Learning High Dimensional Demonstrations Using Laplacian Eigenmaps

Sthithpragya Gupta, Aradhana Nayak, Aude Billard

Abstract—This article proposes a novel methodology to learn a stable robot control law driven by dynamical systems. The methodology requires a single demonstration and can deduce a stable dynamics in arbitrary high dimensions. The method relies on the idea that there exists a latent space in which the nonlinear dynamics appears quasi-linear. The original nonlinear dynamics is mapped into a stable linear DS, by leveraging on the properties of graph embeddings. We show that the eigendecomposition of the Graph Laplacian results in linear embeddings in two dimensions and quasi-linear in higher dimensions. The nonlinear terms vanish, exponentially as the number of datapoints increase, and for large density of points, the embedding appears linear. We show that this new embedding enables to model highly nonlinear dynamics in high dimension and overcomes alternative techniques in both precision of reconstruction and number of parameters required for the embedding. We demonstrate its applicability to control a real robot tasked to perform complex free motion in space.

Index Terms—learning from demonstration (LfD)

I. INTRODUCTION

Robots are often required to execute novel tasks that involve convoluted motions in high dimensions. The traditional methods of robot programming to achieve this may require varying levels of user input that may be infeasible and impractical. In these settings, learning from demonstration (LfD) offers an elegant solution to facilitate robot programming and learning novel trajectories for task completion [1]. The goal is to replace the manual process of programming the robot with an automatic programming process whose input consists of demonstrations performed by an expert. The challenges this paper sets out to address are 1) to learn a stable control law from a single expert demonstration as it is difficult and expensive to generate a large input data set and 2) to stabilize highly nonlinear dynamics in high dimensions.

We set forth that the underlying control law is driven by a dynamical system (DS) and that the demonstration is an instance of one path integral, albeit possibly noisy one, of this DS. If the motion has a target, the DS consists of a vector field with asymptotic stability convergence properties to a single attractor, the target of the motion. If the vector field is properly identified from the single path integral, the DS guarantees generalization of the dynamics to unseen areas of the state space.

The authors are with Learning Algorithms and Systems Laboratory (LASA), EPFL, 1015 Lausanne, Switzerland (sthithapragya.gupta@epfl.ch, aradhana.nayak@epfl.ch, aude.billard@epfl.ch)
This paper proposes a novel methodology to learn the vector field of a stable DS by transforming the original n-dimensional task space to a ‘latent space’ where the dynamics appear linear or quasi linear. The rationale is this latent linear embedding would make it easier to stabilise the DS in original space. To obtain this embedding, we express the DS through a graph representation of points in the demonstration and compute an eigendecomposition of the associated graph Laplacian. We show that a subset of the eigenvectors of the Graph Laplacian describe a space in which the dynamics is linear if 2D and quasi linear for dimension higher than 2. We further show that the non-linearity of the embedding vanish with \( O(1/N^2) \), \( N \) being the number of data points. The dynamics in this embedding follow a Chebyshev polynomial, and hence, we refer to our new embedding as Chebyshev DS in the rest of this paper.

To reconstruct the dynamics in original space, we learn a diffeomorphism between the latent space and the demonstration space, using the fast diffeomorphic matching (FDM) algorithm proposed in [2]. When comparing our embedding to the use of other latent spaces, we obtain better reconstruction of highly nonlinear motions and better generalization of the dynamics when initialized away from the initial demonstration, see example in Figure 1.

The paper is organised as follows. In Section II, we review closely related work and introduce the problem formulation in Section III. Section IV revisits Belkin and Niyogi’s formulation of Laplacian eigenmaps and introduce its application on our use case. The main theoretical results related to the construction of latent space embedding are documented in Section V. Section VI describes the application of the diffeomorphism learning approach to our latent space embedding and choice of hyperparameters. Section VII reports on the quantitative and qualitative comparative evaluations in simulation and robot implementation experiments, that evaluate the efficacy of the proposed latent space. We end the paper by presenting our conclusions in Section IX.

II. BACKGROUND & RELATED WORK

DS-based LfD was first tackled with dynamical motion primitives (DMPs) in [4]. In the classical DMP approach, temporal disturbances are handled by a phase variable, and a demonstration in the joint space is represented by a DMP per joint. Therefore, it is difficult to accurately synchronise the phase across several DMPs. [5] show that even when the DS is stable, the vector field governing the flow of the DS intersects itself as shown in Fig. 2. Furthermore, it is not robust to spatial disturbances during roll-out, as shown in [6].

Basis functions are employed to model the dynamics that encapsulate the nonlinear features of the vector field governing the DS. The basis functions are usually Gaussians ([7], [8], [6]) of appropriate dimension that are parameterised so as to preserve the stability of the DS in accordance to a chosen or optimal Lyapunov function. In the stable estimator of dynamical systems (SEDS) approach ([9]), the Lyapunov function is fixed as the square of the distance to the attractor. In [8], the underlying Lyapunov function is modelled by choosing it from a set of weight sum of asymmetric quadratic functions. In [10], \( \tau - \text{SEDS} \), the demonstration space is transformed by a diffeomorphism so that SEDS can be applied. [11] propose a linear parameter varying DS (LPV-DS) to improve the performance of SEDS using parameterised quadratic Lyapunov functions. [12] and [6] propose the existence of an underlying positive definite contraction metric, and the Gaussian mixture model (GMM) is selected under constraints imposed by the contraction metric. However, only a certain class of contraction metrics is considered.

Fig. 2: Forward orbits generated by the DS learnt using DMP

The limitations of the above approaches are more pronounced as the complexity of the demonstration increases. This is due to an inaccurate estimation of the underlying Lyapunov function that yields a poor replication of the nonlinearity in the demonstration. This can be overcome either by learning the Lyapunov function together with the dynamics with a neural network as in [13] or by learning a diffeomorphism between a latent space and demonstration space. The characteristic feature of the latent space is that the transformed DS in the latent space is either linear or highly simplified. There are two distinct approaches to learning the aforementioned diffeomorphism. The first is a geometric approach that learns from just one demonstration either with FDM [2] or by the large deformation diffeomorphism metric mapping approach in [14]. The latter is hard to invert to the original space, and the trajectories in the latent space are generated by a linear DS of the form \( \dot{x} = -x \) in [2]. While it is true that a diffeomorphism transforms the original DS into a simplified latent space DS, the explicit formulation of the latent space DS is unknown.

In the second approach, this diffeomorphism is learnt from multiple demonstrations such as stable dynamical system learning using Euclideanising flows (SDS-EF) in [3] and its
modification to stochastic systems in [15]. In SDS-EF, the diffeomorphism is expressed using function approximators based on single-layer neural networks, wherein the layer resembles a Gaussian kernel. The formulation relies on the fact that the DS in the latent space is the negative gradient descent of a chosen potential function. As a generative model is learnt, the number of demonstrations required to train the neural network is very high compared to the single diffeomorphism learning approach in [2]. This is why SDS-EF performs poorly when a single demonstration is provided.

In summary, both approaches for diffeomorphism-based reconstruction of stable dynamics rely on an underlying DS in the latent space. In this work, we propose to improve the algorithm in [2] by choosing coordinates in the latent space based on a graph representation of the demonstration.

The approach of using graph Laplacian for dimensionality reduction was introduced in [16]. In our formulation of latent space, we construct an unweighted graph whose nodes represent points in the demonstration and whose edges connect the neighbouring points. The coordinates in latent space correspond to the entries of certain eigenvectors of the Laplacian. We observe that the dynamical system in latent space is quasi-linear by analysing the chosen eigenvector entries. This simplifies the latent space dynamics.

In [17], the problem has been addressed for a demonstration evolving in a 2D space. Therefore, the method is only applicable to learn a demonstration in joint space for a robot with two joints or to learn a demonstration in a 2-dimensional task space. In most common robotic applications, such as pick and place objects, the task space is higher than two dimensions. This paper extends the proposed algorithm in [17] to learn complex demonstrations in high dimensions given a single demonstration for training.

The main contributions of this work:

1) Learning a stable, highly nonlinear autonomous dynamical system (DS) from a single demonstration.
2) Identifying a latent space wherein the dynamics are quasi-linear, easing learning of highly nonlinear dynamics.
3) Proving that the DS in latent space becomes linear for a large number of datapoints \( N \), as the nonlinear terms decrease with \( O(1/N^2) \).
4) Demonstrating that the proposed latent space embedding requires fewer parameters than other approaches with similar accuracy at reconstruction and accounts better for high nonlinearities.

### III. Problem Setup

The provided demonstration consists of \( N \) position–velocity pairs \( \{(y_i', y_i^{\text{v}})\}_{i=1}^N \) wherein the positions are \( y_i^\text{v} \in \mathbb{R}^n \) and the velocities are \( y_i' \in \mathbb{R}^n \). These position–velocity pairs are sampled at a constant frequency from a trajectory that is the path integral of a DS. The DS is defined as follows:

1) The nonlinear function \( f : \mathbb{R}^n \to \mathbb{R}^n \) describing the DS maps the \( n \)-dimensional position vector \( y_i^\text{v} \) to the corresponding velocity vector \( y_i' \) by the ordinary differential equation (ODE)

\[
\dot{y} = f(y), \quad y \in \mathbb{R}^n \tag{1}
\]

2) The DS is globally asymptotically stable at \( y^* \in \mathbb{R}^n \), and, therefore, \( \lim_{t \to \infty} y(t) = y^* \) for all path integrals \( y(t) \) of the DS.

Consider a diffeomorphism \( \psi : \mathbb{R}^n \to \mathbb{R}^n \) that transforms the demonstration space into a latent space. The coordinates in the latent space are defined as \( x := \psi(y) \). The time-parametrised path integrals \( y(t) \) of the DS in (1) transformed by \( \psi \) are denoted as \( x(t) \) and defined as \( x(t) := \psi(y(t)) \). These path integrals are solutions to the following ODE describing the DS in the latent space:

\[
\dot{x} = g(x) := \frac{\partial \psi}{\partial y} (\psi^{-1}(x) f(y^{-1}(x))) \tag{2}
\]

The DS in (2) is also globally and asymptotically stable at \( x^* := \psi(y^*) \) as

\[
x^* = \psi \left[ \lim_{t \to \infty} y(t) \right] = \lim_{t \to \infty} \psi(y(t)) = \lim_{t \to \infty} x(t)
\]

Therefore, one can recover the DS in (1), given the DS in the latent space in (2) by the following transformation:

\[
\dot{y} = \frac{\partial \psi^{-1}}{\partial x} g(\psi(y)) \tag{3}
\]

Hence, the DS of latent space in (2) and the diffeomorphism \( \psi \) are sufficient to reconstruct the dynamics of the DS in (1). The main focus of this work is to construct an accurate and stable latent space DS. Then, state-of-the-art methods are applied to learn \( \psi \), and the DS in the original space is recovered by the transformation in (3).

In the following section, we proceed by obtaining a stable latent space DS from the observed points \( \{(y_i^\text{v}, y_i')\} \). In addition to being stable, we also show that the DS is quasi-linear. The coordinates of the latent space are given by \( n \) carefully chosen eigenvectors of a graph Laplacian. In the next section, we formulate the graph from the observed points following the approach in [16]. We make specific assumptions on the structure of the graph that ensure desired properties of stability and quasi-linearity in the DS in the latent space.

### IV. Generating a Graph from a Single Demonstration

Consider \( K \) copies of the single given demonstration. The graph is denoted by \( G \) and has \( N \times K \) nodes. A node of the graph is denoted as \( v_i^j \), \( i \in \{1, \ldots, N\} \), \( j \in \{1, \ldots, K\} \) and corresponds to the \( i \)-th position–velocity pair of the demonstration in the \( j \)-th copy of the demonstration set. The edges are given as node pairs \( \{(v_i^1, v_i^k)\}_{i=1}^N \) and \( k = 1, \ldots, K \). The edge weight is unity. The nodes labeled as

\[
\{v_1^1, \ldots, v_N^1, v_1^2, \ldots, v_N^2, \ldots, v_1^K, \ldots, v_N^K\} \sim \{1, 2, \ldots, N \times K\}
\]

The \( K \) nodes in the set \( \{v_1^1, \ldots, v_N^K\} \) form a cyclic graph (or simple circuit). Fig. 3(c) is the graph representation of
Fig. 3: Framework of the procedure used to prepare the latent space and learn the diffeomorphic map. (a) An unstable 3D spiral demonstration of complexity $c = 7$ (Section VII); (b) Assuming four copies of the 3-dimensional demonstration (Section IV); (c) Graph representation of demonstrations (Section IV); (d) Distribution of points in latent space corresponding to a single demonstration (Section 5); (e) Latent space points scaled and translated between the start and end points of the demonstration; (f) Mapped points after learning the forward mapping (Section VI)

a demonstration set with $N$ observations and $K = 4$ copies of the demonstration. The $N$th observation of each copy is represented by the node $\nu^k_N$ and corresponds to the last point of the demonstration. This last point is deemed to be at, or close to, the attractor $y^*$ of the DS in (1). By construction, $G$ preserves the local connectivity across the points conveyed in the demonstration. The edge weights for $G$ are formulated as:

$$\text{edge}\{\nu^k_i, \nu^l_j\} = \begin{cases} 1 \text{ if } k = l \& \|i - j\| = 1 \text{ or } i = j = N \\ 0 \text{ otherwise} \end{cases}$$

The eigenvalues and eigenvectors are defined as solutions to the generalised eigenvector problem,

$$L(G)u^l = \lambda u^l, \quad l = 1, \ldots, N \times K, \quad u^l \in \mathbb{R}^{N \times K}$$

where the graph Laplacian is denoted by $L(G)$ and preserves the neighbourhood information of the demonstration set. It has been used extensively in manifold learning for extracting latent representations of data in high dimensions and in spectral clustering.

In this section, we study the entries of eigenvectors of $L(G)$ and show the existence of $n$ eigenvectors that form the basis of the latent space. In this latent space, the embedded dynamics are both stable and quasi-linear. In Subsection IV-A the eigenvector entries are explicitly computed from the eigenequation (1) and are shown to be polynomial functions of the eigenvalues. In Subsection IV-B we show the existence of at least $n$ eigenvalues in the spectrum of $L(G)$, which, if distinct, differ from each other by a term in $O(1/N^2)$. We conclude the section by showing that the entries of eigenvectors corresponding to these $n$ eigenvalues are quasi-linear.

A. Analysis of Laplacian

We make a few observations on the structure of $L(G)$ to simplify the study of its eigenvectors. By definition, $L(G) = D(G) - A(G)$, where the matrices $D(G)$ and $A(G)$ denote the degree and adjacency matrices of $G$. $L(G)$ can be expressed in terms of the block circulant matrix $J$ as follows:

$$L(G) = 2I_{N \times K} - J,$$

$$J = \text{blockcirc}(B_1, B_2, \text{zeros}(N), \ldots, \text{zeros}(N), B_2)$$

where $B_1$ and $B_2$ are block matrices defined as:

$$B_1 = \begin{pmatrix} 1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 1 \\ 0 & 0 & 0 & \cdots & 1 & -1 \end{pmatrix} \in \mathbb{R}^{N \times N},$$

$$B_2 = \begin{pmatrix} \text{zeros}(N - 1) & 0 \\ 0 & 1 \end{pmatrix} \in \mathbb{R}^{N \times N}.$$
where \( bcirc \) denotes a block circular matrix defined as
\[
bcirc(B_1, B_2, \ldots, B_L) := \begin{pmatrix} B_1 & B_2 & \cdots & B_L \\ B_L & B_1 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ B_2 & B_L & \cdots & B_1 \end{pmatrix},
\]
for \( L \geq 3 \), and \( zeros(N) \) denoting an \( N \times N \) matrix of zeros, \( \mathbf{0} \) denoting an \( N \) vector of zeros.

The block circulant structure inside the \( L(G) \) allows us to identify many eigenvalues in the spectrum of \( L(G) \) that repeat with algebraic multiplicity equal to 2. The following proposition gives the exact number of such repeating eigenvalues.

**Proposition 1.** Denote the number of eigenvalues of \( L(G) \) having an algebraic multiplicity of 2 by \( \Lambda \). We have
\[
\Lambda = \begin{cases} \frac{K}{2} - 1 & \text{if } K = 2j, \quad j \in \{1, 2, \ldots\} \\ \frac{K-1}{2} & \text{if } K = 2j - 1, \quad j \in \{2, 3, \ldots\} \end{cases}
\]

(6)

**Proof.** From Section 3.1 of [18], we observe that the eigenvalues and corresponding eigenvectors of \( J \) are determined by the following \( K \) equations, each giving \( N \) eigenvalues and vectors.
\[
H_j v = \lambda_j v, \quad j \in \{0, \ldots, K-1\}
\]
\[
H_j := B_1 + B_2 (\rho_j + \rho_j^{K-1})
\]
where \( \rho_j \) is the \( n \)-th root of 1 given by
\[
\rho_j = \exp\left(\frac{2\pi j}{K}\right), \quad j \in \{0, \ldots, K-1\}
\]
Observe that \( \rho_j^{(K-1)} = \rho_j^{-1} \) for all \( j \). Further,
\[
\rho_j + \rho_j^{-1} = 2 \cos\left(\frac{2\pi j}{K}\right) = 2 \cos\left(\frac{2\pi (K-j)}{K}\right)
\]
\[
= \rho_{K-j} + \rho_{K-j}^{-1}, \quad j \in \{1, \ldots, K-1\}
\]
Because eigenvalues and vectors of \( L(G) \) and \( J \) are identical, the conclusion follows from (7). \( \square \)

Each eigenvector \( u^l \) of \( L(G) \) possesses a peculiar structure, as observed by the following lemma.

**Lemma 1.** Denote the first \( N \) entries of the eigenvector \( u^l \) by \( u^l \in \mathbb{R}^N \). Then we have
\[
u^l = (u^l \quad pu^l \quad \rho^2 u^l \quad \cdots \quad \rho^{K-1} u^l)^T
\]
(8)

where \( \rho \) is the \( K \)th root of unity.

**Proof.** The result follows from the fact that eigenvectors of \( L(G) \) and \( J \) are identical. The eigenvectors of the block circulant matrix \( J \) with \( K \) blocks, each of size \( N \), are given by (8) following the result from Section 3.1 in [18]. \( \square \)

For simplicity in notation, we henceforth denote \( u^l \) and \( u^l \) as \( u \) and \( u \), respectively. The following proposition shows that the first \( N \) entries of \( u \) can be expressed as polynomial functions of their corresponding eigenvalues.

**Proposition 2.** The entries \( u_i \) of \( u = [u_1, \ldots, u_N] \) corresponding to the eigenvalue \( \lambda \) are expressed as combination of Chebyshev polynomials \( T_i \) and \( V_i \) as
\[
u_i(\lambda) = u_1 \left[T_i \left(1 - \frac{\lambda}{2}\right) - \frac{\lambda}{2} V_{i-1} \left(1 - \frac{\lambda}{2}\right) \right]
\]
(9)

\( T_i \) and \( V_i \) are Chebyshev polynomials of first and second kind defined as
\[
T_i(\theta) = \cos(i \arccos(\theta))
\]
(10)
\[
V_i(\theta) = \frac{\sin((i + 1) \arccos(\theta))}{\sin(\arccos(\theta))}
\]
(11)

**Proof.** From the eigenequation (5), we have
\[
u_2 = (1 - \lambda)u_1
\]
(12)
\[
u_i = (2 - \lambda)u_{i-1} - u_{i-2}, \quad i = 3, \ldots, N - 1
\]
The entries \( u_i \) of any vector of \( L(G) \) can be modelled as a combination of Chebyshev polynomials of the first and second kind that are defined as \( T(\theta) \) and \( V(\theta) \), respectively, as follows:
\[
T_0(\theta) = 1
\]
(13)
\[
T_1(\theta) = \theta
\]
\[
T_i(\theta) = 2\theta T_{i-1}(\theta) - T_{i-2}(\theta)
\]

which yields \( T_i(\theta) \) in (10). Similarly,
\[
V_0(\theta) = 1
\]
(14)
\[
V_1(\theta) = 2\theta
\]
\[
V_i(\theta) = 2\theta V_{i-1}(\theta) - V_{i-2}(\theta)
\]

which yields \( V_i(\theta) \) in (11). Choosing \( \theta = (1 - \lambda/2) \) in both (13) and (14), we obtain the recursion in (12) with the combination proposed in (9). \( \square \)

**B. Latent space arising from Laplacian eigenmaps**

The set of basis vectors of the latent space is the set of first \( N \) components of eigenvectors corresponding to the \( n \) smallest non-zero repeating eigenvalues of \( L(G) \) for \( G \) with \( K = n + 1 \) paths. These particular eigenvector components are chosen as they have monotonically increasing or decreasing entries, as shown in the following lemma. We shall soon see that this property also proves the stability of the DS in the latent space. We shall also see the need for choosing \( K = n + 1 \).

**Lemma 2.** Consider the eigenvector \( u \) corresponding to the eigenvalue \( \lambda \) s.t.
\[
0 < \lambda \leq 2 \left[1 - \cos\left(\frac{\pi}{N-\frac{1}{2}}\right)\right]
\]
(15)
The entries \( u_i \) of \( u = [u_1, \ldots, u_N] \), which are the first \( N \) components of the eigenvector \( u \), either increase or decrease monotonically s.t.
\[
u_j \geq (\leq)u_{j+1}, \quad j = \{1, \ldots, N\}
\]

**Proof.** In Appendix A \( \square \)
Because the sequence \( \{u_j\}_{j=1}^N \) is bounded, by the monotone convergence theorem, the sequence converges to the lower bound if \( u_j \geq u_{j+1} \) and to the upper bound if \( u_j \leq u_{j+1} \). This shows that the DS in the latent space resulting from the first \( N \) entries of \( u \) is stable.

Lemma 2 holds for eigenvectors corresponding to eigenvalues satisfying a certain upper bound. In the following lemma, we show that such eigenvalues always exist in the spectrum of \( L(G) \).

**Lemma 3.** The smallest repeating eigenvalue \( K - 1 \) eigenvalues of \( L(G) \), denoted \( \lambda_j \), \( j = 1...K - 1 \), have as upper bound:

\[
\lambda_j(L) < 2 \left( 1 - \cos \left( \frac{\pi}{N} \right) \right), \quad j = \{1,...,(K-1)\}
\]

*Proof.* In Appendix B

From Lemma 3 in order for us to have \( n \) eigenvalues satisfying (16), we choose the number of paths \( K = n + 1 \) in \( G \). The first \( N \) components of the eigenvectors corresponding to these \( n \) eigenvalues are monotonic by Lemma 2. Choosing these vectors as basis vectors of the latent space, we obtain a *stable* representation of the demonstration in the latent space.

Next we tackle the issue of *quasi linearity* of the embedded demonstration in the latent space. Observe that the upper bound in (16) belongs to \( O(1/N^2) \) by considering a power series expansion of the cosine function in (16):

\[
\lambda_j(L) < 2 \left( \left( \frac{\pi}{N} \right)^2 \frac{1}{2!} - \left( \frac{\pi}{N} \right)^4 \frac{1}{4!} + \ldots \right) \in O(1/N^2)
\]

with \( j = \{1,...,n\} \) for a graph with \( K = n + 1 \). Therefore, the upper bound holds for at least \( n \) eigenvalues. In the following proposition, we show that the eigenvectors corresponding to these eigenvalues also differ entry-wise by a small term that belongs to \( O(1/N^2) \). These eigenvectors form the basis of the \( n \)-dimensional latent space.

**Proposition 3.** Consider the graph \( G \) constructed with \( K = n + 1 \) copies of the demonstration. The vector space given by the first \( N \) components of \( n \) eigenvectors corresponding to the smallest \( n \) repeating eigenvalues of \( L(G) \) is the latent space. In this space, the graph embedding of the demonstration is linear up to an order \( O(1/N^2) \). In particular, \( \|u^i - u^m\|^2 \in O(1/N^2) \) for \( u^i \) and \( u^m \) (in (1)) corresponding to \( u^i \) and \( u^m \) (in (5)) respectively for repeating eigenvalues of \( L(G) \).

*Proof.* In Appendix C

**Remark 1.** Observe from (9) that the eigenvectors corresponding to eigenvalues repeating with algebraic multiplicity equal to 2 differ only in the scaling factor \( u_0 \). Given a \( 2D \) demonstration, the latent space is composed of the first \( N \) components of the eigenvector corresponding to the smallest repeating eigenvalue. Therefore, we obtain a linear embedding of the dynamics in the latent space. This result appears in (17).

Note that the construction of the graph (and hence the latent space) does not require the position–velocity values from the supplied demonstration. Because \( G \) assumes a binary structure, only \( N \) (length of the demonstration) and \( n \) (dimensions of demonstration space) are required.

V. FORMULATION OF LATENT SPACE

We denote the latent representation of our data set in the embedding as \( X := \{x^1, x^2, \ldots, x^N\} \) and drop the scaling term \( u_0 \) in (9) for simplicity. Each \( x^i \) is \( n \)-dimensional.

\[
x^i = (x_{i,1}, x_{i,2}, \ldots, x_{i,N})^T, \quad i = \{1, \ldots, N\}
\]

\[
x_{i,j} = T_i(1 - \frac{\lambda_j}{2}) - \frac{\lambda_j}{2} V_{i-1} (1 - \frac{\lambda_j}{2}), \quad j = \{1, \ldots, n\}
\]

with \( T_i \) and \( V_i \) being defined in Equations (10) and (11), respectively. From Proposition 3 we know that \( x^i \) are quasi co-linear, and the interpoint spacing between successive points increases monotonically. As previously outlined, the latent space is prepared by considering \( K \) copies of the demonstration. The formulation from (18) can further be simplified as:

\[
x_{i,j} = \cos(i \arccos(1 - \frac{\lambda_j}{2})) - \frac{\lambda_j}{2} \sin(i \arccos(1 - \frac{\lambda_j}{2}))
\]

\[
= \cos(i a_j) - b \sin(i a_j) = \sqrt{b_j^2 + 1} \sin(\gamma - i a_j)
\]

where \( a_j = \arccos(1 - \frac{\lambda_j}{2}), \quad b_j = \frac{\lambda_j}{2 \sin(\gamma)} \) and \( \gamma_j = \arcsin(\frac{1}{\sqrt{b_j^2 + 1}}) \). It follows from the above formulation that

\[
x^1 = 1_n - \lambda
\]

\[
x^N = \sqrt{b^2 + 1} \sin(\gamma - N a)
\]

where \( \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)^T, \quad a = (a_1, a_2, \ldots, a_n)^T, \quad b = (b_1, b_2, \ldots, b_n)^T \) and \( \gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)^T \). To model the input for learning the diffeomorphism, the order of the points in \( X \) is reversed. With a slight abuse of notation, we refer to this reversed set also as \( X \in \mathbb{R}^{N \times n} \). \( X \) hereby represents a latent space embedding of the system that starts at \( x_0 \) (formerly \( x^N \)) and terminates at \( x^* \) (formerly \( x^1 \)), as shown in Figure 3 (c).

VI. LEARNING THE Diffeomorphism

We propose to follow the FDM approach in (2) to learn the diffeomorphism \( \psi : \mathbb{R}^n \rightarrow \mathbb{R}^n \) between the latent space prepared in Section V and the demonstration space such that \( \psi(x^i) = y^i \) for every \( i \in \{1, \ldots, N\} \). We collectively denote points in the latent space by \( X := \{x^1, \ldots, x^N\} \) and points in the demonstration space by \( Y := \{y^1, \ldots, y^N\} \). The diffeomorphism is expressed as a composition of \( M \) individual diffeomorphisms as \( \psi = \psi_M \circ \ldots \circ \psi_1 \), wherein each \( \psi_j : \mathbb{R}^n \rightarrow \mathbb{R}^n \) is a diffeomorphism in the layer \( j \) of the algorithm.
A. Hyperparameter tuning

The hyperparameters associated with learning the diffeomorphism in accordance with the FDM approach are $\mu$, $\beta$ and $M$:

- $M$: Number of diffeomorphisms constituting the forward map $\psi$. A lower $M$ value is favourable as it lowers the chance of overfitting and the computational complexity associated with inverting the learnt forward map.
- $\mu \in (0, 1)$: Constrains each diffeomorphism to remain invertible.
- $\beta \in (0, 1)$: Behaves like the learning rate while learning the forward map. A lower value is favourable for stable convergence and reduced chance of overfitting.

The metric used to measure the relevance of the hyperparameter combinations ($\mu$, $\beta$, $M$) was the mean-squared error (MSE) between the demonstration $Y$ and the final deformed input space $\psi(X)$. The combination with the least number of successive diffeomorphisms ($M$) was selected from amongst the hyper-parameter combinations ($\mu$, $\beta$, $M$), producing a normalised MSE less than $1e^{-5}$ (unless having slightly more diffeomorphisms significantly reduced MSE). Amongst the combinations with the same number of successive diffeomorphisms, there are two possible candidates – the combination with the least $\beta$ value and the combination with the lowest MSE. If none of the combinations could produce an MSE lower than $1e^{-5}$, the combination with the lowest MSE was selected.

The hyperparameters for learning the diffeomorphism associated with Euclideanising flows are $L$ and $n_f$:

- $L$: The number of layers constituting the network. A lower $L$ value is favourable as it lowers the chance of overfitting.
- $n_f$: The number of features per layer. Once again, a lower value is favourable to a lower chance of overfitting.

The demonstration data were split into train and test sets, with the training set constituting $70\%$ of the data. The metric used to measure the relevance of the hyperparameter combinations ($L$, $n_f$) was the MSE on the test set.

VII. Results

The performance of the proposed Chebyshev-based formulation of the latent space [20] for learning a DS using successive diffeomorphisms is presented in this section. This framework has been compared with the original FDM approach and the SDS-EF approach. The efficacy of using the proposed latent space has been evaluated by generating the forward orbits of the learnt dynamical systems from the original starting points. These forward orbits were compared with their respective original demonstrations. The replication generated by the learnt DS differed from the original demonstration in terms of speed, path traced and number of time-stamped points. Consequently, generally used methods such as the $L2$-norm could not be employed to measure the similarity between the demonstration and the replication. However, because both the replication and demonstration are temporal sequences, we employed FastDTW [19], a variant of dynamic time warping, to measure similarity. Lower scores are favourable.

The following spiral trajectories (of unit radius and varying complexities) were used as demonstrations to evaluate and compare the performance of the different methods:

1) Five 3-dimensional spherical spirals from an unstable DS
2) Five 3-dimensional spherical spirals from a stable DS

The formulation of the 3-dimensional spirals from an unstable DS (hereby referred to as unstable 3D spirals) and those from a stable DS (hereby referred to as stable 3D spirals) can be found in Appendix [2]. Test demonstrations were formulated using $c \in \{1, 3, 7, 10, 15\}$ for the unstable DS and $c \in \{1, 2, 3, 5, 7\}$ for the stable DS. From each of the forward orbits, $N = 500$ points were uniformly (temporally) sampled to be used for performance evaluation. As stated before, the value of $c$ regulates the complexity of the spiral. A spiral with higher $c$ makes the learning and replication tasks more challenging.

The tuned hyperparameters utilised by the presented methods for learning the maps for unstable 3D spirals are presented in Tables [1] & [2].

To generate the forward orbits of the learnt dynamical systems, 20 other starting points were randomly picked (consistent across different methods and spirals) in a vicinity of $0.1m$ around the original starting point $(0, 0, 1)$. The forward orbits generated for spiral cases $c = 7$ (for unstable 3D spirals) and $c = 3$ (for stable 3D spirals) are presented in Figs [1a] and [6].

Each replication was compared to the original demonstration. The mean normalised DTW scores with their standard deviations are presented in Fig. [5a] where lower scores are favourable. Of the different methods, SDS-EF performs the best, closely followed by the proposed Chebyshev-
Fig. 5: Case-wise normalised similarity scores of replications generated from DS learnt using different methods – proposed Chebyshev approach, FDM and SDS-EF – for spirals of varying complexity. (left) Score computed using fast dynamic time warping (DTW) [19] – a lower score is better; (right) $p$-values from the t-test conducted on the DTW scores of replications generated from DS learnt using the proposed Chebyshev approach and FDM for spirals of varying complexity.

Fig. 6: Replications generated by the DS learnt using different methods for stable 3D spirals ($c = 3$)
TABLE I: Hyperparameters associated with the proposed Chebyshev-based method and FDM for learning to replicate unstable 3D spirals

| c | Method | β  | µ  | M | MSE      |
|---|--------|----|----|---|----------|
| 1 | Chebyshev | 0.90 | 0.6 | 50 | 7.38E-07 |
| 3 | Chebyshev | 0.90 | 0.8 | 75 | 5.73E-07 |
| 7 | FDM | 0.90 | 0.9 | 175 | 9.64E-07 |
| 10 | Chebyshev | 0.50 | 0.9 | 300 | 3.57E-06 |
| 15 | FDM | 0.50 | 0.9 | 750 | 1.56E-05 |

TABLE II: Hyperparameters associated with the SDS-EF method for learning to replicate unstable 3D spirals

| c | L | n_f | MSE |
|---|---|-----|-----|
| 1 | 10 | 250 | 1.18E-02 |
| 3 | 10 | 150 | 1.99E-02 |
| 7 | 30 | 350 | 5.66E-02 |
| 10 | 40 | 200 | 7.26E-02 |
| 15 | 40 | 300 | 1.32E-01 |

Fig. 7: Replications generated by the DS learnt using the proposed Chebyshev-based method (first column) and the original FDM method (third column) for additional demonstrations; Vector field corresponding to the DS learnt using the Chebyshev-based method (second column) and the original FDM method (fourth column)

TABLE I: Hyperparameters associated with the proposed Chebyshev-based method and FDM for learning to replicate unstable 3D spirals

TABLE II: Hyperparameters associated with the SDS-EF method for learning to replicate unstable 3D spirals

VIII. ROBOT IMPLEMENTATION

Eight demonstrations of varying dimensionality and complexity were provided to the robot as part of the robot implementation. These were as follows:

- In the third case, the DS learnt using FDM converges at an incorrect attractor, consequently producing erroneous replication.
- In the fourth case, the DS learnt using FDM cannot capture the initial swirl of the spiral.
The similarity scores computed using FastDTW are presented in Table III. The similarity scores between the kinesthetic demonstration and the corresponding replication. A lower score is better.

| Case | a  | b  | c  | d  | e  | f  |
|------|----|----|----|----|----|----|
| Score| 25.92 | 19.28 | 81.25 | 9.32 | 9.74 | 18.31 |

- Numerically generated demonstrations (2):
  - 2-dimensional task-space data corresponding to the end-effector’s motion on an Archimedean 2D spiral with a radius of 15 cm (similar to [66])
  - 3-dimensional task-space data corresponding to the end-effector’s motion on an unstable 3D spiral (c = 7) with a radius of 20 cm

- Kinesthetically recorded demonstrations (6): 7-dimensional joint position commanded to the robot yielded by the learnt DS was supplemented with a constant 3-dimensional orientation vector. The 6-dimensional end-effector pose was commanded to the robot with a constant 3-dimensional orientation vector. The 6-dimensional end-effector pose was commanded to the robot with a constant 3-dimensional orientation vector.

IX. Conclusion

In this article, a latent space representation of a single demonstration was obtained. The representation was quasi-linear, given a demonstration of a 2-dimensional or higher Euclidean space. The theoretical findings were verified with numerical experiments. Lower DTW scores implying higher replication accuracy were obtained for several demonstrations in which the proposed latent space was used instead of the commonly chosen Hurwitz matrices to describe linear dynamics in the latent space.

However, the proposed latent space does not affect the learnt diffeomorphism. In the present work, a state-of-the-art method of diffeomorphic point matching (FDM) is employed to learn the diffeomorphism. In the future, we would like to incorporate the knowledge of the latent space to simplify the learning of the diffeomorphism.

APPENDIX A

Proof of Lemma 2

Following Proposition 2, the eigenvector entries $u_i$ of the eigenvector $u = [u_1, \ldots, u_N]$ are

$$u_i(\lambda) = u_0 \left[ T_i(1 - \frac{\lambda}{2}) - \frac{\lambda}{2} V_{i-1} (1 - \frac{\lambda}{2}) \right]$$

for an arbitrary $u_0 \in \mathbb{R}$ and with $T_i$ and $V_i$ defined as

$$T_i(\theta) = \cos(i \arccos(\theta)), \quad V_i(\theta) = \frac{\sin((i + 1) \arccos(\theta))}{\sin(\arccos(\theta))}$$

To study of the stationary points of Eq. 9 we differentiate Eq. 9 with respect to the index $i$:

$$\frac{\partial u_i}{\partial i} = u_0 \left[ -\theta \sin((i-1)\theta) - \theta \frac{\lambda \cos((i-1)\theta)}{\sin(\theta)} \right],$$

$$\theta := \cos^{-1}(1 - \frac{\lambda}{2})$$

Therefore, the series $\{u_i\}$ is either monotonically increasing or monotonically decreasing for $\lambda$ and $i$ such that

$$\sin((i-1)\theta) + \frac{\lambda \cos((i-1)\theta)}{2 \sin(\theta)} = 0$$

Let the eigenvalue $\lambda \in [0, 4]$ be expressed as

$$\lambda = 2 \left( 1 - \cos(\theta - 2\pi j) \right), \quad j \in \mathbb{N}.$$  

Replacing in (25), we obtain

$$\tan((i-1)\theta) + \frac{\theta}{2} = \frac{\pi j}{\cos^{-1}(1 - \frac{\lambda}{2})} + \frac{1}{2} $$

As (28) must hold for $i = N$, $\lambda$ should satisfy (15)
Fig. 9: Cases (a) through (f): Pairs of light painting of the kinesthetically supplied demonstrations and comparison between the Cartesian position of the end-effector during the demonstration (red) and the replication (blue) produced by the learnt DS

**APPENDIX B**

**Proof of Lemma 3**

Define

\[
B_i = B_1 + \alpha(i)B_2, \quad \alpha(i) = 2 \cos \left( \frac{2\pi i}{K} \right), \tag{29}
\]

\[
i = \begin{cases} 
1, 2, \ldots, \frac{K-1}{2} & \text{if } K \text{ is odd} \\
1, 2, \ldots, \frac{K}{2} - 1 & \text{if } K \text{ is even}
\end{cases}
\]

Observe that the smallest repeating eigenvalue of \(L(G)\) is the largest repeating eigenvalue of \(J\). From Proposition 1, this is the largest eigenvalue in one of the \(B_i\) matrices defined in \(29\). Define

\[
B_i = I + M_i, \tag{30}
\]

\[
M_i = \begin{pmatrix} 
0 & 1 & 0 & \cdots & 0 \\
1 & -1 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & \alpha(i) - 2
\end{pmatrix} = M_1(i) + M_2, \tag{31}
\]

with \(\alpha(i)\) defined in \(29\). The matrices \(M_1(i)\) and \(M_2\) are defined as

\[
M_1(i) = \begin{pmatrix} 
M_0 & \nu \\
\nu^T & \alpha(i) - 2
\end{pmatrix}, \quad M_2 = \text{diag}(0, -1, \cdots, -1, 0)
\]
where, and
\[
M_0 = \begin{pmatrix}
0 & 1 & \cdots & 0 & 0 \\
1 & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & 1 \\
0 & 0 & \cdots & 1 & 0 \\
\end{pmatrix} \in \mathbb{R}^{(N-1) \times (N-1)}, \quad v = \begin{pmatrix}
0 \\
\vdots \\
0 \\
1 \\
\end{pmatrix}
\]

From Theorem 3.7 in [21], because \(M_0\) is the adjacent matrix of a path graph with \(N - 1\) vertices, and therefore,
\[
Eig(M_0) = 2\cos\left(\frac{\pi j}{N}\right), \quad j = 1, \ldots, N - 1.
\]

Let \(\lambda_N\) and \(\lambda_1\) denote the largest and smallest eigenvalues of an \(N \times N\) matrix. From Theorem 4.3.17 in [22], the largest eigenvalue of each \(M_1(i)\) matrix is bounded below by
\[
2\cos\left(\frac{\pi}{N}\right) = \lambda_{N-1}(M_0) \leq \lambda_N(M_1(i)), \quad \text{for all } i
\]

From Corollary 4.3.15 in [22], the largest eigenvalue of all \(M_i\) matrices is bounded above and below by
\[
2\cos\left(\frac{\pi}{N}\right) - 1 = \lambda_N(M_1(i)) + \lambda_1(M_2) < \lambda_N(M_i) < 2\cos\left(\frac{\pi}{N}\right) \quad \text{for all } i
\]

The inequality is strict as \(M_1(i)\) and \(M_2\) do not have a common eigenvector. From (30),
\[
Eig(L(G)) = 1 - Eig(M_i), \quad i = \begin{cases}
1, 2, \ldots, \frac{K - 1}{2} & \text{K odd} \\
1, 2, \ldots, \frac{K}{2} - 1 & \text{K even}
\end{cases}
\]

As the eigenvalues of all \(M_i\) repeat with multiplicity equal to 2, therefore, from (32), the smallest \(K - 1\) repeating eigenvalues of \(L\) (assuming \(K\) is odd) are upper bounded as given by (16).

APPENDIX C

Proof of Proposition 3

The \(n\) smallest repeating eigenvalues belong to the order \(O(1/N^2)\) from (17). The eigenvectors of \(L(G)\) under consideration are same as those of \(B_i\) and satisfy its eigenequation, therefore,
\[
B_1 u_i + \alpha(i) B_2 u_i = \lambda_i u_i, \quad \alpha(i) = 2 \cos\left(\frac{2\pi i}{n + 1}\right),
\]

\[i = \begin{cases}
1, 2, \ldots, \frac{n}{2} & n \text{ is even} \\
1, 2, \ldots, \frac{n - 1}{2} & n \text{ is odd}
\end{cases}
\]

We suppress the dependency on \(i\) for simplicity and denote \(u_i = [u^1_i, \ldots, u^N_i]\). Consider \(\|u_i^1\|_2 \in O(1)\) to be fixed for all \(i\). Expanding the eigenequation (33) we obtain
\[
\begin{align*}
\lambda_2 u_i = & \lambda_i u_i^1 - u_i = -u_i + O\left(1/N^2\right) \\
\lambda_2 u_i^2 = & \lambda_i u_i^2 - u_i = -u_i + O\left(1/N^2\right) \\
\lambda_2 u_i^3 = & \lambda_i u_i^3 - u_i = 2u_i + O\left(1/N^2\right) \\
\vdots
\end{align*}
\]

\[
u_i^N = \frac{1}{1 - \lambda_1} u_{i-1} = \pm u_i + O\left(1/N^2\right)
\]

This shows that \(\|u^i - u^m\|_2 \in O\left(1/N^2\right)\), wherein \(u^i\) and \(u^m\) denote the first \(N\) components of two arbitrary vectors in the set of eigenvectors of \(L(G)\) corresponding to the smallest \(n\) repeating eigenvalues.

APPENDIX D

Unstable 3D spirals
\[
x_t = \sin(t) \cos(ct) \\
y_t = \sin(t) \sin(ct) \quad (34) \\
z_t = \cos(t)
\]

for \(t \in [0, 3.14]\). The constant term \(c\) regulates the complexity of the spiral.

Stable 3D spirals
\[
x_t = \sin(\theta) \cos(\psi) \\
y_t = \sin(\theta) \sin(\psi) \\
z_t = \cos(\theta) \\
\dot{\theta}_t = 0.3(\pi - \theta_t) \\
\dot{\psi}_t = 0.3(2\pi - \psi_t) \\
\theta_{t+1} = \theta_t + 0.003\theta_t \\
\psi_{t+1} = \psi_t + 0.003\psi_t
\]

For each \(c\), trajectories were generated until the distance between subsequent points fell below \(1e - 2.2\).

2-dimensional Archimedean Spiral
\[
\begin{align*}
\theta_{goal} &= 3\pi \\
R &= 0.1m \text{ (initial radius of spiral)} \\
\dot{\theta} &= \frac{\theta_{goal}}{T} \\
x_t &= r_t \cos(\theta_t) \\
y_t &= r_t \sin(\theta_t) \\
\theta_{t+1} &= \theta(t + 1) \\
r_{t+1} &= R \frac{\theta_{t+1}}{\theta_{goal}}
\end{align*}
\]

for \(t \in [0, 12]\).

REFERENCES

[1] Harish Ravichandar et al. "Recent Advances in Robot Learning from Demonstration". In: Annual Review of Control, Robotics, and Autonomous Systems 3.1 (May 2020), pp. 297–330. ISSN: 2573-5144, 2573-5144. DOI: 10.1146/annualrev-control-100819-063206
[2] Nicolas Perrin and Philipp Schlehuber-Caissier. “Fast diffeomorphic matching to learn globally asymptotically stable nonlinear dynamical systems”. en. In: Systems & Control Letters 96 (Oct. 2016), pp. 51–59. ISSN: 01676911. DOI: 10.1016/j.sysconle.2016.06.018

[3] Muhammad Asif Rana et al. “Euclideanizing flows: Diffeomorphic reduction for learning stable dynamical systems”. In: Learning for Dynamics and Control. PMLR. 2020, pp. 630–639.

[4] Stefan Schaal. “Is imitation learning the route to humanoid robots?” en. In: Trends in Cognitive Sciences 3.6 (June 1999), pp. 233–242. ISSN: 13646613. DOI: 10.1016/S1364-6613(99)01327-3.

[5] Akshara Rai et al. “Learning coupling terms for obstacle avoidance”. In: 2014 IEEE-RAS International Conference on Humanoid Robots. IEEE. 2014, pp. 512–518.

[6] Harish Ravichandar, Iman Salehi, and Ashwin Dani. “Learning partially contracting dynamical systems from demonstrations”. In: Conference on Robot Learning. PMLR. 2017, pp. 369–378.

[7] S. Mohammad Khansari-Zadeh and Aude Billard. “Learning Stable Nonlinear Dynamical Systems With Gaussian Mixture Models”, en. In: IEEE Transactions on Robotics 27.5 (Oct. 2011), pp. 943–957. ISSN: 1552-3098, 1941-0468. DOI: 10.1109/TRO.2011.2159412.

[8] S. Mohammad Khansari-Zadeh and Aude Billard. “Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions”. en. In: Robotics and Autonomous Systems 62.6 (June 2014), pp. 752–765. ISSN: 09218890. DOI: 10.1016/j.robot.2014.03.001.

[9] S. Mohammad Khansari-Zadeh and Aude Billard. “Learning Stable Nonlinear Dynamical Systems With Gaussian Mixture Models”. en. In: IEEE Transactions on Robotics 27.5 (Oct. 2011), pp. 943–957. ISSN: 1552-3098, 1941-0468. DOI: 10.1109/TRO.2011.2159412.

[10] Klaus Neumann and Jochen J. Steil. “Learning robot motions with stable dynamical systems under diffeomorphic transformations”, en. In: Robotics and Autonomous Systems 70 (Aug. 2015), pp. 1–15. ISSN: 09218890. DOI: 10.1016/j.robot.2015.04.006

[11] Nadia Figueroa and Aude Billard. “A Physically-Consistent Bayesian Non-Parametric Mixture Model for Dynamical System Learning.” In: CoRL. 2018, pp. 927–946.

[12] Harish chaandar Ravichandar and Ashwin Dani. “Learning position and orientation dynamics from demonstrations via contraction analysis”. In: Autonomous Robots 43.4 (2019), pp. 897–912.

[13] J. Zico Kolter and Gaurav Manek. “Learning Stable Deep Dynamics Models”. In: Advances in Neural Information Processing Systems. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019.

[14] Sarang C Joshi and Michael I Miller. “Landmark matching via large deformation diffeomorphisms”. In: IEEE transactions on image processing 9.8 (2000), pp. 1357–1370.

[15] Julen Urain et al. “ImitationFlow: Learning Deep Stable Stochastic Dynamic Systems by Normalizing Flows”. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2020, pp. 5231–5237.

[16] Belkin Mikhail and Niyogi Partha. “Laplacian eigenmaps for dimensionality reduction and data representation”. In: Neural computation 15.6 (2003), pp. 1373–1396.

[17] Bernardo FICHERA and Aude Billard. Linearization and Identification of Multiple-Attractors Dynamical System through Laplacian Eigenmaps. 2022. DOI: 10.48550/ARXIV.2202.09171.

[18] G. J. Tee. “Eigenvectors of block circulant and alternating circulant matrices”. In: New Zealand Journal of Mathematics 36.8 (2007), pp. 195–211.

[19] Stan Salvador and Philip Ka-Fai Chan. “FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space”. In: 2004.

[20] Klas Kronander and Aude Billard. “Passive Interaction Control With Dynamical Systems”. en. In: IEEE Robotics and Automation Letters 1.1 (Jan. 2016), pp. 106–113. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2015.2509025.

[21] Ravindra B. Bapat. Graphs and matrices. Vol. 27. Springer, 2010.

[22] Roger A. Horn and Charles R. Johnson. Matrix analysis. Cambridge university press, 2012.