Proceedings of Machine Learning Research vol 242:1–14, 2024 6th Annual Conference on Learning for Dynamics and Control

Signatures Meet Dynamic Programming: Generalizing Bellman Equations for Trajectory Following

Motoya Ohnishi [*] *University of Washington* MOHNISHI@CS.WASHINGTON.EDU

Iretiayo Akinola *NVIDIA*

Jie Xu *NVIDIA*

Ajay Mandlekar *NVIDIA*

Fabio Ramos *University of Sydney, NVIDIA*

Editors: A. Abate, K. Margellos, A. Papachristodoulou

Abstract

Path signatures have been proposed as a powerful representation of paths that efficiently captures the path's analytic and geometric characteristics, having useful algebraic properties including fast concatenation of paths through tensor products. Signatures have recently been widely adopted in machine learning problems for time series analysis. In this work we establish connections between value functions typically used in optimal control and intriguing properties of path signatures. These connections motivate our novel control framework with signature transforms that efficiently generalizes the Bellman equation to the space of trajectories. We analyze the properties and advantages of the framework, termed signature control. In particular, we demonstrate that (i) it can naturally deal with varying/adaptive time steps; (ii) it propagates higher-level information more efficiently than value function updates; (iii) it is robust to dynamical system misspecification over long rollouts. As a specific case of our framework, we devise a model predictive control method for path tracking. This method generalizes integral control, being suitable for problems with unknown disturbances. The proposed algorithms are tested in simulation, with differentiable physics models including typical control and robotics tasks such as point-mass, curve following for an ant model, and a robotic manipulator.

Keywords: Decision making, Path signature, Bellman equation, Integral control, Model predictive control, Robotics

1. Introduction

Path tracking has been a central problem for autonomous vehicles (e.g., Schwarting et al. (2018); Aguiar and Hespanha (2007)), imitation learning, learning from demonstrations (cf. Hussein et al. (2017); Argall et al. (2009)), character animations (with mocap systems; e.g., Peng et al. (2018)), robot manipulation for executing plans returned by a solver (cf. Kaelbling and Lozano-Pérez (2011); Garrett et al. (2021)), and for flying drones (e.g., Zhou and Schwager (2014)), just to name a few.

Typically, path tracking is dealt with by using reference dynamics if available or is formulated as a control problem with a sequence of goals to follow using optimal controls based on dynamic programming (DP) (Bellman, 1953). In particular, DP over the scalar or *value* of a respective policy is often studied and analyzed through the lenses of the Bellman expectation (or optimality) equation which describes the evolution of values or rewards over time (cf. Kakade (2003); Sutton and Barto

[*] The work of MO was partially conducted during his internship at NVIDIA. Refer to project page: https://sites.google.com/view/sigctrl/ for supplementary information.

(2018); Kaelbling et al. (1996)). This is done by computing and updating a *value function* which maps states to values.

However, value (cumulative reward) based trajectory (or policy) optimization is suboptimal for path tracking where appropriate waypoints are unavailable. Further, value functions capture state information exclusively through its scalar value, which makes it unsuitable for the problems that require the entire trajectory information to obtain a desirable control strategy.

In this work, we adopt a rough-path theoretical approach in DP; specifically, we exploit path signatures (cf. Chevyrev and Kormilitzin (2016); Lyons (1998)), which have been widely studied as a useful geometrical feature representation of path, and have recently attracted the attention of the machine learning community (e.g., Chevyrev and Kormilitzin (2016); Kidger et al. (2019); Salvi et al. (2021); Morrill et al. (2021); Levin et al. (2013); Fermanian (2021)). Our decision making framework predicated on signatures, named signature control, describes an evolution of signatures over an agent's trajectory through DP. By demonstrating how it reduces to the Bellman equation as a special case, we show that the S*-function* representing the signatures of future path (we call it *path-to-go* in this paper) is cast as an effective generalization of value function. In addition, since an S-function naturally encodes information of a long trajectory, it is robust against misspecification of dynamics. Our signature control inherits some of the properties of signatures, namely, time-parameterization invariance, shift invariance, and tree-like equivalence (cf. Lyons et al. (2007); Boedihardjo et al. (2016)); as such, when applied to tracking problems, there is no need to specify waypoints.

In order to devise new algorithms from this framework, including model predictive controls (MPCs) (Camacho and Alba, 2013), we present path cost designs and their properties. In fact, our signature control generalizes the classical integral control (see Khalil (2002)); it hence shows robustness against unknown disturbances, which is demonstrated in robotic manipulator simulation.

Notation: Throughout this paper, we let $\mathbb{R}$, $\mathbb{N}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{Z}_{>0}$ be the set of the real numbers, the natural numbers ($\{0, 1, 2, \ldots\}$), the nonnegative real numbers, and the positive integers, respectively. Also, let $[T] := \{0, 1, \ldots, T-1\}$ for $T \in \mathbb{Z}_{>0}$. The floor and the ceiling of a real number a is denoted by $\lfloor a \rfloor$ and $\lceil a \rceil$, respectively. Let $\mathbb{T}$ denote time for random dynamical systems, which is defined to be either $\mathbb{N}$ (discrete-time) or $\mathbb{R}_{\geq 0}$ (continuous-time).

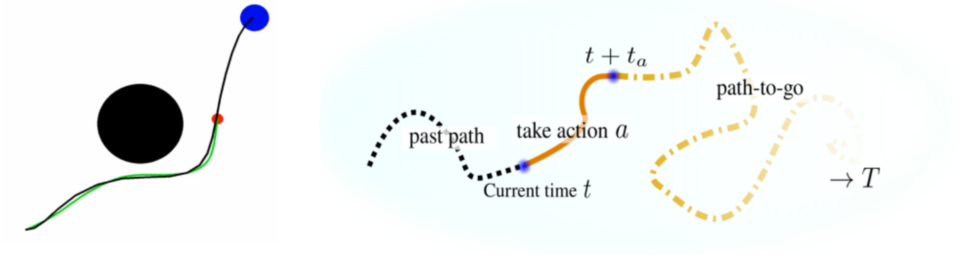

Figure 1: Left: simple tracking example. The black and blue circles represent an obstacle and the goal. Given a path (black line), a point-mass (red) follows this reference via minimization of deviation of signatures in an online fashion with optimized action repetitions. Right: illustration of path-to-go formulation as an analogy to value-to-go in the classical settings.

2. Related work

Path signature: Path signatures are mathematical tools developed in rough path research (Lyons, 1998; Chen, 1954; Boedihardjo et al., 2016). For efficient computations of metrics over signatures, kernel methods (Hofmann et al., 2008; Aronszajn, 1950) are employed (Király and Oberhauser, 2019; Salvi et al., 2021; Cass et al., 2021; Salvi et al., 2021). Signatures have been applied to various applications, such as sound compression (Lyons and Sidorova, 2005), time series data analysis and regression (Gyurkó et al., 2013; Lyons, 2014; Levin et al., 2013), action and text recognition (Yang et al., 2022; Xie et al., 2017), and neural rough differential equations (Morrill et al., 2021). Also, deep signature transform is proposed in (Kidger et al., 2019) with applications to reinforcement learning which is still within Bellman equation based paradigm. In contrast, our framework is solely replacing this Bellman backup with signature based DP. Theory and practice of path signatures in machine learning are summarized in (Chevyrev and Kormilitzin, 2016; Fermanian, 2021).

Value-based control and RL: Value function based methods are widely adopted in RL and optimal control. However, value functions capture state information exclusively through its scalar value, which lowers sample efficiency. To alleviate this problem, model-based RL methods learn one-step dynamics across states (cf. Sun et al. (2019); Du et al. (2021); Wang et al. (2019); Chua et al. (2018)). In practical algorithms, however, even small errors on one-step dynamics could diverge along multiple time steps, which hinders the performance (cf. Moerland et al. (2023)). To improve sample complexity and generalizability of value function based methods, on the other hand, successor features (e.g., Barreto et al. (2017)) have been developed. Costs over the spectrum of the Koopman operator were proposed by Ohnishi et al. (2021), but without DP. We tackle the issue from another angle by capturing sufficiently rich information in a form of *path-to-go*, which is robust against long horizon problems; at the same time, it subsumes value function (and successor feature) updates.

Path tracking: Traditionally, the optimal path tracking control is achieved by using reference dynamics or by assigning time-varying waypoints (Paden et al., 2016; Schwarting et al., 2018; Aguiar and Hespanha, 2007; Zhou and Schwager, 2014; Patle et al., 2019). In practice, MPC is usually applied for tracking time-varying waypoints and PID controls are often employed when optimal control dynamics can be computed offline (cf. Khalil (2002)). Some of the important path tracking methodologies were summarized in (Rokonuzzaman et al., 2021); namely, pure pursuit (Scharf et al., 1969), Stanley controller (Thrun et al., 2007), linear control such as the linear quadratic regulator after feedback linearization (Khalil, 2002), Lyapunov's direct method, robust or adaptive control using simplified or partially known system models, and MPC. Those methodologies are highly sensitive to time step sizes and require analytical (and simplified) system models, and misspecifications of dynamics may also cause significant errors in tracking accuracy even when using MPC. Due to those limitations, many ad-hoc heuristics are often required when applying them in practice. Our signature based method can systematically remedy some of those drawbacks in its vanilla form.

3. Preliminaries

3.1. Path signature

Let $\mathcal{X} \subset \mathbb{R}^d$ be a state space and suppose a path (a continuous stream of states) is defined over a compact time interval $[s, t]$ for $0 \leq s < t$ as an element of $\mathcal{X}^{[s,t]}$. The path signature is a collection of infinitely many features (scalar coefficients) of a path with *depth* one to infinite. Coefficients of each depth roughly correspond to the geometric characteristics of paths, e.g., displacement and area surrounded by the path can be expressed by coefficients of depth one and two.

The formal definition of path signatures is given below. We use $T((\mathcal{X}))$ to denote the space of formal power series and $\otimes$ to denote the tensor product operation (see supplementary materials).

Definition 1 (Path signatures (Lyons et al., 2007)) *Let $\Sigma \subset \mathcal{X}^{[0,T]}$ be a certain space of paths. Define a map on Σ over $T((\mathcal{X}))$ by $S(\sigma) \in T((\mathcal{X}))$ for a path $\sigma \in \Sigma$ where its coefficient corresponding to the basis $e_i \otimes e_j \otimes \ldots \otimes e_k$ is given by*

$$S(\sigma)_{i,j,\ldots,k} := \int_{0<\tau_k<T} \cdots \int_{0<\tau_j<\tau_k} \int_{0<\tau_i<\tau_j} dx_{\tau_i,i} dx_{\tau_j,j} \ldots dx_{\tau_k,k}.$$

Here, $x_{t,i}$ is the ith coordinate of σ at time t.

The space Σ is chosen so that the path signature $S(\sigma)$ of $\sigma \in \Sigma$ is well-defined. Given a positive integer m, the truncated signature $S_m(\sigma)$ is defined accordingly by a truncation of $S(\sigma)$ (as an element of the quotient denoted by $T^m(\mathcal{X})$; see supplementary materials for the definition).

Properties of the path signatures: The basic properties of path signature allow us to develop our decision making framework and its extension to control algorithms. Such properties are also inherited by the algorithms we devise, providing several advantages over classical methods for tasks such as path tracking (further details are in Section 5 and 6). We summarize these properties below:

- The signature of a path is invariant under a constant shift and time reparameterizations. Straightforward applications of signatures thus represent *shape* information of a path irrespective of waypoints and/or absolute initial positions.
- A path is uniquely recovered from its signature up to tree-like equivalence (e.g., path with *detours*) and the magnitudes of coefficients decay as depth increases. As such, (truncated) path signatures contain sufficiently rich information about the state trajectory, providing a valuable and compact representation of a path in several control problems.
- Any real-valued continuous map on the certain space of paths can be approximated to arbitrary accuracy by a linear map on the space of signatures (Arribas, 2018). This universality property enables us to construct a kernel operating over the space of trajectories, which will be critical to derive our control framework in section 4.
- The path signature has a useful algebraic property known as Chen's identity (Chen, 1954), stating that the signature of the concatenation of paths can be computed by the tensor product of the signatures of the paths. Let $X : [a, b] \to \mathbb{R}^d$ and $Y : [b, c] \to \mathbb{R}^d$ be two paths. Then,

  $$S(X * Y)_{a,c} = S(X)_{a,b} \otimes S(Y)_{b,c},$$

  where $*$ denotes the concatenation operation (we shift the starting time of path when necessary).

3.2. Dynamical systems and path tracking

Since we are interested in cost definitions over the entire path and not limited to the form of the cumulative cost over a trajectory with fixed time interval (or integral along continuous time), Markov Decision Processes (MDPs; Bellman (1957)) are no longer the most suitable representation. Instead, we assume that the system dynamics of an agent is described by a stochastic dynamical system Φ

(SDS; in other fields the term is also known as Random Dynamical System (RDS) (Arnold, 1998)). We defer the mathematical definition to supplementary materials. In particular, let π be a policy in a space Π which defines the SDS Φ_π (it does not have to be a map from a state to an action). Examples of stochastic dynamical systems include Markov chains, stochastic differential equations, and additive-noise (zero-mean i.i.d.) systems.

Our main problem of interest is path tracking which we formally define below:

Definition 2 (Path tracking) *Let $\Gamma : \Sigma \times \Sigma \to \mathbb{R}_{\geq 0}$ be a cost function on the product of the spaces of paths over the nonnegative real number, satisfying:*

$$\forall \sigma \in \Sigma : \ \Gamma(\sigma, \sigma) = 0; \qquad \forall \sigma^* \in \Sigma, \ \forall \sigma \in \Sigma \text{ s.t. } \sigma \not\equiv_\sigma \sigma^* : \ \Gamma(\sigma, \sigma^*) > 0,$$

where $\equiv_\sigma$ is any equivalence relation of path in Σ. Given a reference path $\sigma^* \in \Sigma$ and a cost, the goal of path tracking is to find a policy $\pi^* \in \Pi$ such that a path σ_π generated by the SDS Φ_π satisfies

$$\pi^* \in \arg\min_{\pi \in \Pi} \Gamma(\sigma_\pi, \sigma^*).$$

With these definitions we can now present a novel control framework, namely, signature control.

4. Signature control

An SDS creates a discrete or continuous path $\mathbb{T} \cap [0, T] \to \mathcal{X}$. One may transform the obtained path onto Σ as appropriate (see supplementary materials for detailed description and procedures). Our signature control problem is described as follows:

4.1. Problem formulation

Problem 3 (signature control) *Let $T \in [0, \infty)$ be a horizon. The signature control problem reads*

$$\text{Find } \pi^* \text{ s.t. } \pi^* \in \arg\min_{\pi \in \Pi} c\left(\mathbb{E}_\Omega\left[S\left(\sigma_\pi\left(x_0, T, \omega\right)\right)\right]\right), \tag{4.1}$$

where $c : T((\mathcal{X})) \to \mathbb{R}_{\geq 0}$ is a cost function over the space of signatures. $\sigma_\pi(x_0, T, \omega)$ is the (transformed) path for the SDS Φ_π associated with a policy π, x_0 is the initial state and ω is the realization for the noise *model.*

This problem subsumes the MDP as we shall see in Section 4.2. The given formulation covers both discrete-time (i.e., $\mathbb{T}$ is $\mathbb{N}$) and continuous-time cases through interpolation over time. To simplify notation, we omit the details of probability space (e.g., realization ω and sample space Ω) in the rest of the paper with a slight sacrifice of rigor (see supplementary materials for details). Given a reference σ^*, when $c(S(\sigma)) = \Gamma(\sigma, \sigma^*)$ and $\equiv_\sigma$ denotes tree-like equivalence, signature control becomes the path tracking Problem 2. To effectively solve it we exploit DP in the next section.

4.2. Dynamic programming over signatures

Path-to-go: Let $a \in \mathcal{A}$ be an *action* which basically constrains the realizations of path up to time t_a (actions studied in MDPs constrain the one-step dynamics from a given state). Given $T \in \mathbb{T}$, *path-to-go*, or the future path generated by π, refers to $\mathcal{P}^\pi$ defined by

$$\mathcal{P}^\pi(x, t) = \sigma_\pi(x, T - t), \ \forall t \in [0, T].$$

It follows that each realization of the path assumed to be constrained by an action a can be written by

$$\mathcal{P}^\pi(x,t) = \mathcal{P}^\pi_a(x,t) * \mathcal{P}^\pi(x^+, t_a + t), \quad \mathcal{P}^\pi_a(x,t) := \sigma_\pi(x, \min\{T-t, t_a\}),$$

where x^+ is the state reached after t_a from x (see Figure 1 Right). Under Markov assumption (see supplementary materials), the path generation after t_a does not depend on action a. To express this in the signature form, we exploit the Chen's identity, and define the *signature-to-go* function (or in short S-function):

$$\mathcal{S}^\pi(a,x,t) := \mathbb{E}\left[S(\mathcal{P}^\pi(x,t))|a\right]. \tag{4.2}$$

Using the Chen's identity, the law of total expectation, the Markov assumption, the properties of tensor product and the path transformation, we obtain the update rule:

Theorem 4 (Signature Dynamic Programming for Decision Making) *Let the function $\mathcal{S}$ be defined by (4.2). Under the Markov assumption, it follows that*

$$\mathcal{S}^\pi(a,x,t) = \mathbb{E}\left[S(\mathcal{P}^\pi_a(x,t)) \otimes \mathcal{ES}^\pi(x^+, t+t_a)|a\right]$$

where the expected S-function $\mathcal{ES}^\pi$ *is defined by taking expectation over actions as below:*

$$\mathcal{ES}^\pi(x,t) := \mathbb{E}\left[\mathcal{S}^\pi(a,x,t)\right].$$

Truncated signature formulation: For the mth-depth truncated signature (note that $m=\infty$ for signature with no truncation), we obtain,

$$(S(X)\otimes S(Y))_m = (S(X)_m \otimes S(Y)_m)_m =: S(X)_m \otimes_m S(Y)_m. \tag{4.3}$$

Therefore, when the cost only depends on the first mth-depth signatures, keeping track of the first mth-depth S-function $\mathcal{S}^\pi_m(a,x,t)$ suffices, and the cost function c can be efficiently computed as

$$c(\mathcal{S}^\pi_m(a,x,t)) = c\left(\mathbb{E}\left[S_m(\mathcal{P}^\pi_a(x,t)) \otimes_m \mathcal{ES}^\pi_m(x^+, t+t_a)|a\right]\right).$$

Reduction to the Bellman equation: Recall that the Bellman expectation equation w.r.t. action-value function or Q-function is given by

$$Q^\pi(a,x,t) = \mathbb{E}\left[r(a,x) + \gamma V^\pi(x^+, t+1)\middle|a\right], \tag{4.4}$$

where $V^\pi(x,t) = \mathbb{E}[Q^\pi(a,x,t)]$ for all $t \in \mathbb{N}$, where $\gamma \in (0,1]$ is a discount factor. We briefly show how this equation can be described by a S-function formulation. Here, the action a is the typical action input considered in MDPs. We suppose discrete-time system ($\mathbb{T} = \mathbb{N}$), and that the state is augmented by reward and time, and suppose $t_a = 1$ for all $a \in \mathcal{A}$. Let the depth of signatures to keep be $m = 2$. Then, by properly defining (see supplementary materials for details) the interpolation and transformation, we obtain the path illustrated in Figure 2 Left over the time index and immediate reward. For this two dimensional path, note a signature element of depth two represents the surface surrounded by the path (colored by yellow in the figure), which is equivalent to the value-to-go. As such, define the cost $c : T^2(\mathcal{X}) \to \mathbb{R}_{\geq 0}$ by $c(s) = -s_{1,2}$, and Chen equation becomes

$$\begin{aligned} c(\mathcal{S}^\pi_2(a,x,t)) &= c\left(\mathbb{E}\left[S_2(\mathcal{P}^\pi_a(x,t)) \otimes_2 \mathcal{ES}^\pi_2(x^+, t+1)|a\right]\right) \\ &= \mathbb{E}\left[-S_{1,2}(\mathcal{P}^\pi_a(x,t)) + c\left(\mathcal{ES}^\pi_2(x^+, t+1)\right)|a\right]. \end{aligned}$$

Now, it reduces to the Bellman expectation equation (4.4) because

$$c\left(\mathcal{S}^\pi_2(a,x,t)\right) = -\gamma^t Q^\pi(a,x,t),\ c\left(\mathcal{ES}^\pi_2(x,t)\right) = -\gamma^t V^\pi(x,t),\ S_{1,2}(\mathcal{P}^\pi_a(x,t)) = \gamma^t r(a,x).$$

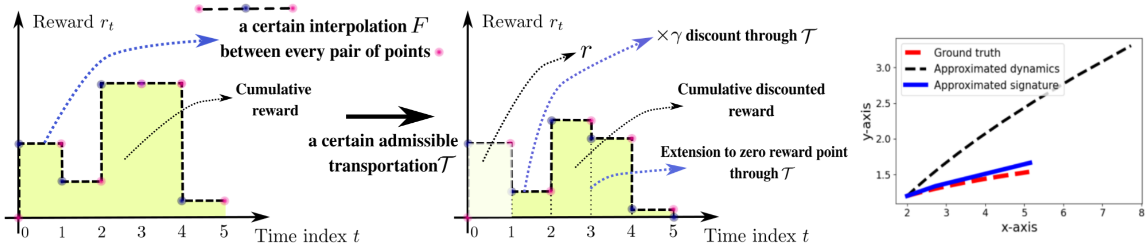
Reward r_t
a certain interpolation F
between every pair of points
Cumulative reward
a certain admissible transportation $\mathcal{T}$
0 1 2 3 4 5
Time index t
Reward r_t
r
$\times\gamma$ discount through $\mathcal{T}$
Cumulative discounted reward
Extension to zero reward point through $\mathcal{T}$
0 1 2 3 4 5
Time index t
Ground truth
Approximated dynamics
Approximated signature
y-axis
1.5
2.0
2.5
3.0
x-axis
2 3 4 5 6 7 8

Figure 2: Left: how a cumulative (discounted) reward is represented by our path formulation by an interpolation for representing the value as surface and by a transportation for discounting and concatenations of paths. Right: an error of approximated one-step dynamics propagates through time steps; while an error on signature has less effect over horizon.

Algorithm 1 Signature MPC

```
Input: initial state $x_0$; signature depth $m$; initial signature of past path $s_0 = \mathbf{1}$; # actions for rollout $N$; surrogate cost $\ell$ and regularizer $\ell_{\text{reg}}$; terminal $S$-function $\mathcal{TS}_m$; simulation model $\hat{\Phi}$
while not task finished do
    Observe the current state $x_t$.
    Update the signature of past path: $s_t = s_{t-1} \otimes_m S_m(\sigma(x_{t-1}, x_t))$, where $S_m(\sigma(x_{t-1}, x_t))$ is the signature transform of the subpath traversed since the last update from $t-1$ to $t$.
    Compute the $N$ optimal future actions $\mathbf{a}^* := (a_0, a_1, \dots, a_{N-1})$ using a simulation model $\hat{\Phi}$ that minimize the cost of the signature of the entire path (See Equation (5.1)).
    Run the first action $a_0$ for the associated duration $t_{a_0}$.
end
```

5. Signature MPC

We discuss an effective cost formulation over signatures for flexible and robust MPC, followed by additional numerical properties of signatures that benefit signature control.

Signature model predictive control: We present an application of Chen equation to MPC control– an iterative, finite-horizon optimization framework for control. In our signature MPC formulation, the optimization cost is defined for the signature of the full path being tracked, i.e., the previous path seen so far and the future path generated by the optimized control inputs (e.g., distance from the reference path signature for path tracking problem). Our algorithm, given in Algorithm 1, works in the receding-horizon manner and computes a fixed number of actions (the execution time for the full path can vary as each action may have a different time scale; i.e., each action is taken effect up to optimized (or fixed) time $t_a \in \mathbb{T}$).

Given the signature s_t of transformed past path (depth m) and the current state x_t at time t, the actions are selected by minimizing a two-part objective which is the sum of the surrogate cost ℓ and some regularizer ℓ_{reg}:

$$J = \begin{cases} \ell\Big(s_t \quad \otimes_m \quad \mathbb{E}\left[S_m(\sigma_{\mathbf{a}}(x_t)) \otimes_m \mathcal{TS}_m(x_0, s_t, \sigma_{\mathbf{a}}(x_t))\right]\Big) & \text{surrogate cost} \\ + \quad \ell_{\text{reg}}\Big(\mathbb{E}\left[\mathcal{TS}_m(x_0, s_t, \sigma_{\mathbf{a}}(x_t))\right]\Big) & \text{regularizer} \end{cases} \tag{5.1}$$

where the path $\sigma_{\mathbf{a}}(x_t)$ is traced by the optimization variable $\mathbf{a} := (a_0, a_1, \ldots a_{N-1})$, and $\mathcal{TS}_m : \mathcal{X} \times T^m(\mathcal{X}) \times \Sigma \rightarrow T^m(\mathcal{X})$ is the *terminal S-function* that may be defined arbitrarily (as an analogy to terminal value in Bellman equation based MPC; see supplementary materials).

Terminal S-function returns the signature of the terminal path-to-go. For the tracking problems studied in this work, we define the terminal subpath (path-to-go) as the final portion of the reference path starting from the closest point to the end-point of roll-out. This choice optimizes for actions up until the horizon anticipating that the reference path can be tracked afterward. We observed that this choice worked the best for simple classical examples analyzed in this work.

Error explosion along time steps: We consider robustness against misspecification of dynamics. Figure 2 Right shows an example where the dashed red line is the ground truth trajectory with the true dynamics. When there is an approximation error on the one-step dynamics being modeled, the trajectory deviates significantly (black dashed line). On the other hand, when the same amount of error is added to each term of signature, the recovered path (blue solid line) is less erroneous. This is because signatures capture the entire trajectory globally (see supplementary materials for details).

Computations of signatures: We compute the signatures through the kernel computations using an approach in (Salvi et al., 2021). We emphasize that the discrete points we use for computing the signatures of (past/future) trajectories are not regarded as waypoints, and their placement has negligible effects on the signatures as long as they sufficiently maintain the “shape” of the trajectories.

6. Experimental results

We conduct experiments on both simple classical examples and simulated robotic tasks. We also present the relation of a specific instance of signature control to the classical integral control to show its robustness against disturbance. For more experiment details, see supplementary materials.

Simple point-mass: We use double-integrator point-mass as a simple example to demonstrate our approach (as shown in Figure 1 Left). In this task, a point-mass is controlled to reach a goal position while avoiding the obstacle in the scene. We first generate a collision-free path via RRT* (Karaman and Frazzoli, 2011) which is suboptimal in terms of tracking speed (taking 72 seconds). We then employ our signature MPC to follow this reference by producing the actions (*i.e.* accelerations), resulting in a better tracking speed (taking around 30 seconds) while matching the trajectory shape.

Two-mass, spring, damper system: To view the integral control (Khalil, 2002) within the scope of our proposed signature control formulation, recall a second depth signature term corresponding to the surface surrounded by the time axis, each of the state dimension, and the path, represents each dimension of the integrated error. In addition, a first depth signature term with the initial state x_0 represents the immediate error, and the cost c may be a weighted sum of these two. To test this, we consider two-mass, spring, damper system; the disturbance is assumed zero for planning, but is 0.03 for executions. We compare signature MPC where the cost is the squared Euclidean distance between the signatures of the reference and the generated paths with truncation upto the first and the second depth. The results of position evolutions of the two masses are plotted in Figure 3 Top. As expected, the black line (first depth) does not converge to zero error state while the blue line (second depth) does (see supplementary materials). If we further include other signature terms, signature control effectively becomes a generalization of integral controls, which we will see for robotic arm experiments later.

Table 1: Selected results on path following with an ant robot model. Comparing signature control, and baseline MPC and SAC RL with equally assigned waypoints. "Slow" means it uses more time steps than our signature control for reaching the goal.

	Deviation (distance) from reference			
	Mean (10^{-2}m)	Variance (10^{-2}m)	# waypoints	reaching goal
signature control	$\mathbf{21.266}$	$\mathbf{6.568}$	N/A	success
baseline MPC	102.877	182.988	880	fail
SAC RL	446.242	281.412	880	fail
baseline MPC (slow)	10.718	5.616	1500	success
baseline MPC (slower)	1.866	0.026	2500	success

Ant path tracking: In this task, an Ant robot is controlled to follow a "2"-shape reference path. The action being optimized is the torque applied to each joint actuator. We test the tracking performances of signature control and baseline standard MPC on this problem. Also, we run soft actor-critic (SAC) (Haarnoja et al., 2018) RL algorithm where the state is augmented with time index to manage waypoints and the reward (negative cost) is the same as that of the baseline MPC. For the baseline MPC and SAC, we equally distribute 880 waypoints to be tracked along the path and time stamp is determined by equally dividing the total tracking time achieved by signature control. Table 1 compares the mean/variance of deviation (measured by distance in meter) from the closest of 2000 points over the reference, and Figure 3 (Bottom Left) shows the resulting behaviors of MPCs, showing significant advantages of our method in terms of tracking accuracy. The performance of SAC RL is insufficient as we have no access to sophisticated waypoints over joints (see Peng et al. (2018) for the discussion). When more time steps are used, baseline MPC becomes a bit better. Note our method can tune the trade-off between accuracy and progress through regularizer.

Robotic manipulator path tracking: In this task, a robotic manipulator is controlled to track an end-effector reference path. Similar to the Ant task, 270 waypoints are equally sampled along the reference path for the baseline MPC method and SAC RL to track. To show robustness of signature control against unknown disturbance (torque: $\mathrm{N} \cdot \mathrm{m}$), we test different scales of disturbance force applied to each joint of the arm. The means/variances of the tracking deviation of the three approaches for selected cases are reported in Table 2 and the tracking paths are visualized in Figure 3 (Bottom Right). For all cases, our signature control outperforms the baseline MPC and SAC RL, especially the difference becomes much clearer when the disturbance becomes larger. This is because the signature MPC is insensitive to waypoint designs but rather depends on the "distance" between the target and the rollout paths in the signature space, making the tracking speed adaptive.

7. Discussions

This work presented signature control, a novel framework that generalizes value-based dynamic programming to reason over entire paths through Chen equation. There are many promising avenues for future work (which is relevant to the current limitations of our work), such as developing a more complete theoretical understanding of guarantees provided by the signature control framework, and developing additional RL algorithms that inherit the benefits of our signature control framework.

Table 2: Results on path tracking with a robotic manipulator end-effector. Comparing signature control, and baseline MPC and SAC RL with equally assigned waypoints under unknown fixed disturbance.

		Deviation (distance) from reference	
	Disturbance ($\mathrm{N} \cdot \mathrm{m}$)	Mean ($10^{-2}\mathrm{m}$)	Variance ($10^{-2}\mathrm{m}$)
signature control	$+30$	$\mathbf{1.674}$	$\mathbf{0.002}$
	± 0	$\mathbf{0.458}$	$\mathbf{0.001}$
	-30	$\mathbf{1.255}$	$\mathbf{0.002}$
baseline MPC	$+30$	2.648	0.015
	± 0	0.612	0.007
	-30	5.803	0.209
SAC RL	$+30$	15.669	0.405
	± 0	3.853	0.052
	-30	16.019	0.743

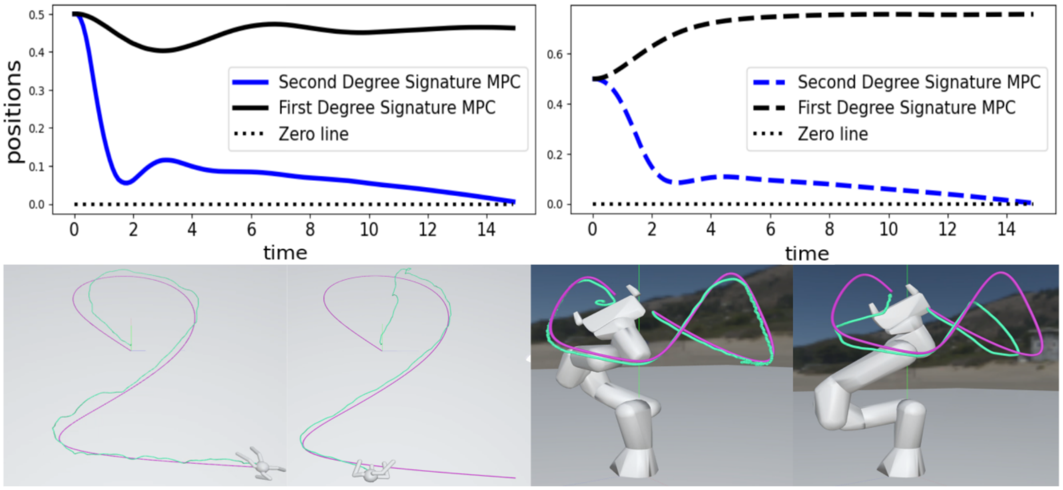

Figure 3: Top (two-mass spring, damper system): the plots show the evolutions of positions of two masses for signature MPC with/without second depth signature terms, showing how signature MPC reduces to integral control. Down: (Left two; Ant) tracking behaviors of signature control (left) and baseline MPC (right) for the same reaching time, where green lines are the executed trajectories. (Right two; Robotic arm): tracking behaviors of signature control (left) and baseline MPC (right) under disturbance -30.

While we emphasize that the run times of MPC algorithms used in this work for signature control and baseline are almost the same, adopting some of the state-of-the-art MPC algorithm running in real-time to our signature MPC is an important future work.

Acknowledgments

We thank the anonymous reviewers for improving this work.

References

A. P. Aguiar and J. P. Hespanha. Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty. *IEEE Trans. Automatic Control*, 52(8):1362–1379, 2007.

B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

L. Arnold. *Random dynamical systems*. Springer, 1998.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

I. P. Arribas. Derivatives pricing using signature payoffs. *arXiv preprint arXiv:1809.09466*, 2018.

A. Barreto, W. Dabney, R. Munos, J. J. Hunt, T. Schaul, H. P. van Hasselt, and D. Silver. Successor features for transfer in reinforcement learning. *Advances in Neural Information Processing Systems*, 30, 2017.

R. Bellman. An introduction to the theory of dynamic programming. Technical report, The Rand Corporation, Santa Monica, Calif., 1953.

R. Bellman. A Markovian decision process. *Journal of Mathematics and Mechanics*, pages 679–684, 1957.

H. Boedihardjo, X. Geng, T. Lyons, and D. Yang. The signature of a rough path: uniqueness. *Advances in Mathematics*, 293:720–737, 2016.

E. F. Camacho and C. B. Alba. *Model predictive control*. Springer Science & Business Media, 2013.

T. Cass, T. Lyons, and X. Xu. General signature kernels. *arXiv preprint arXiv:2107.00447*, 2021.

K. Chen. Iterated integrals and exponential homomorphisms. *Proceedings of the London Mathematical Society*, 3(1):502–512, 1954.

I. Chevyrev and A. Kormilitzin. A primer on the signature method in machine learning. *arXiv preprint arXiv:1603.03788*, 2016.

K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in Neural Information Processing Systems*, 31, 2018.

S. Du, S. Kakade, J. Lee, S. Lovett, G. Mahajan, W. Sun, and R. Wang. Bilinear classes: A structural framework for provable generalization in RL. In *International Conference on Machine Learning*, pages 2826–2836. PMLR, 2021.

A. Fermanian. *Learning time-dependent data with the signature transform*. PhD thesis, Sorbonne Université, 2021.

C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez. Integrated task and motion planning. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:265–293, 2021.

L. G. Gyurkó, T. Lyons, M. Kontkowski, and J. Field. Extracting information from the signature of a financial data stream. *arXiv preprint arXiv:1307.7244*, 2013.

T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

T. Hofmann, B. Schölkopf, and A. J. Smola. Kernel methods in machine learning. 2008.

A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

L. P. Kaelbling and T. Lozano-Pérez. Hierarchical task and motion planning in the now. In *IEEE International Conference on Robotics and Automation*, pages 1470–1477, 2011.

L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

S. M. Kakade. *On the sample complexity of reinforcement learning*. University of London, University College London (United Kingdom), 2003.

S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894, 2011.

H. K. Khalil. *Nonlinear systems; 3rd ed.* 2002.

P. Kidger, P. Bonnier, I. Perez A., C. Salvi, and T. Lyons. Deep signature transforms. *Advances in Neural Information Processing Systems*, 32, 2019.

F. J. Király and H. Oberhauser. Kernels for sequentially ordered data. *Journal of Machine Learning Research*, 20, 2019.

D. Levin, T. Lyons, and H. Ni. Learning from the past, predicting the statistics for the future, learning an evolving system. *arXiv preprint arXiv:1309.0260*, 2013.

T. Lyons. Rough paths, signatures and the modelling of functions on streams. *arXiv preprint arXiv:1405.4537*, 2014.

T. J. Lyons. Differential equations driven by rough signals. *Revista Matemática Iberoamericana*, 14 (2):215–310, 1998.

T. J. Lyons and N. Sidorova. Sound compression–a rough path approach. In *Proceedings of the 4th International Symposium on Information and Communication Technologies*, 2005.

T. J. Lyons, M. Caruana, and T. Lévy. *Differential equations driven by rough paths*. Springer, 2007.

T. M. Moerland, J. Broekens, A. Plaat, C. M. Jonker, et al. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1):1–118, 2023.

J. Morrill, C. Salvi, P. Kidger, and J. Foster. Neural rough differential equations for long time series. In *International Conference on Machine Learning*, pages 7829–7838. PMLR, 2021.

M. Ohnishi, I. Ishikawa, K. Lowrey, M. Ikeda, S. Kakade, and Y. Kawahara. Koopman spectrum nonlinear regulator and provably efficient online learning. *arXiv preprint arXiv:2106.15775*, 2021.

B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Trans. Intelligent Vehicles*, 1(1):33–55, 2016.

B. K. Patle, A. Pandey, D. R. K. Parhi, A. J. D. T. Jagadeesh, et al. A review: On path planning strategies for navigation of mobile robot. *Defence Technology*, 15(4):582–606, 2019.

X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne. DeepMimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018.

M. Rokonuzzaman, N. Mohajer, S. Nahavandi, and S. Mohamed. Review and performance evaluation of path tracking controllers of autonomous vehicles. *IET Intelligent Transport Systems*, 15(5): 646–670, 2021.

C. Salvi, T. Cass, J. Foster, T. Lyons, and W. Yang. The signature kernel is the solution of a Goursat PDE. *SIAM Journal on Mathematics of Data Science*, 3(3):873–899, 2021.

L. L. Scharf, W. P. Harthill, and P. H. Moose. A comparison of expected flight times for intercept and pure pursuit missiles. *IEEE Trans. Aerospace and Electronic Systems*, (4):672–673, 1969.

W. Schwarting, J. Alonso-Mora, and D. Rus. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:187–210, 2018.

W. Sun, N. Jiang, A. Krishnamurthy, A. Agarwal, and J. Langford. Model-based RL in contextual decision processes: PAC bounds and exponential improvements over model-free approaches. In *Conference on Learning Theory*, pages 2898–2933. PMLR, 2019.

R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, et al. Stanley: The robot that won the DARPA grand challenge. *The 2005 DARPA grand challenge: the great robot race*, pages 1–43, 2007.

T. Wang, X. Bao, I. Clavera, J. Hoang, Y. Wen, E. Langlois, S. Zhang, G. Zhang, P. Abbeel, and J. Ba. Benchmarking model-based reinforcement learning. *arXiv preprint arXiv:1907.02057*, 2019.

Z. Xie, Z. Sun, L. Jin, H. Ni, and T. Lyons. Learning spatial-semantic context with fully convolutional recurrent network for online handwritten Chinese text recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 40(8):1903–1917, 2017.

W. Yang, T. Lyons, H. Ni, C. Schmid, and L. Jin. Developing the path signature methodology and its application to landmark-based human action recognition. In *Stochastic Analysis, Filtering, and Stochastic Optimization: A Commemorative Volume to Honor Mark HA Davis's Contributions*, pages 431–464. Springer, 2022.

D. Zhou and M. Schwager. Vector field following for quadrotors using differential flatness. In *IEEE International Conference on Robotics and Automation*, pages 6567–6572, 2014.