarrow \tilde{\Omega}$ is continuous and continuously differentiable, because L is continuous and continuously differentiable. Importantly, the diffeomorphism is injective, i.e.

$$\forall \mathbf{x}_1, \mathbf{x}_2 \in \Omega : (\mathbf{x}_1 \neq \mathbf{x}_2 \Rightarrow \tau(\mathbf{x}_1) \neq \tau(\mathbf{x}_2)) \quad (43)$$

If $\mathbf{x}_1, \mathbf{x}_2 \in \Omega$ are arbitrary vectors with $\mathbf{x}_1 \neq \mathbf{x}_2$ and $\mathbf{x}_{1,2} \neq \mathbf{0}$, four different cases are distinguished

- (1) $L(\mathbf{x}_1) \neq L(\mathbf{x}_2)$ and $\mathbf{x}_1 \approx \mathbf{x}_2$
- (2) $L(\mathbf{x}_1) = L(\mathbf{x}_2)$ and $\mathbf{x}_1 \approx \mathbf{x}_2$
- (3) $L(\mathbf{x}_1) \neq L(\mathbf{x}_2)$ and $\mathbf{x}_1 \sim \mathbf{x}_2$
- (4) $L(\mathbf{x}_1) = L(\mathbf{x}_2)$ and $\mathbf{x}_1 \sim \mathbf{x}_2$,

where $\mathbf{x} \sim \mathbf{y}$ means that there exists a real number $\lambda > 0$ for which $\mathbf{x} = \lambda \mathbf{y}$ holds. Cases (1) and (2) are unproblematic because $\tau(\mathbf{x}_1) \neq \tau(\mathbf{x}_2)$ directly follows from $\mathbf{x}_1 \approx \mathbf{x}_2$. In order to analyze case (3), we calculate the directional derivative of L along the direction of \mathbf{x} which exists due to the total differentiability of τ and satisfies the following inequality

$$\nabla_{\mathbf{x}} L(\mathbf{x}) = \mathbf{x}^T \nabla L(\mathbf{x}) > 0 . \quad (45)$$

This is directly according to condition (iv) of the considered Lyapunov candidate. L is thus strictly monotonically increasing along a given direction in Ω . With $L(\mathbf{x}_1) \neq L(\mathbf{x}_2)$ we therefore infer that $\|\tau(\mathbf{x}_1)\| \neq \|\tau(\mathbf{x}_2)\|$ and thus $\tau(\mathbf{x}_1) \neq \tau(\mathbf{x}_2)$. Case (4) is invalid, because $L(\mathbf{x}_1) = L(\mathbf{x}_2) \Rightarrow \|\tau(\mathbf{x}_1)\| = \|\tau(\mathbf{x}_2)\|$ and with $\mathbf{x}_1 \sim \mathbf{x}_2$ it follows that $\mathbf{x}_1 = \mathbf{x}_2$ which is in contradiction to the assumption that $\mathbf{x}_1 \neq \mathbf{x}_2$. Therefore, τ is injective. It

directly follows that $\tau : \Omega \rightarrow \tilde{\Omega}$ is surjective because $\tilde{\Omega}$ is the image of τ and thus bijective.

The inverse function $\tau^{-1} : \tilde{\Omega} \rightarrow \Omega$ exists because of the bijectivity and is continuous and continuously differentiable. The reason is that the directional derivative of $L(\mathbf{x})$ along \mathbf{x} is strictly monotonically increasing.

In order to show that the diffeomorphism τ maps each L onto the fixed function $\tilde{L} : \tilde{\Omega} \rightarrow \mathbb{R}$, $\tilde{\mathbf{x}} \mapsto \tilde{\mathbf{x}}^2$, the following equivalence holds per definition

$$\|\tau(\mathbf{x})\| = \sqrt{L(\mathbf{x})} \Leftrightarrow \|\tau(\mathbf{x})\|^2 = L(\mathbf{x}) . \quad (46)$$

The transformed function becomes quadratic with the use of Eq. (46)

$$\tilde{L}(\tilde{\mathbf{x}}) = L(\tau^{-1}(\tilde{\mathbf{x}})) = \|\tau(\tau^{-1}(\tilde{\mathbf{x}}))\|^2 = \|\tilde{\mathbf{x}}\|^2 . \quad (47)$$

Each Lyapunov candidate that satisfies $(\mathbf{x} - \mathbf{x}^*)^T \cdot \nabla L > 0$ (such as the NILC and the WSAQF approach) and the diffeomorphism in Lem. 3 are therefore applicable for implementation of flexible and desired Lyapunov candidates with the τ -SEDS approach.

In this particular case, the Jacobian $J_\tau(\mathbf{x})$ of the diffeomorphism τ_L can be derived analytically. We again set $\mathbf{x}^* = \mathbf{0}$ for simplicity.

$$\begin{aligned} J_\tau(\mathbf{x})_{ij} &= \frac{\partial}{\partial x_i} \tau_j(\mathbf{x}) \\ &= \frac{\partial}{\partial x_i} \sqrt{L(\mathbf{x})} \cdot \frac{x_j}{\|\mathbf{x}\|} \\ J_\tau(\mathbf{x}) &= \frac{\nabla L(\mathbf{x})}{2\sqrt{L(\mathbf{x})}} \cdot \frac{\mathbf{x}^T}{\|\mathbf{x}\|} + \sqrt{L(\mathbf{x})} \left(\frac{I}{\|\mathbf{x}\|} - \frac{\mathbf{x}\mathbf{x}^T}{\|\mathbf{x}\|^3} \right) , \end{aligned} \quad (48)$$

where $I \in \mathbb{R}^{d \times d}$ is the identity matrix, $L(\mathbf{x})$ is the Lyapunov candidate and $\nabla L(\mathbf{x})$ denotes the gradient of the Lyapunov candidate. It is important to note that this Jacobian has a removable singularity at $\mathbf{x} = \mathbf{0}$ and is thus well-defined for the limit case of $\|\mathbf{x}\| \rightarrow 0$ where $J_\tau(\mathbf{x}) = \mathbf{0} : \mathbf{x} = \mathbf{0}$.

This approach based on the framework for diffeomorphic candidate transformations τ introduced in this section that applies SEDS and WSAQF or NILC as a basis for learning is called τ -SEDS (WSAQF, NILC) or simply τ -SEDS in the following. Note that this theoretical framework is not restricted to the special forms of the used approaches and thus serves as fundamental framework for learning complex motions under diffeomorphic transformations.

7. Experimental Results

This section introduces the experimental results obtained for the different approaches and compares them qualitatively and quantitatively. This comprises the steps **3**) to **5**) of Alg.1.

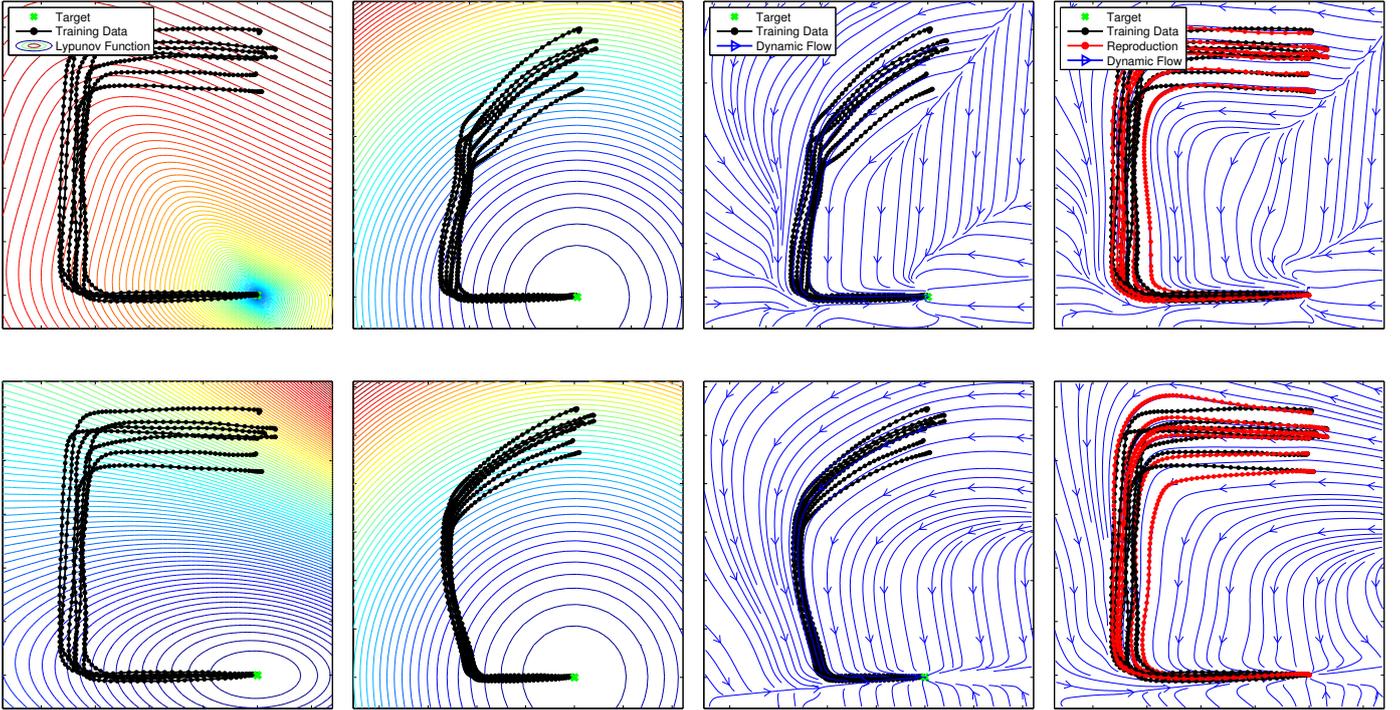


Figure 4: Lyapunov candidate constructed by the WSAQF approach (first row), and Lyapunov candidate originating from the NILC approach (second row). Desired Lyapunov function L and data set D in the original space Ω (first column). Transformed Lyapunov function \tilde{L} and transformed data set \tilde{D} in $\tilde{\Omega}$ (second column). Dynamical system $\tilde{\mathbf{y}}$ learned by SEDS using data set \tilde{D} which admits to a quadratic Lyapunov function in the transformed space $\tilde{\Omega}$ (third column). The result \mathbf{y} in Ω after applying the inverse transformation τ^{-1} of $\tilde{\mathbf{y}}$ (fourth column).

7.1. Reproduction Accuracy

To measure the accuracy of a reproduction is an important tool to evaluate the performance of a movement generation method. We use the swept error area⁵ (SEA) as an error functional to evaluate the reproduction accuracy of the methods. It is computed by

$$E = \frac{1}{N} \sum_{i=1}^{N_{\text{traj}}} \sum_{k=1}^{N^i-1} A(\hat{\mathbf{x}}^i(k), \hat{\mathbf{x}}^i(k+1), \mathbf{x}^i(k), \mathbf{x}^i(k+1)) \quad (49)$$

where $\hat{\mathbf{x}}^i(\cdot)$ is the equidistantly re-sampled reproduction of the demonstration $\mathbf{x}^i(\cdot)$ with the same number of samples N^i and $A(\cdot)$ denotes the function which calculates the area of the enclosed tetragon generated by the four points $\hat{\mathbf{x}}^i(k)$, $\hat{\mathbf{x}}^i(k+1)$, $\mathbf{x}^i(k)$, and $\mathbf{x}^i(k+1)$.

7.2. Illustrative Example: τ -SEDS

This experiment illustrates the processes of diffeomorphic transformation and learning of the SEDS approach in combination with the WSAQF and NILC Lyapunov candidates. The experimental results are again obtained for a sharp-C-like movement from a library of 30 human handwriting motions called LASA data set [14]. This data provide realistic handwritten motions and is used in several

different studies about the learning of dynamical systems applied for movement generation [13, 19, 8]. As mentioned, the movement violates a quadratic Lyapunov candidate (shown in Fig. 1). The previously introduced Lyapunov candidates are used for transformation and comparison. The first candidate function is constructed by means of the NILC technique [19] (results shown in first row). The second function is obtained with the WSAQF approach (second row). Learning in the transformed space is done by SEDS⁶, which is initialized with $K = 5$ Gaussian functions and trained for maximal 500 iterations. The function τ (see Eq. (42)) is used as corresponding diffeomorphic candidate transformation.

Fig. 4 illustrates the intermediate steps obtained during the learning and transformation phase. The plots in the first column in Fig. 4 show the different Lyapunov candidates that are consistent with the respective six demonstrations. The training data set D is then prepared for learning by transforming the data set into \tilde{D} that is defined in the transformed space which is consistent with a quadratic Lyapunov candidate $\tilde{L}(\tilde{\mathbf{x}})$. The result of the data transformation and the Lyapunov candidate is illustrated in Fig. 4 (second column). We then apply the SEDS learning approach to obtain $\tilde{\mathbf{y}}$ which is stable according to a quadratic Lyapunov function \tilde{L} in $\tilde{\Omega}$ after learning

⁵This measure was first defined in [16].

⁶We used the SEDS software 1.95 by Khansari-Zadeh et al. [22]

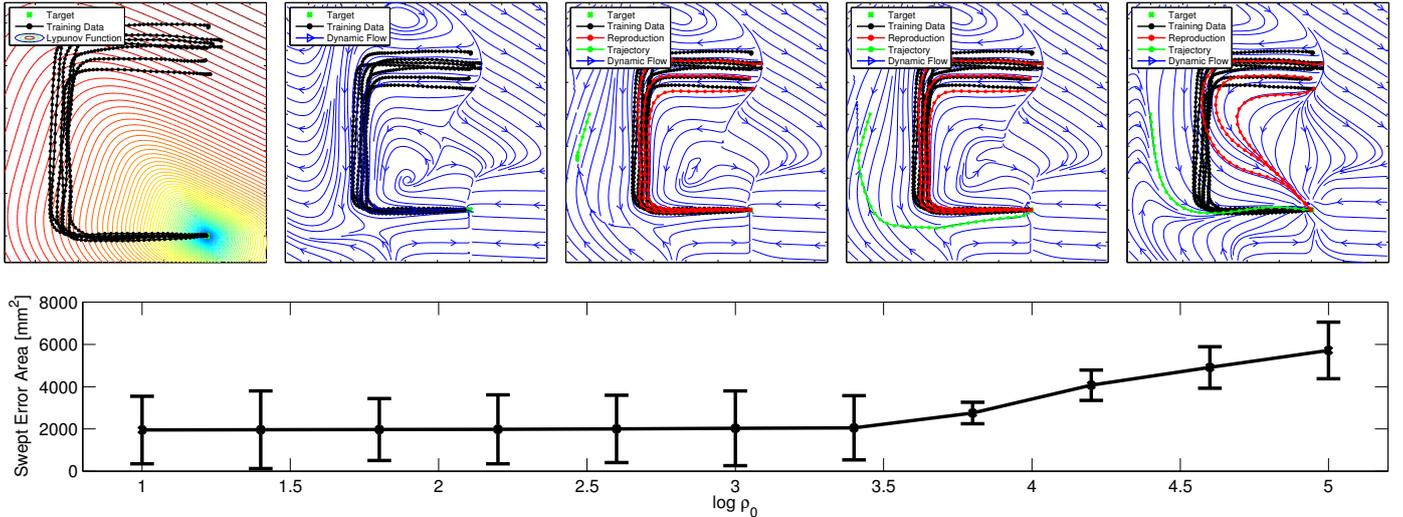


Figure 5: Explicit stabilization of the sharp-C-shape through CLF. A Lyapunov candidate function learned with the WSAQF approach (top left). An unstable dynamical system of six demonstrations with GMR (top second). The stabilized system for three demonstrations with parameter $\rho_0 = 10$ (top third), $\rho_0 = 1000$ (top fourth), $\rho_0 = 100000$ (top fifth). SEA of the stabilized system with changing ρ_0 (bottom).

the data \tilde{D} . The result of the learning is depicted by the dynamic flow after learning the transformed demonstrations \tilde{D} in Fig. 4 (third column). It is illustrated that the new data is not violating the quadratic function and thus allows an accurate modeling of the data. Finally, the inverse transformation τ_L^{-1} is used to obtain the dynamics \mathbf{y} for the original data D in the original space Ω (step 5). Eq. (18) was used for back-transformation. Note that the obtained vector field has no discontinuities and provides a gentle generalization of the applied data set D irrespective of the used Lyapunov candidate L , see Fig. 4 (fourth column).

The experiment shows that the class of learnable demonstration of SEDS is enhanced by means of the proposed framework based on diffeomorphic transformations. The experiment also reveals that the generalization capability of the learner transfers to the original space which is an important prerequisite for such systems. Please compare the results of this experiment to Fig. 1 and Fig. 3.

7.3. Investigating the Control Lyapunov Approach

The explicit stabilization during runtime with online corrections of the learned dynamical system in the CLF-DM approach is parameterized with ρ_0 and κ_0 defining the function $\rho(\|\mathbf{x}\|)$, see Eq. (13), which shifts the activation threshold of a correction signal $\mathbf{u}(\mathbf{x})$. Basically, two fundamental problems concerning these parameters are inherent to this approach. First, the parameters should be selected and scaled according to the scalar product $\nabla L(\mathbf{x})^T \hat{\mathbf{y}}(\mathbf{x})$ in order to allow an appropriate stabilization, where $\hat{\mathbf{y}}$ is defined according to Eq. (12). The optimization process of these parameters is independent of the actual learning of the Lyapunov candidate L , hence, the learning of $\hat{\mathbf{y}}$ constitutes a separate process. Optimization of these param-

eters usually requires several iterations and is thus computationally expensive. Second, the parameters can only deal with a finite range of the scalar product $\nabla L(\mathbf{x})^T \hat{\mathbf{y}}(\mathbf{x})$. This is particularly problematic whenever the scalar product is too small in some region and at the same time too big for $\rho(\|\mathbf{x}\|)$ in another. This can lead to inaccurate reproduction capabilities or numerical integration problems. The respective parameterization appears to be too simple in such situations. However, the introduction of more parameters is unsatisfying.

Fig. 5 demonstrates the effect of the parameterization and shows the learning of six demonstrations by means of the CLF approach. The first two plots in the figure shows the result of the WSAQF approach for learning the Lyapunov candidate (top first) and the learning of the dynamical system by means of the Gaussian mixture regression (GMR) approach (top second). As expected, the simple GM regression method results in an unstable estimate of the three demonstrations. The second row of Fig. 5 shows the experimental results. We selected $\kappa_0 = 0.05$ which remains fixed, and varied ρ_0 in the range from [10, 100000] logarithmically in 11 steps. We recorded the SEA measure in this experiment. The bottom plot of Fig. 5 shows the SEA of the demonstrations and the respective reproductions. It is demonstrated that the reproduction accuracy decreases with increasing ρ_0 . For small ρ_0 , the reproduction accuracy is high, while for big ρ_0 , the reproductions become inaccurate. In this case, the reproduction is forced to converge faster and thus follows the gradient of the Lyapunov candidate rather than the demonstrations. However, a too strong reduction of ρ_0 is also an insufficient strategy. The reason is that the correction signal is too small and can hinder convergence to the target attractor by induction of numerical spurious attractors. This is es-

pecially problematic if perturbations are present and drive the trajectory in such a spurious attractor. This can be avoided by increasing ρ_0 which simultaneously increases the CS and thus “overwrites” the numerical attractors. These both facts introduce a trade-off in the CLF approach for which a careful parameter selection is necessary.

The three top right plots in Fig. 5 visualize these situations for $\rho_0 = 10, 1000, 100000$ from left to right. The plots comprise the demonstrations used as training data (black trajectories), the reproduction by the dynamical system (red trajectories), the dynamical flow (blue trajectories and arrows), and a trajectory (green trajectory) that was iterated for one hundred steps. Overall, the left plot shows good reproduction accuracy but also a spurious numerical attractor at the end of the green trajectory; the center plot shows a good reproduction and convergence capability for which tuning the CLF parameters was successful; and the right plot shows a correction signal that is too strong which results in a poor reproduction of the demonstrations and a strong convergence behavior.

These effects are especially problematic if we assume that two different Lyapunov candidates $L \sim L'$ are given, which are equivalent in the sense that their gradients point into the same direction: $\nabla L \sim \nabla L'$. These functions can be transformed into each other by means of a continuously differentiable function $\varphi(L) : \mathbb{R} \rightarrow \mathbb{R}$ with $\varphi(0) = 0$ and $\frac{\partial}{\partial L}\varphi > 0$. This has a direct influence on the diffeomorphic transformation which copes with the function $\varphi(L(\mathbf{x})) : \Omega \rightarrow \mathbb{R}$. Only the training data \tilde{D} in the transformed space $\tilde{\Omega}$ looks different. These changes disappear after back transformation of the dynamical system into the original space. In contrast, the parameters κ_0 and ρ_0 of the correction signal are not automatically related to φ and need to be re-tuned whenever a new Lyapunov candidate is applied.

In summary, this sensitivity of the CLF approach to the parameters of the runtime correction is unsatisfactory to the degree that a separate optimization process -which is computationally costly- is required. The diffeomorphic transformation approach τ -SEDS, however, requires no additional parameters since it merges the learning of the Lyapunov candidate and the actual dynamics through the SEDS approach.

7.4. Performance Experiments

This experiment compares the proposed approach τ -SEDS with the state of the art methods and SEDS as baseline in a qualitative and quantitative manner. The performance of the different approaches is analyzed on the LASA data set [14]. For evaluation, we use the neurally imprinted Lyapunov candidate (NILC) and the weighted sum of asymmetric quadratic functions (WSAQF) approach as Lyapunov candidates and combine them with Gaussian mixture regression through the control Lyapunov function - dynamic movements (CLF-DM) approach; the stable estimator of dynamical systems through the diffeomorphic transformation (τ -SEDS) approach; and neurally im-

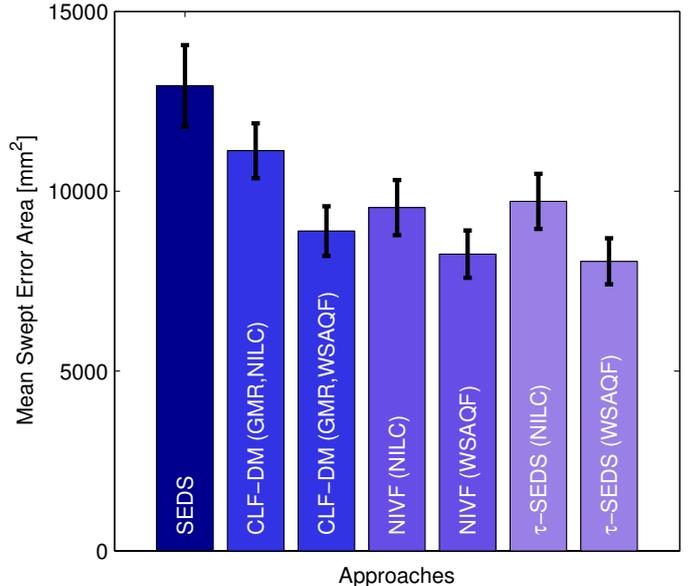


Figure 6: The SEA for the different method combinations on the library of 30 handwritten motions.

printed vector fields (NIVF) through quadratic programming. The accuracy of the reproductions is again measured according to the SEA. The results are stated in Fig. 6.

The WSAQF Lyapunov candidates are learned and optimized according to the following: the trade-off parameter $\bar{w} = 0.9$ is fixed; an additional regularization term⁷ $\lambda \cdot \|\theta\|^2$ is added in order to obtain smoother functions; and the number of asymmetric functions \mathcal{L} is increased until a minimum of the violation between Lyapunov candidate L and demonstrations D is observed. The NILC candidates are based on a network with $R = 100$ hidden neurons where the regularization parameter is decreased from 10^0 to 10^{-5} logarithmically in 5 equidistant steps until the violation reaches a minimum. The Lyapunov properties are validated in $N_C = 10^5$ uniformly distributed samples⁸.

The results for each shape are averaged over ten initializations. The SEDS and GMR models were initialized with $K = 7$ Gaussian functions in the mean square error (MSE) mode and trained for maximally 1500 iterations. The parameters of the CLF integration were selected from 9 logarithmically equidistant steps from 10^{-6} to 10^3 .

The experiments first reveal that the average SEA of the SEDS approach is extraordinary large in comparison to the other approaches. The reason is that some of the 30 shapes (e.g. the sharp-C-shape) are violating a quadratic Lyapunov candidate function. This leads to inaccurate reproductions and thus to a poor average performance, because these shapes are principally not learnable by SEDS.

⁷ $\theta = \{P^0, \dots, P^{\mathcal{L}}, \mu^0, \dots, \mu^{\mathcal{L}}\}$ collects all parameters of the WSAQF approach. $\lambda = 0.1$ in the experiments.

⁸see [19] for details

Approach (LC)	Dynamics	Stability	Integration
SEDS (Quad.)	$L = \tilde{x}^T \tilde{x}$	Global	Yes
CLF-DM (NILC)	$\tilde{x}^T \nabla L > 0$	Local	No
CLF-DM (WSAQF)	$\tilde{x}^T \nabla L > 0$	Global	No
NIVF (NILC)	$\tilde{x}^T \nabla L > 0$	Local	Yes
NIVF (WSAQF)	$\tilde{x}^T \nabla L > 0$	Local	Yes
τ -SEDS (NILC)	$\tilde{x}^T \nabla L > 0$	Local	Yes
τ -SEDS (WSAQF)	$\tilde{x}^T \nabla L > 0$	Global	Yes

Table 1: Qualitative comparison of the different method combinations. The τ -SEDS (WSAQF) approach is promising. Approaches which use the NILC approach as Lyapunov candidate cannot guarantee stability globally.

The performance of the dynamical systems explicitly stabilized by the CLF approach is significantly better than for the original SEDS approach, but slightly worse than for the other approaches. This is due to the fact that the selection of the CLF parameters was restricted to a discrete set of variables and that the parameterization of ρ is insufficient for learning of some of the shapes. However, the learning performance can in principle be increased but would demand more computational resources.

The τ -SEDS and the NIVF approach reach the best results among the tested methods. The difference in the results originating from the two Lyapunov candidates are non-negligible. The WSAQF approach performs slightly better than the NILC due to the fact that the error functional also directly implements a reduction of the violation, see Eq. (33). The simple alignment of the candidates’ gradient and the velocity of the demonstrations is -in some cases- not sufficient to implement a violation-free Lyapunov candidate with NILC. In fact, it is also possible to use the learning functional of the WSAQF approach for the NILC approach and vice versa.

An additional qualitative summary of the several approaches can be found in Tab. 1. The table assigns three important properties to the discussed approaches. The first property is the class of learnable functions. SEDS is in principle restricted to dynamics that satisfy a quadratic Lyapunov function. All other approaches allow much larger classes of dynamics irrespective if the NILC or the WSAQF approach are applied. The second column states the range of the stability property and distinguishes between local and numerical stability guarantees or constructively proven global asymptotic stability. The last column in the table states that the Lyapunov candidate is integrated into the learning procedure. This is not the case for the CLF-DM approach because stabilization is only applied online. The τ -SEDS (WSAQF) approach appears to be the only approach which provides a large class of learnable demonstrations, allows global stability that is proven constructively and integrates the Lyapunov candidate into the learning of the dynamical system directly while simultaneously performing in a reliable manner. We provide the results for this approach for all 30 movements in Fig. 7.

8. Robotics Experiment

We apply the presented approach in a robotic scenario involving the humanoid robot iCub [23]. Such robots are typically designed to solve service tasks in environments where a high flexibility is required. Robust adaptability by means of learning is thus a prerequisite for such systems. The experimental setting is illustrated in Fig. 8 (left). A human tutor physically guides iCub’s right arm in the sense of kinesthetic teaching using a recently established force control on the robot. The tutor can thereby actively move all joints of the arm to place the end-effector at the desired position. Beginning on the right side of the workspace, the tutor first moves the arm around the obstacle on the table, touches its top, and then moves the arm towards the left side of the obstacle where the movement stops. This procedure is repeated three times.

The recorded demonstrations comprise between $N_{\text{traj}} = 542$ and $N_{\text{traj}} = 644$ samples. We apply the original SEDS and τ -SEDS (WSAQF) approach to learn the demonstrations and equip SEDS in both cases with $K = 5$ Gaussians and iterate for maximally 1500 steps. The WSAQF was parameterized with $\lambda = 0.01$ and $\bar{w} = 0.9$ and comprised $\mathcal{L} = 3$ basis functions. The results of the experiment are visualized in Fig. 8. The center plot of the figure shows the result of learning the robot demonstrations with the SEDS approach. The dynamical system constructed by SEDS is not able to follow the demonstrations because the dynamics are restricted to a quadratic Lyapunov function. The right plot of Fig. 8 visualizes the back transformation of the SEDS dynamics into the original space which yields good reproductions while simultaneously guaranteeing asymptotic stability by construction. A closer inspection reveals that the reproductions from the learned dynamical system actually both are smoother and emphasize more clearly the main feature of the touching the utmost point of the tower, thereby not smoothing out this important feature of the demonstrated movement.

9. Discussion

The experimental results obtained by application of the theoretically derived framework substantiates the success of this transformation procedure. Proposition 1 and Proposition 2 left many degrees of freedom that can be used in different ways as we performed in this paper. Therefore, many questions arise that we discuss in this chapter. It is, however, important to note that the majority of the answers given are based on empirical observations and conjectures. We nevertheless believe that these points are interesting to focus on in the recent future.

- What are “good” Lyapunov candidates? One of the main requirements on Lyapunov candidates is that they are consistent with the demonstrations - learning thus appears as a method of choice to obtain an appropriate candidate. Another important point is

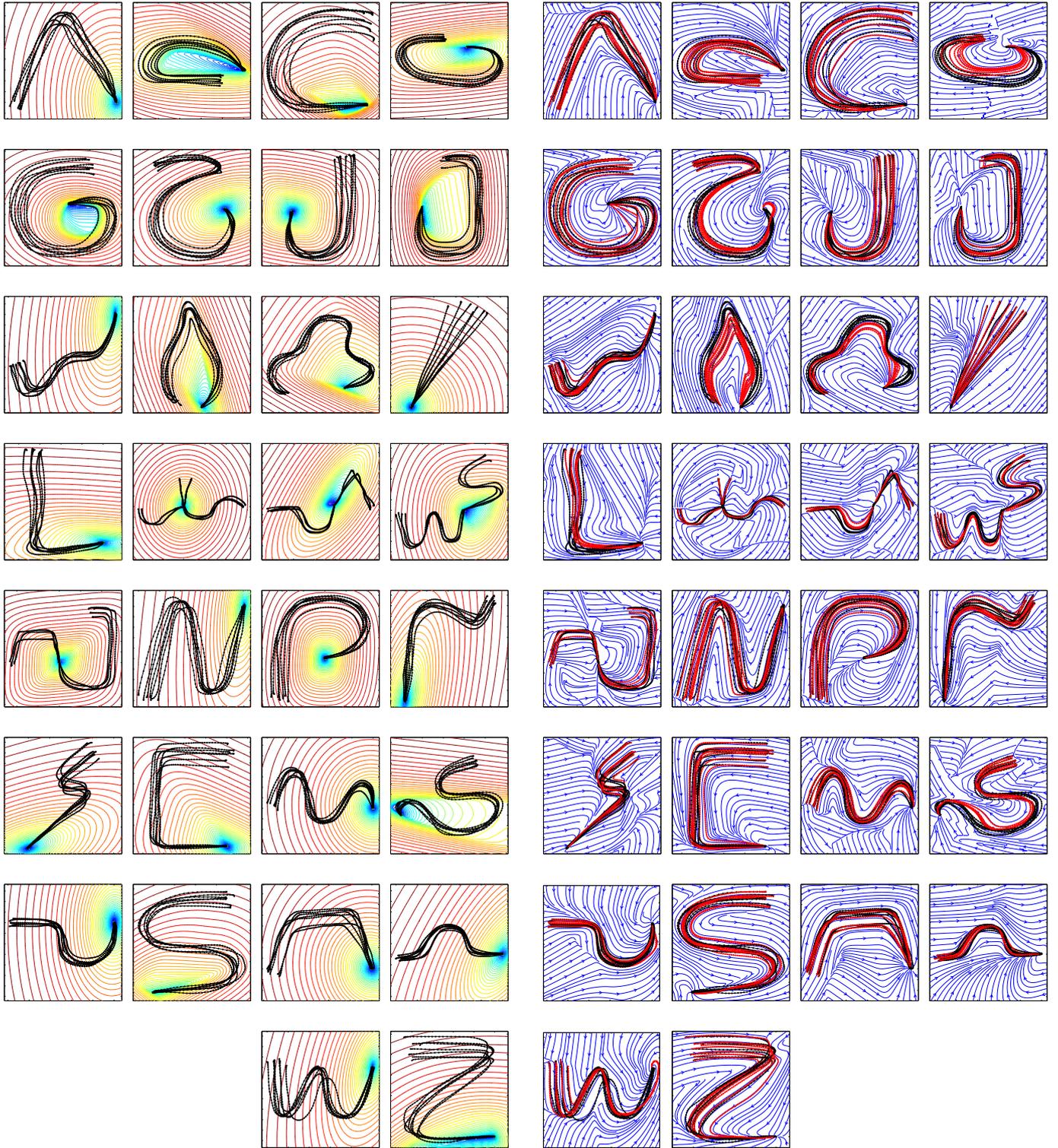


Figure 7: The collection of Lyapunov candidates constructed with the WSAQF approach for the 30 handwritten motions (left block). The corresponding dynamical systems constructed with the τ -SEDS (WSAQF) approach. This approach generates accurate and stable movements.

that robotics movements should be smooth. Strong accelerations are undesired because they are dangerous, both, for humans in the vicinity of the robot and for the robot itself. The applied Lyapunov can-

didate is a major factor concerning the smoothness of the resulting dynamics. The construction of smooth scalar functions that reduce the risk of undesired jerkiness is thus indispensable.

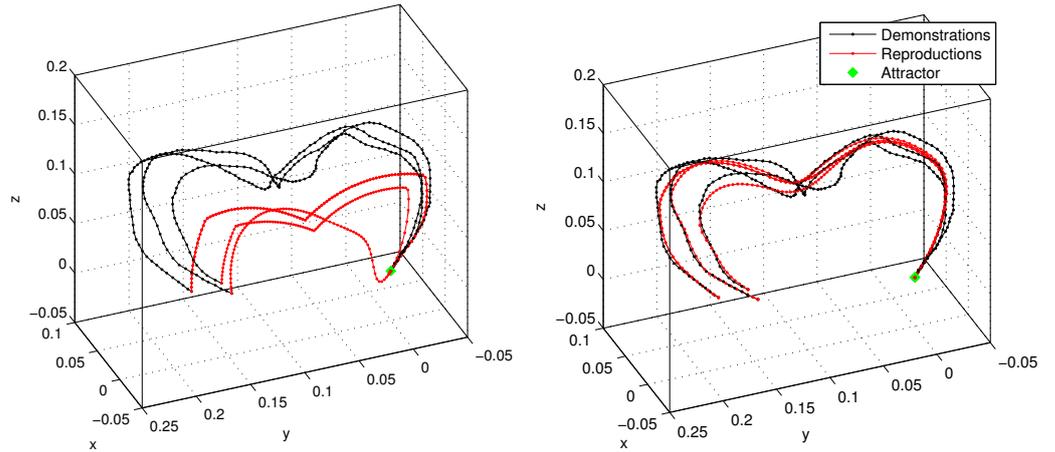


Figure 8: Kinesthetic teaching of iCub and the results of the iCub experiment. The tutor moves iCub’s right arm from the right to the left side of the small colored tower (left). Reproduction (red) of the demonstrated trajectories (black) in meter by the original SEDS approach (center) which is inaccurate. The reproductions of τ -SEDS with WSAQF candidate according to the back transformation in the original space Ω are accurate and stable (right).

- What is the class of learnable dynamics? The class of learnable dynamics is mainly driven by the Lyapunov candidate function (WSAQF, NILC). The diffeomorphic candidate transformation $\tau_L : \Omega \rightarrow \tilde{\Omega}$ defined in Eq. (42) requires Lyapunov candidates that fulfill the inequality $(\mathbf{x} - \mathbf{x}^*)^T \cdot \nabla L > 0$, see Eq. (36). Hence, the class is indeed restricted but still much larger than for quadratic Lyapunov candidates. It is nevertheless worth to investigate Lyapunov candidates and diffeomorphic candidate transformations that allow more complex dynamics or even universal approximation capabilities.
- What are “good” diffeomorphisms? It is clear that the diffeomorphic candidate transformation τ has an impact on the resulting dynamics. The differential properties of the learned dynamical system transfer via τ_L into the original space. It is, however, unclear how the properties change after back transformation. We believe that the diffeomorphism should be curved only as much as necessary (to obtain the quadratic Lyapunov candidate) and as less as possible (to keep the differential properties rather unchanged).
- What is the role of the learner? The learner is certainly the major ingredient for a successful learning of a non-linear dynamical system. Different properties are very important, such as the generalization ability, the smoothness of the solution and the space $\tilde{\Omega}$ for which asymptotic stability can be guaranteed. SEDS, e.g., is globally stable ($\tilde{\Omega} = \mathbb{R}^d$) which results also in a global solution after back transformation if the Lyapunov candidate is also globally (such as the WSAQF approach) valid in $\Omega = \mathbb{R}^d$.
- How is the generalization ability of the learner affected by the transformation? The learner is sup-

posed to minimize the error in the transformed space $\tilde{\Omega}$. However, the back transformation changes the error values according to Eq. (30). The experiments showed that this had no negative effect but without any theoretical justification.

- How is the proposed approach related to the idea of movement primitives? One of the key features of movement primitives is that they can be superimposed to create more complex motions without inducing unstable behavior. The super-position of models based on the original SEDS formulation includes this feature of stability but is restricted to the same class of learnable dynamics as the SEDS approach itself, i.e. superimposed SEDS movements are stable according to a quadratic Lyapunov function. The resulting dynamics of two super-imposed τ -SEDS models are not necessarily stable because of the potentially different underlying Lyapunov candidates used for transformation. It is nevertheless possible to guarantee stability of the resulting dynamics if the used Lyapunov candidates are identical. How to choose an applicable Lyapunov candidate which satisfies the requirements of all demonstrations at the same time is yet unclear.
- Is the robot’s stability guaranteed? It is mathematically proven that the dynamical systems used for motion control of the robot are stable if used the mentioned movement primitive approaches. This does not necessarily mean that the movement of the real robot is also stable. However, several recent experimental results showed that this seems practically irrelevant.

The answering to these questions will be left for future research. The following section concludes the paper.

10. Conclusion

SEDS is a very exciting approach to learn dynamical systems while simultaneously ensuring global asymptotic stability. One of the main advantages is that SEDS guarantees stability of point-to-point movements globally by construction. However, the class of dynamics that are accurately learnable is restricted to be consistent with a quadratic Lyapunov function. This property is undesired because it effectively prevents the learning of complex movements.

Other approaches such as CLF-DM or NIVF enable learning of larger sets of dynamics. However, not without the disadvantages of a correction signal or the lack of constructive mathematical guarantees.

This paper therefore proposes a theoretical framework that enhances the class of learnable movements by employing SEDS⁹ indirectly after a diffeomorphic transformation. In comparison to the state of the art, the proposed τ -SEDS (WSAQF) approach appears to be the only approach which provides a large class of learnable demonstrations, allows global stability that is proven constructively and integrates the Lyapunov candidate into the learning of the dynamical system directly while simultaneously performing in a reliable manner.

The key idea is to build a flexible data-driven Lyapunov candidate that is consistent with the given demonstrations. The diffeomorphic candidate transformation τ is then used to map the data set into the transformed space where the demonstrations follow a quadratic Lyapunov function. Learning is applied on the transformed data set by SEDS. The learned dynamical system will then accurately reproduce the demonstrations while simultaneously satisfying the conditions for asymptotic fixpoint stability. Finally, the back transformation of the dynamical system is performed. Complex dynamics that are accurately following the original demonstrations that are at the same time stable according to the previously defined Lyapunov candidate are obtained; i.e. the Lyapunov candidate becomes a Lyapunov function for the back-transformed dynamics. This new approach is called τ -SEDS. Interestingly, we could easily reuse the complete implementation of the SEDS approach. This means that the framework allows a modular implementation without much coding effort.

The theoretical results are complemented by experimental results from robotics which illustrate the effect of the learning and transformation. The generality of the framework is demonstrated by using different Lyapunov candidates. This emphasizes the fact that this framework is not restricted to the special form of the Lyapunov candidate and the SEDS approach.

⁹We thank S. Mohammad Khansari-Zadeh and the LASA Lab at EPFL for providing the open source software of SEDS.

ACKNOWLEDGMENT

This research is funded by the German BMBF within the Intelligent Technical Systems Leading-Edge Cluster.

References

- [1] M. Mühlhig, M. Gienger, J. J. Steil, Interactive imitation learning of object movement skills, *Autonomous Robots* 32 (2) (2012) 97–114.
- [2] A. Pistillo, S. Calinon, D. G. Caldwell, Bilateral physical interaction with a robot manipulator through a weighted combination of flow fields, in: *IEEE Conf. IROS*, 2011, pp. 3047–3052.
- [3] A. Billard, S. Calinon, R. Dillmann, S. Schaal, *Robot programming by demonstration*, Vol. 1, Springer, 2008.
- [4] S. Schaal, Is imitation learning the route to humanoid robots?, *Trends in cognitive sciences* 3 (6) (1999) 233–242.
- [5] F. L. Moro, N. G. Tsagarakis, D. G. Caldwell, On the kinematic motion primitives (kmps) - theory and application, *Frontiers in Neurobotics* 6 (10).
- [6] A. Ude, A. Gams, T. Asfour, J. Morimoto, Task-specific generalization of discrete and periodic dynamic movement primitives, *IEEE Transactions on Robotics* 26 (5) (2010) 800–815. doi:10.1109/TRO.2010.2065430.
- [7] A. Ijspeert, J. Nakanishi, S. Schaal, Learning attractor landscapes for learning motor primitives, in: *NIPS*, 2003, pp. 1523–1530.
- [8] A. Lemme, K. Neumann, R. F. Reinhart, J. J. Steil, Neurally imprinted stable vector fields, in: *Proc. Europ. Symp. on Artificial Neural Networks*, 2013, pp. 327–332.
- [9] S. M. Khansari-Zadeh, A. Billard, BM: An iterative algorithm to learn stable non-linear dynamical systems with gaussian mixture models, in: *IEEE Conf. ICRA*, 2010, pp. 2381–2388.
- [10] F. Reinhart, A. Lemme, J. Steil, Representation and generalization of bi-manual skills from kinesthetic teaching, in: *Proc. Humanoids*, 2012, pp. 560–567.
- [11] E. Gribovskaya, A. Billard, Learning nonlinear multi-variate motion dynamics for real-time position and orientation control of robotic manipulators, in: *IEEE Conf. Humanoids*, 2009, pp. 472–477.
- [12] K. Dautenhahn, C. L. Nehaniv, The agent-based perspective on imitation, *Imitation in animals and artifacts* (2002) 1–40.
- [13] S. M. Khansari-Zadeh, A. Billard, Learning stable nonlinear dynamical systems with gaussian mixture models, *IEEE Transactions on Robotics* 27 (5) (2011) 943–957.
- [14] S. M. Khansari-Zadeh, <http://www.amarsi-project.eu/open-source> (2012).
- [15] S. M. Khansari-Zadeh, A dynamical system-based approach to modeling stable robot control policies via imitation learning, Ph.D. thesis, École Polytechnique Fédérale de Lausanne, Switzerland (2012).
- [16] S. M. Khansari-Zadeh, A. Billard, Learning control lyapunov function to ensure stability of dynamical system-based robot reaching motions, *Robotics and Autonomous Systems*.
- [17] Z. Artstein, Stabilization with relaxed controls., *Nonlinear Analysis TMA* 7 (11) (1983) 1163–1173.
- [18] E. D. Sontag, A universal construction of artstein’s theorem on nonlinear stabilization, *Systems & control letters* 13 (2) (1989) 117–123.
- [19] K. Neumann, A. Lemme, J. J. Steil, Neural learning of stable dynamical systems based on data-driven lyapunov candidates, in: *IEEE Proc. Int. Conf. on Intelligent Robots and Systems*, 2013, pp. 1216–1222.
- [20] K. Neumann, M. Rolf, J. J. Steil, Reliable integration of continuous constraints into extreme learning machines, *Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*.
- [21] M. Khansari, A. Lemme, Y. Meirovitch, B. Schrauwen, M. A. Giese, A. Ijspeert, A. Billard, J. J. Steil, Workshop on benchmarking of state-of-the-art algorithms in generating human-like robot reaching motions, in: *Humanoids*, 2013.

- [22] S. M. Khansari-Zadeh,
<http://lisa.epfl.ch/people/member.php?SCIPER=183746/>
(2013).
- [23] N. G. Tsagarakis, G. Metta, G. Sandini, D. Vernon, R. Beira, F. Becchi, L. Righetti, J. Santos-Victor, A. J. Ijspeert, M. C. Carrozza, et al., icub: the design and realization of an open humanoid platform for cognitive and neuroscience research, *Advanced Robotics* 21 (10) (2007) 1151–1175.