When Can We Treat Trajectories as Points?

Parasara Sridhar Duggirala* & Donald R. Sheehy†

Abstract

In the formal verification of dynamical systems, one often looks at a trajectory through a state space as a sample behavior of the system. Thus, metrics on trajectories give important information about the different behavior of the system given different starting states. In the important special case of linear dynamical systems, the set of trajectories forms a finite-dimensional vector space. In this paper, we exploit this vector space structure to define (semi)norms on the trajectories, give an isometric embedding from the trajectory metric into low-dimensional Euclidean space, and bound the Lipschitz constant on the map from start states to trajectories as measured in one of several different metrics. These results show that for an interesting class of trajectories, one can treat the trajectories as points while losing little or no information.

1 Introduction

The starting point for many problems in computational geometry is a discrete set of points in a Euclidean space. Alternatively, many interesting questions arise from sets of paths or trajectories, and usually, such problems require very different ideas and methods. In this paper, we consider a class of trajectories that arise naturally in the field of formal verification of cyber-physical systems (CPSs)[1] in which one can transform a collection of trajectories into a set of points while losing little or no information.

Motivated partly by recent work on using algorithms from computational geometry to analyze trajectories through the state space of a CPS [12], we highlight an interesting class of systems studied in that field, for which natural metrics on trajectories can be nicely embedded into low-dimensional Euclidean space. Generally, it would be inefficient to treat trajectories as points. Even though a discrete trajectory in the plane broken into k pieces can be thought of as a single point in $\mathbb{R}^{2k}$, the blowup in dimension can be prohibitive for most geometric algorithms, especially those where the low-dimensional (i.e. $d = 2$ or 3) structure can be exploited.

Control software in safety critical CPSs such as autonomous vehicles and power plants should always satisfy the prescribed safety specification. One of the main challenges in verifying CPS safety properties is that they involve a mix of continuous and discrete behaviors. Even if we ignore the discrete switching, the continuous dynamics in most real world CPSs are highly nonlinear and difficult to analyze. For example, when we "simplify" the nonlinear dynamics to a linear approximation, verifying such linear systems of high dimensions is still challenging due to the curse of dimensionality. In dynamic analysis techniques, a few sample executions, also called trajectories of the systems are computed. Whether the system satisfies the desired property or not is inferred after carefully analyzing the generated executions. As these techniques purely depend on sample executions, they can be easily integrated into the testing and debugging phase of CPS design.

In many CPSs, the state space is modeled as a Euclidean space. As in other dynamical systems, the next state is a function of the current state and often some inputs or controls. The executions that form the primary data are trajectories in a Euclidean space. For this work, we consider the simplest case where the system is governed by a linear dynamical system. That is, the derivative of the state is a linear function of the current state. We will show how one can model the geometry of the space of trajectories as a set of points. Naturally, this will depend on a choice of a metric on the space of trajectories. We will consider several different metrics on trajectories including L_p-type metrics as well as the Fréchet distance and the Skorokhod distance.

The main goal is to show when and to what extent one can study the class of trajectories arising from a CPS using algorithms and data structures designed for points. The hope is that this will open the door to more applications of classical computational geometry of points in Euclidean space to problems in formal verification.

2 Metrics, Norms, and Samples

In this section, we review several notions that will be very familiar to most readers. We include the formal definitions for completeness, because our results are given in considerable generality and the fine distinctions

*Computer Science Department, University of Connecticut, psd@uconn.edu

†Computer Science Department, University of Connecticut, don.r.sheehy@uconn.edu

[1]For this paper, we can identify the buzzword "cyberphysical systems" used in the verification literature with the more general notion of a dynamical system.

in the definitions (e.g. pseudometrics and seminorms) will be important.

A *metric space* $\mathcal{X}$ is a pair (X, d) where X is a set and $\mathrm{d} : X \times X \to \mathbb{R}$ is a function satisfying the following conditions.

1. Nonnegative: $\mathrm{d}(x, y) \geq 0$
2. Symmetric: $\mathrm{d}(x, y) = \mathrm{d}(y, x)$
3. Triangle Inequality: $\mathrm{d}(x, z) \leq \mathrm{d}(x, y) + \mathrm{d}(y, z)$
4. Identity of Indiscernables: $\mathrm{d}(x, y) = 0$ if and only if $x = y$.

A function d satisfying the first three properties is called a *pseudometric*.

Let V be a vector space over some subfield of the complex numbers $\mathbb{C}$. A *norm* V is a function $\|\cdot\| : V \to \mathbb{R}$ satisfying the following conditions.

1. Nonnegative: $\|v\| \geq 0$
2. Absolutely scalable: $\|cv\| = |c|\|v\|$
3. Triangle Inequality: $\|u + v\| \leq \|u\| + \|v\|$
4. Definite: If $\|v\| = 0$, then $v = 0$.

A function satisfying the first three conditions of a norm is called a *seminorm*. Every norm on V induces a metric on V defined as $\mathrm{d}(u, v) = \|u - v\|$. Similarly, a seminorm induces a pseudometric.

A function between (pseudo)metric spaces, $f : (X, \mathrm{d}_X) \to (Y, \mathrm{d}_Y)$ is λ-Lipschitz if for all $x, x' \in X$, we have

$$\mathrm{d}_Y(f(x), f(x')) \leq \lambda \mathrm{d}_X(x, x').$$

This is a basic stability condition on mappings between metric spaces. It is most often used to describe real-valued functions using the standard metric on $\mathbb{R}$.

An *ε-sample* of a subset U of a metric space is a subset $S \subseteq U$ such that for all $u \in U$, there exists $s \in S$ such that $\mathrm{d}(u, s) \leq \varepsilon$. This notion is closely related to the *Hausdorff distance*, a (pseudo)metric on compact subsets of a (pseudo)metric space.[2] It is defined as follows.

$$\mathrm{d}_H(S, T) = \max\{\max_{s \in S} \min_{t \in T} \mathrm{d}(s, t), \max_{t \in T} \min_{s \in S} \mathrm{d}(s, t)\}$$

Using this definition, an ε-sample of U is a subset $S \subseteq U$ such that $\mathrm{d}_H(S, U) \leq \varepsilon$.

[2] Compactness here is primarily required for distance minimizers to exist.

3 States and Trajectories

Let $\mathcal{X} = (X, \mathrm{d})$ denote a metric space. For this paper, $\mathcal{X}$ will represent the space of *states* of some system. Usually, we will only consider states in $\mathbb{R}^d$, but it is useful to give the following definitions in full generality.

A trajectory in $\mathcal{X}$ is a continuous function $f : [0, 1] \to \mathcal{X}$. We are only considering maps from the unit interval, though many of the results in this paper generalize naturally to other finite length intervals. We use $\mathrm{Tr}(\mathcal{X})$ to denote the set of all trajectories in $\mathcal{X}$.

3.1 Sampling trajectories

A dynamical system may be viewed as a function from states to trajectories. If we endow both the state space and the trajectory space with metrics, we can ask when such a function (the dynamical system) is Lipschitz. Having a bound on the Lipschitz constant associated to such a system justifies sampling trajectories by sampling start states. For indeed, the Lipschitz condition implies that a good sample of the valid start states will give a correspondingly good sample of the trajectories in the following precise sense.

Proposition 1 *Let $\mathcal{X}$ be a set of states and let $\Theta \subset \mathcal{X}$ be a set of start states. Let $\mathrm{Tr}(\mathcal{X})$ be a metric space of trajectories in $\mathcal{X}$ with metric T. If $S \subseteq \Theta$ is an ε-sample of Θ and $\xi : \mathcal{X} \to \mathrm{Tr}(\mathcal{X})$ is λ-Lipschitz, then*

$$\xi(S) := \{\xi(s) \mid s \in S\} \text{ is a } \lambda\varepsilon\text{-sample of } \xi(\Theta).$$

Proof. Fix any $\gamma \in \xi(\Theta)$. Then $\gamma = \xi(x)$ for some $x \in \Theta$. So, there exists $s \in S$ such that $\mathrm{d}(s, x) \leq \varepsilon$. If $\gamma' = \xi(s)$, then,

$$\begin{aligned} T(\gamma, \gamma') &= T(\xi(x), \xi(s)) \\ &\leq \lambda \mathrm{d}(x, s) \\ &\leq \lambda\varepsilon. \end{aligned}$$

So, for all $\gamma \in \xi(\Theta)$, there exists $\gamma' \in \xi(S)$ such that $T(\gamma, \gamma') \leq \lambda\varepsilon$. We conclude that $\xi(S)$ is a $\lambda\varepsilon$-sample of $\xi(\Theta)$ as desired. □

3.2 Vector Spaces of Trajectories

The set of all trajectories mapping the interval $[0, 1]$ to the state space $\mathbb{R}^d$ naturally forms a vector space. Let $c \in \mathbb{R}$ be a scalar and let $\phi, \psi : [0, 1] \to \mathbb{R}^d$ be trajectories. Scalar multiplication and vector addition are defined as

$$(c\phi)(t) := c\phi(t), \text{ and}$$
$$(\phi + \psi)(t) := \phi(t) + \psi(t).$$

In general, the dimension of this vector space is infinite. However, the following important case yields a finite-dimensional space of trajectories.

Consider systems that evolve in $\mathbb{R}^d$ in continuous time. At a given instant in time, the system state is denoted as a vector $x \in \mathbb{R}^d$ and its evolution is given as a linear differential equation, i.e.,

$$\dot{x} = Ax, \tag{1}$$

where $A \in \mathbb{R}^{d\times d}$. By using an extra variable and allowing A to be singular, this includes also the case of *affine* systems

$$\dot{x} = Ax + B, \tag{2}$$

where $B \in \mathbb{R}^{d\times 1}$. The system of trajectories, denoted as $\xi : \mathbb{R}^d \times \mathbb{R}_{\geq 0} \to \mathbb{R}^d$ are solutions of the initial value problem of the differential equation given in Equation (1). Given an initial state x_0, $\xi(x_0, t)$ denotes the state of the system at time instance t. The closed form expression for the trajectories is given in Equation (3) below.

$$\xi(x_0, t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bd\tau, \tag{3}$$

where $e^{At} = I + \frac{At}{1!} + \frac{(At)^2}{2!} + \frac{(At)^3}{3!} + \ldots$ represents the matrix exponential operation. We denote the trajectory starting from initial state x_0 as ξ_{x_0}, i.e., $\xi_{x_0}(t) = \xi(x_0, t)$ so that it fits with our previous definition of trajectory.

Definition 1 *Trajectories of linear dynamical systems (such as given in Equation (1)) satisfy the* superposition principle. *Given any state $x_0 \in \mathbb{R}^d$, vectors $v_1, v_2, \ldots, v_m \in \mathbb{R}^d$, and scalars $\alpha_1, \alpha_2, \ldots, \alpha_m \in \mathbb{R}$, the following equality is satisfied.*

$$\xi(x_0 + \sum_{i=1}^{m} \alpha_i v_i, t) = \xi(x_0, t) + \sum_{i=1}^{m} \alpha_i(\xi(x_0 + v_i, t) - \xi(x_0, t))$$

The observation made in Definition 1 follows from the closed form solution given in Equation (3). Using the superposition principle, we can infer that for any pair of states $x_0, x_1 \in \mathbb{R}^n$ and for any vector $v \in \mathbb{R}^d$,

$$\xi(x_0 + v, t) - \xi(x_0, t) = \xi(x_1 + v, t) - \xi(x_1, t)$$

These equations reveal the low dimensional vector space structure of the space of trajectories. Indeed, given any basis $(b_1, \ldots, b_n)$ for the state space, a trajectory starting from $x = \sum_{i=1}^{n} x_i b_i$ is the corresponding linear combination of *basis trajectories*:

$$\xi_x = \sum_{i=1}^{n} x_i \xi_{b_i}.$$

When the matrix A determining the system is nonsingular, then the closed form for the trajectory (Equation (3)) simplifies to

$$\xi_x(t) = e^{At}x.$$

For the rest of the paper, we will assume this case for simplicity.

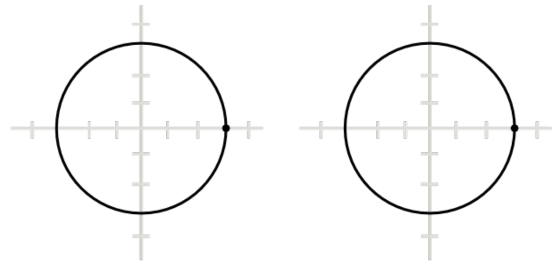

Figure 1: Two different trajectories with the same image. Left: $3e^{10\pi i t}$. Right: $3e^{2\pi i t}$.

4 Metrics Spaces of Trajectories

In this section, we will define several classes of metrics on trajectories. If one is content to view a trajectory $f : [0, 1] \to \mathbb{R}^d$ as merely a set of points in the state space, i.e.

$$\operatorname{im} f := \{f(t) \mid t \in [0, 1]\},$$

then the Hausdorff distance provides an easy to define metric on trajectories. That is, one can define

$$T_H(f, g) := \mathrm{d}_H(\operatorname{im} f, \operatorname{im} g).$$

The Hausdorff distance has a natural geometric interpretation as the minimum radius r such that expanding $\operatorname{im} f$ by r would cover $\operatorname{im} g$ and vice versa. Unfortunately, the Hausdorff distance ignores the continuous structure of the input trajectories. For example, it sees no difference between the two trajectories of Figure 1, the first of which makes five revolutions and the second makes only one.

Indeed, the Hausdorff distance only give a pseudometric on trajectories as the distance between a trajectory f and its reverse trajectory $g(t) = f(1 - t)$ is precisely 0.

The L_p-type metrics on trajectories are defined for an integer $p \geq 1$ as follows:

$$L_p(f, g) := \left(\int_0^1 \|f(t) - g(t)\|_p^p dt\right)^{\frac{1}{p}}.$$

This includes, in particular, the L_∞ distance:

$$L_{\mathrm{d},\infty}(f, g) := \max_{t\in[0,1]} \mathrm{d}(f(t), g(t)),$$

where $\mathrm{d}(\cdot, \cdot)$ can be any metric on the state space. We abuse notation and write L_∞ to denote $L_{\mathrm{d},\infty}$ with $\mathrm{d}(x, y) := \|x - y\|_2$. We do this primarily because it is such a popular metric and can also be used to define other metrics as we will see.

A drawback of L_p-type metrics (usually L_∞) is their inability to recognize similarity of trajectories that differ only by some continuous reparameterization of time.

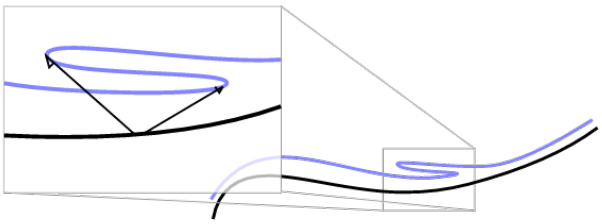

Figure 2: In this example the Fréchet distance between the curves is less than the $\mathcal{L}_\infty$ distance because reparameterization allows the lower trajectory to slow down in the neighborhood of the upper trajectory's zigzag.

For this reason, the Fréchet distance is often considered. It is defined as

$$\mathrm{d}_F(f,g) := \min_{h\in H} L_\infty(f, g\circ h),$$

where H is the set of orientation-preserving homeomorphisms $h:[0,1]\to[0,1]$, i.e. the continuous reparameterizations of time.

The Fréchet distance is also called the "dog walking distance" using the metaphor that a person walks along one trajectory and the dog walks along another [1, 13]. The person may adjust their speed (reparameterize time) so as to minimize the length of the leash (the L_∞ distance). For example in Figure 2, the lower trajectory can slow down to lessen the impact of the zigzag in the upper trajectory. It is an interesting exercise to show that using a different $\mathcal{L}_p$-type metric instead of L_∞ in the definition, does not result in a metric.

Technically, the Fréchet distance gives a pseudometric on trajectories. The trajectories f and $f\circ h$ have Fréchet distance zero despite being different functions. The triangle inequality follows by composing homeomorphisms.

The minimization in the definition of the Fréchet distance allows a substantial amount of freedom to align trajectories, sometimes too much freedom to be realistic. The *Skorokhod* distance addressed this issue by penalizing excessive time reparameterization [11]. This distance may be viewed as treating time as another spatial parameter. One starts with a metric on the space×time product $\mathbb{R}^n\times[0,1]$. For example, one could use the ℓ_p-product metric

$$\ell_p((x,s),(y,t)) := (\mathrm{d}(x,y)^p + |s-t|^p)^{1/p}.$$

This includes the ℓ_∞ metric

$$\ell_\infty((x,s),(y,t)) := \max\{\mathrm{d}(x,y), |s-t|\}.$$

Note that the definition does not specify a particular metric d on the state space. The graph of a trajectory ξ is the trajectory

$$\mathrm{Gr}(\xi)(t) = (\xi(t), t).$$

One defines a Skorokhod distance (assuming a product metric is fixed) as

$$\mathrm{d}_S(f,g) := \mathrm{d}_F(\mathrm{Gr}(f), \mathrm{Gr}(g)).$$

5 Hilbert Spaces of Trajectories

Just as with finite-dimensional vector spaces, the L_2-norm on trajectories results in a Hilbert space. The inner product is simply the integral of the standard Euclidean inner product, i.e.

$$\langle \xi_x, \xi_y\rangle := \int_0^1 \xi_x(t)^\top \xi_y(t)dt.$$

If we only consider the trajectories coming from a linear, dynamical system $\dot{x} = Ax$, the resulting Hilbert space only has the dimension of the state space as previously observed, but moreover, the induced metric is Euclidean and can be computed explicitly.

Theorem 2 *Given a dynamical system in $\mathbb{R}^d$ governed by $\dot{x} = Ax$, there exists a matrix $L\in\mathbb{R}^{d\times d}$ such that for any $x,y\in\mathbb{R}^d$,*

$$L_2(\xi_x,\xi_y) = \|Lx - Ly\|_2.$$

Proof.

$$\begin{aligned}\|\xi_x\|_2 &= \int_0^1 \xi_x(t)^\top \xi_x(t)dt\\ &= \int_0^1 (e^{At}x)^\top(e^{At}x)dt\\ &= x^T\left(\int_0^1 (e^{At})^\top(e^{At})dt\right)x\end{aligned}$$

Let M be the matrix $\int_0^1 (e^{At})^\top(e^{At})dt$ so that $\|\xi_x\|_2 = x^\top Mx$. As the matrices are positive definite for all t, it follows that M is also positive definite. Thus, the Cholesky decomposition $M = LL^\top$ exists and the matrix L has the property that

$$\|\xi_x\|_2 = (Lx)^\top(Lx).$$

This fact about the L_2 norm immediately implies the corresponding claim about the L_2 metric. □

Computational Issues and Implications Following the proof of Theorem 2, a natural approach to computing L, at least approximately, is to discretize the integral $\int_0^1 (e^{At})^\top(e^{At})dt$ and compute the pieces using the leading terms of the expansion of the matrix exponential, $e^{At} = I + \frac{At}{1!} + \frac{(At)^2}{2!} + \cdots$. The result is a positive definite matrix M whose Cholesky decomposition gives the desired linear operator L.

One immediate use for Theorem 2 is in the analysis of collections of trajectories. Any data analysis depending on the distances between trajectories, such as clustering or subsampling, would naturally require computing many pairwise distances. If the trajectories are discretized into k pieces, the straightforward computation of the L_2 distance would require $O(kd)$ time. If instead, one first computes L, then the time to compute these distances is reduced to $O(d^2)$. For high-fidelity measurements with large k, this can be a substantial speedup. This idea, though technically simple, has not been exploited in the literature on formal verification of these systems.

6 The Lipschitz Bound

It is only a small consolation if a car that always crashes, spends most of its time in an "uncrashed" state. Unfortunately, it is the nature of L_2 metrics to average distances over time, so such bad behaviors of a system could be missed. For this reason, a max-norm like L_∞ is often preferable. However, an isometric embedding of the trajectories into Euclidean space as in Theorem 2 is not possible (consider $A = 0$ for example). The alternative we propose in this section is to at least bound the Lipschitz constant of the system when viewed as a mapping from a metric on states to a metric on trajectories. We give such a bound in considerable generality; it applies to any seminorm on trajectories and we show its implications for other pseudometrics including the Fréchet and Skorokhod distance.

Theorem 3 *Let $\mathbb{R}^d$ be the states equipped with a norm $\|\cdot\|$. Let $\dot{x} = Ax$ be a linear dynamical system and let $\mathrm{Tr}_A(\mathbb{R}^d)$ be the trajectories in $\mathbb{R}^d$ arising from A endowed with some seminorm, $\|\cdot\|_{\mathrm{Tr}}$. Let $(b_1, \ldots, b_d)$ be a basis for $\mathbb{R}^d$ and let $(\xi_1, \ldots, \xi_d)$ be the corresponding basis for $\mathrm{Tr}_A(\mathbb{R}^d)$. Let $t = \|(\|\xi_1\|_{\mathrm{Tr}}, \ldots, \|\xi_d\|_{\mathrm{Tr}})\|$ Then, the mapping from $\mathbb{R}^d$ to $\mathrm{Tr}_A(\mathbb{R}^d)$ is t-Lipschitz. That is, for any pair of states $x, y \in \mathbb{R}^d$,*

$$\|\xi_x - \xi_y\|_{\mathrm{Tr}} \leq t\|x - y\|.$$

Proof. Write the states x and y in terms of the basis $(b_1, \ldots, b_d)$ as follows.

$$x = \sum_{i=1}^{d} x_i b_i \text{ and } y = \sum_{i=1}^{d} y_i b_i.$$

Then, the trajectories ξ_x and ξ_y can be written in the corresponding basis of trajectories as follows.

$$\xi_x = \sum_{i=1}^{d} x_i \xi_i \text{ and } \xi_y = \sum_{i=1}^{d} y_i \xi_i.$$

We can now bound $\|\xi_x - \xi_y\|_{\mathrm{Tr}}$ as follows.

$$\begin{aligned}
\|\xi_x - \xi_y\|_{\mathrm{Tr}} &= \|\sum_{i=1}^{d}(x_i - y_i)\xi_i\|_{\mathrm{Tr}} && \text{[by definition]}\\
&\leq \sum_{i=1}^{d}\|(x_i - y_i)\xi_i\|_{\mathrm{Tr}} && \text{[triangle ineq.]}\\
&= \sum_{i=1}^{d}(x_i - y_i)\|\xi_i\|_{\mathrm{Tr}} && \text{[norms are linear]}\\
&\leq t\|x - y\| && \text{[Cauchy-Schwarz]}
\end{aligned}$$

□

The hypothesis about working with (semi)norms in the theorem above are necessary to apply the Cauchy-Schwarz inequality. Thus, it's not clear how to duplicate this proof while replacing the norms on trajectories with a more elaborate distance on trajectories. However, the theorem naturally extends to give a Lipschitz bound when the distances on trajectories are measured using either the Fréchet distance or the Skorokod distance by using the relationship between these metrics and the L_∞ norm.

Theorem 4 *Let $\mathbb{R}^d$ be the states equipped with a norm $\|\cdot\|$. Let A be a linear dynamical system and let $\mathrm{Tr}_A(\mathbb{R}^d)$ be the trajectories in $\mathbb{R}^d$ arising from A endowed with either the Fréchet distance or the Skorokhod distance. Let $(b_1, \ldots, b_d)$ be a basis for $\mathbb{R}^d$ and let $(\xi_1, \ldots, \xi_d)$ be the corresponding basis for $\mathrm{Tr}_A(\mathbb{R}^d)$. Let $t = \|(\|\xi_1\|_\infty, \ldots, \|\xi_d\|_\infty)\|$ Then, the mapping from $\mathbb{R}^d$ to $\mathrm{Tr}_A(\mathbb{R}^d)$ is t-Lipschitz.*

Proof. It suffices to observe that for any pair of trajectories $f, g : [0, 1] \to \mathbb{R}^d$ and any ℓ_p-product metric on $\mathbb{R}^d \times [0, 1]$, the following inequalities hold.

$$\mathrm{d}_S(f, g) \leq \mathrm{d}_F(f, g) \leq L_\infty(f, g)$$

These inequalities hold by replacing the minimization over homeomorphisms with the specific choice of the identity homeomorphism. Thus, the theorem follows from Theorem 3 using the L_∞ metric on trajectories. □

7 Related Work, Conclusion, and Future Work

Proving properties of software systems while leveraging the data generated from the sample executions has been a well studied topic in the domain of formal verification [10, 14, 7, 15, 9]. However, these techniques do not deal with the CPS where the dynamics of the physical environment is of utmost importance. Recent techniques to integrate the information generated from sample trajectories for proving properties of CPS have been investigated [4, 5]. In a recent work [6], the fact that the trajectories form a vector space has been leveraged to improve the scalability of verification by two orders of magnitude [2]. Techniques similar to [5] to provide probabilistic guarantees about trajectories of CPS have been investigated in [8].

This work attempts to address the gap between the data-driven verification technique and computational geometry. The focus on linear dynamical systems is because of two reasons. First, linear dynamical systems describe a large set of control systems that are in deployment. Second, these systems enjoy a rich set of properties (such as the superposition principle) that can be readily exploited to represent trajectories as points. Mapping from trajectories to points would also help us in performing topological data analysis [3] over trajectories.

While in this paper we considered a specific sub-class of dynamical systems, in our future work, we intend to apply similar techniques to nonlinear dynamics. Our goal is to eventually perform data-driven analysis of trajectories where partial or no model information is available.

References

[1] Helmut Alt and Michael Godau. Computing the fréchet distance between two polygonal curves. *Int. J. Comput. Geometry Appl*, 5(1 & 2):75–91, 1995.

[2] Stanley Bak and Parasara Sridhar Duggirala. Simulation-equivalent reachability of large linear systems with inputs. In *International Conference on Computer Aided Verification*, pages 401–420. Springer, 2017.

[3] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[4] Alexandre Donzé and Oded Maler. Systematic simulation using sensitivity analysis. In *International Workshop on Hybrid Systems: Computation and Control*, pages 174–189. Springer, 2007.

[5] Parasara Sridhar Duggirala, Sayan Mitra, and Mahesh Viswanathan. Verification of annotated models from executions. In *Proceedings of the Eleventh ACM International Conference on Embedded Software*, page 26. IEEE Press, 2013.

[6] Parasara Sridhar Duggirala and Mahesh Viswanathan. Parsimonious, simulation based verification of linear systems. In *International Conference on Computer Aided Verification*, pages 477–494. Springer, 2016.

[7] Michael D Ernst, Jeff H Perkins, Philip J Guo, Stephen McCamant, Carlos Pacheco, Matthew S Tschantz, and Chen Xiao. The daikon system for dynamic detection of likely invariants. *Science of Computer Programming*, 69(1-3):35–45, 2007.

[8] Chuchu Fan, Bolun Qi, Sayan Mitra, and Mahesh Viswanathan. Dryvr: Data-driven verification and compositional reasoning for automotive systems. In *International Conference on Computer Aided Verification*, pages 441–461. Springer, 2017.

[9] Pranav Garg, Christof Löding, P Madhusudan, and Daniel Neider. Ice: A robust framework for learning invariants. In *International Conference on Computer Aided Verification*, pages 69–87. Springer, 2014.

[10] Patrice Godefroid, Nils Klarlund, and Koushik Sen. Dart: directed automated random testing. In *ACM Sigplan Notices*, volume 40, pages 213–223. ACM, 2005.

[11] Rupak Majumdar and Vinayak S. Prabhu. Computing the skorokhod distance between polygonal traces. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, pages 199–208, 2015.

[12] Rupak Majumdar and Vinayak S Prabhu. Computing distances between reach flowpipes. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, pages 267–276. ACM, 2016.

[13] Günter Rote. Computing the fréchet distance between piecewise smooth curves. *Computational Geometry: Theory and Applications*, 37(3):162–174, 2007.

[14] Koushik Sen, Darko Marinov, and Gul Agha. Cute: a concolic unit testing engine for c. In *ACM SIGSOFT Software Engineering Notes*, volume 30, pages 263–272. ACM, 2005.

[15] Rahul Sharma. *Data-driven Verification*. PhD thesis, Stanford University, 2016.