1–49, 2023

Safely Learning Dynamical Systems

Amir Ali Ahmadi AAA@PRINCETON.EDU
Abraar Chaudhry AZC@PRINCETON.EDU
Princeton University

Vikas Sindhwani SINDHWANI@GOOGLE.COM
Stephen Tu STEPHENTU@GOOGLE.COM
Robotics at Google, New York

Abstract

A fundamental challenge in learning an unknown dynamical system is to reduce model uncertainty by making measurements while maintaining safety. In this work, we formulate a mathematical definition of what it means to safely learn a dynamical system by sequentially deciding where to initialize the next trajectory. In our framework, the state of the system is required to stay within a safety region for a horizon of T time steps under the action of all dynamical systems that (i) belong to a given initial uncertainty set, and (ii) are consistent with the information gathered so far.

For our first set of results, we consider the setting of safely learning a linear dynamical system involving n states. For the case $T = 1$, we present a linear programming-based algorithm that either safely recovers the true dynamics from at most n trajectories, or certifies that safe learning is impossible. For $T = 2$, we give a semidefinite representation of the set of safe initial conditions and show that $\lceil n/2 \rceil$ trajectories generically suffice for safe learning. Finally, for $T = \infty$, we provide semidefinite representable inner approximations of the set of safe initial conditions and show that one trajectory generically suffices for safe learning.

Our second set of results concerns the problem of safely learning a general class of nonlinear dynamical systems. For the case $T = 1$, we give a second-order cone programming based representation of the set of safe initial conditions. For $T = \infty$, we provide semidefinite representable inner approximations to the set of safe initial conditions. We show how one can safely collect trajectories and fit a polynomial model of the nonlinear dynamics that is consistent with the initial uncertainty set and best agrees with the observations.

Keywords: learning dynamical systems, safe learning, uncertainty quantification, robust optimization, conic optimization

1. Problem Formulation and Outline of Contributions

In many applications such as robotics, autonomous systems, and safety-critical control, one needs to learn a model of a dynamical system by observing a small set of its trajectories in a safe manner. This model can serve as a tool for making predictions about unobserved trajectories of the system. It can also be used for accomplishing downstream control objectives. Often, an important challenge during the initial stages of learning is that deploying even a conservative learning strategy on a real world system, such as a robot, is fraught with risk. How should the robot be "set loose" (i.e., initialized) in the real world so that our uncertainty about its dynamics is reduced, but with guarantees that the robot will remain safe (e.g., it does not exit a pre-specified region in state space)? How much more aggressive can our learning strategy get "on the fly" as uncertainty is reduced? This interplay between *safety and uncertainty while learning dynamical systems* is the central theme of this paper. We propose a mathematical formulation that captures the essence of this interplay and study the optimization problems that arise from the formulation in several settings.

The central object of our mathematical framework is a discrete-time dynamical system

$$x_{t+1} = f_\star(x_t), \tag{1}$$

where $f_\star : \mathbb{R}^n \to \mathbb{R}^n$ is an *unknown* map. This could be either a naturally arising autonomous system, or a closed-loop control system with a fixed feedback policy. Our interest is in the problem of safe data acquisition for estimating the unknown map $f_\star$ from a collection of length-T trajectories $\{\phi_{f_\star,T}(x_j)\}_{j=1}^m$, where $\phi_{f,T}(x) := (x, f(x), \dots, f^{(T)}(x))$.

In our setting, we are given as input a set $S \subset \mathbb{R}^n$, called the *safety region*, in which the state should remain throughout the learning process. We say that a state x is T*-step safe* under a map $f : \mathbb{R}^n \to \mathbb{R}^n$ if $f^{(i)}(x) \in S$ for all $i = 0, \dots, T$. We define $S^T(f) \subseteq S$ to be the set of states that are T-step safe under f. In order to safely learn $f_\star$, we require that measurements are made only at points in $S^T(f_\star)$. Obviously, if we make no assumptions about $f_\star$, this task is impossible. We assume, therefore, that the map $f_\star$ belongs to a set of dynamics U_0, which we call the *initial uncertainty set*. As experience is gathered, the uncertainty over $f_\star$ decreases. Let us denote the uncertainty set after we have observed k trajectories $\{\phi_{f_\star,T}(x_j)\}_{j=1}^k$ by,

$$U_k := \{f \in U_0 \mid \phi_{f,T}(x_j) = \phi_{f_\star,T}(x_j)\,, j = 1, \dots, k\}.$$

Observe that $U_{k+1} \subseteq U_k$ for all k. For a nonnegative integer k, define

$$S_k^T := \bigcap_{f \in U_k} S^T(f)\,,$$

the set of points that are T-step safe under all dynamics consistent with the initial uncertainty set and the data after observing k trajectories. We refer to the set S_k^T as the T*-step safe set* (the dependence on k is implicit). Note that $S_k^T \subseteq S_{k+1}^T$ for all k. A primary goal of this paper is to characterize the sets S_k^T as feasible regions of tractable optimization problems. In certain settings where an exact tractable characterization is not possible, our goal would be to find tractable inner approximations of these sets. For robustness reasons, we would like these inner approximations to be full-dimensional so that safe queries to the system can be made while tolerating perturbations which may arise during implementation.

A secondary goal of this paper is to provide algorithms for what we define as the T*-step safe learning problem*. Fix a scalar $\bar{\varepsilon} > 0$ and a norm $||.||$ on $\mathbb{R}^n$. Given a safety region $S \subset \mathbb{R}^n$ and an initial uncertainty set U_0, the T-step safe learning problem (up to accuracy $\bar{\varepsilon}$ and with respect to norm $||.||$) is to sequentially choose vectors $x_1, \dots, x_m$, for some nonnegative integer m, such that:

1. **(Safety)** for each $k = 1, \dots, m$, $x_k \in S_{k-1}^T$,
2. **(Learning)** $\sup_{f \in U_m, x \in S^T(f_\star)} \|f(x) - f_\star(x)\| \le \bar{\varepsilon}$.

If for a given T, no such sequence of vectors $x_1, \dots, x_m$ exists (for any m), we say that T-step safe learning is impossible. Note that if T-step safe learning is possible, then T'-step safe learning is also possible for any $T' < T$. Moreover, since the highest rate of safe information assimilation is achieved when $T = 1$, to prove that safe learning is impossible for any T, it is necessary and sufficient to prove its impossibility for $T = 1$.

In many situations, the choice of the sequence of $\{x_1, \dots, x_m\}$ that achieves T-step safe learning may not be unique. We further suppose that for a function $c : \mathbb{R}^n \mapsto \mathbb{R}$ that takes nonnegative values over S, initializing the unknown system at a state $x \in S$ comes at a cost of $c(x)$. In such a setting, we are interested in safely learning the dynamical system at minimum total initializtion cost. Ideally, we wish to minimize $\sum_{k=1}^{m} c(x_k)$ over sequences $\{x_1, \ldots, x_m\}$ that satisfy the safe learning conditions 1 and 2 above. However, such an optimization problem cannot be solved without knowing the action of the true dynamics $f_\star$ on the initialization points $\{x_k\}$ ahead of time. Hence, a natural online algorithm is to sequentially solve the following greedy optimization problem

$$\min_{x \in S_{k-1}^T} \ c(x) \ , \tag{2}$$

whose optimal solution gives the next cheapest T-step safe initialization point x_k, given information gathered before time k. A byproduct of our primary goal of characterizing the sets S_k^T tractably is efficient algorithms for solving the optimization problem (2).

Contributions. In this paper, we derive tractable conic programs that exactly characterize or inner approximate T-step safe sets (for any k) for both linear systems and a general class of nonlinear systems in the extreme cases when $T = 1$ and $T = \infty$. For linear systems, we also address the case when $T = 2$, and provide algorithms for solving the exact (i.e., $\bar{\varepsilon} = 0$) T-step safe learning problem when $T = 1, 2, \infty$. Throughout the paper, we assume that the safety region S is a polyhedron.

More specifically, for linear systems, we give an exact linear programming-based characterization of the one-step safe set when U_0 is a polytope, and an exact semidefinite programming-based characterization of the two-step safe set when U_0 is an ellipsoid. Based on the former characterization, we present a linear programming-based algorithm that either learns the unknown dynamics by making at most n one-step safe queries, or certifies the impossibility of safe learning (for any T). In the case of $T = 2$, we show that under mild assumptions, $\lceil \frac{n}{2} \rceil$ trajectories (whose initializations are computed by semidefinite programming) suffice for safe learning. Roughly speaking, these algorithms sequentially solve (2) and add appropriate safe perturbations to ensure that the remaining uncertainty U_k is shrinking. Finally, when $T = \infty$, under the assumption that U_0 is a compact subset of Schur-stable matrices, we present a sum of squares hierarchy of semidefinite programs that provide full-dimensional inner approximations of the infinite-step safe set. Under mild assumptions, we show that a single trajectory randomly initialized from our inner approximation suffices for safe learning.

Turning to nonlinear systems, we consider the case when the dynamics in (1) consists of a linear term plus a nonlinear function with bounded growth. When $T = 1$, we give an exact second-order cone programming-based representation of the safe set when the uncertainty around the linear dynamics is represented by a polyhedron. When $T = \infty$, we provide a hierarchy of semidefinite representable inner approximations to the infinite-step safe set. Under the assumption that the nonlinear function growth is relatively small compared to the uncertainty around the linear part of the dynamics, we prove that our hierarchy provides a full-dimensional inner approximation. By using our safe set representations, we show how one can safely collect trajectories to refine uncertainty regarding the linear term of the dynamics and fit a polynomial model of the nonlinear dynamics that is consistent with the initial uncertainty set and best agrees with the observations.

Outline. In Section 2, we cover the relevant literature around safe learning and control. Section 3, Section 4, and Section 5 present our results and algorithms for safely learning linear systems when $T = 1, 2, \infty$, respectively. Section 6 and Section 7 contain our results for nonlinear systems when $T = 1$ and $T = \infty$, respectively. Future directions for research are presented in Section 8. All omitted proofs can be found in Appendix A.

2. Related Work

The idea of using conic and robust optimization techniques for verifying various properties of a known dynamical system has been the focus of much research in the control and optimization communities (Parrilo, 2000; Lasserre, 2010; Boyd et al., 1994; Blekherman et al., 2013). Our work borrows some of these techniques to instead learn a dynamical system from data subject to certain safety constraints. Learning dynamical systems from data is an important problem in the field of system identification; see, e.g., Åström and Eykhoff (1971); Keesman (2011); Brunton and Kutz (2019), and references therein.

The problem of additionally accounting for safety constraints during system identification has recently gained attention; see, e.g., Brunke et al. (2021) for an excellent survey of this growing research field. Here, we highlight a few of the key technical tools used: Gaussian process models (Akametalu et al., 2014; Berkenkamp and Schoellig, 2015; Berkenkamp et al., 2017), control barrier functions (Cheng et al., 2019; Taylor et al., 2020; Luo et al., 2021), set invariance and uncertainty propagation (Artstein and Raković, 2008; Gurriet et al., 2019; Koller et al., 2019), "safety critics" in reinforcement learning (Zhang et al., 2020; Bharadhwaj et al., 2021), and backup controllers (Mannucci et al., 2018; Wabersich and Zeilinger, 2021).

We highlight two related works for safely learning linear dynamical systems that, similar to our work, rely on optimization formulations to ensure safety. The first is the work of Dean et al. (2019), which uses convex programming methods to approximate the solution to a finite-time horizon linear-quadratic optimal control problem with both model uncertainty and state/action constraints. The feasible sets of these convex programs can be repurposed to provide inner approximations to the set of initial conditions that remain T-step safe (in the language of our paper). While the problem setting of Dean et al. (2019) includes both control and bounded disturbances (and is hence more general than what we consider), their inner approximations are in general conservative, as they are of the form of ℓ_∞-balls around some fixed center in S_0^T. In comparison, our results provide exact characterizations of the T-step safety sets for $T = 1, 2$. In addition, the work in Dean et al. (2019) does not provide an implementable algorithm for handling the case when $T = \infty$ since the size of their convex programs grow with the horizon length T. We note that follow-up work (Chen et al., 2021, 2022) provides better heuristics for constructing inner approximations to the safety regions, but again these approximations are not exact.

The second related paper is that of Lu et al. (2017). In this work, the authors address the problem of "one-step safety" and "trajectory safety" in a probabilistic framework. While similar sounding to our problem setup, in their work the initial condition x_0 is *fixed* and the question of either characterizing or inner approximating the T-step safety sets S_0^T is not addressed. Furthermore, the proposed algorithm in the T-step setting requires a separate computation to check safety for every time step $t \in \{1, \ldots, T\}$, and hence, similar to Dean et al. (2019), cannot be implemented in the $T = \infty$ setting. The algorithm also requires checking safety for each coordinate of the state separately and relies on nonconvex optimization without optimality guarantees.

We would like to also highlight some papers on the topic of learning to stabilize dynamical systems, which has recently gained attention. The work of Dean et al. (2020) studies linear systems with noise and shows how to learn a stabilizing controller efficiently, both with respect to the number of required samples and cost. The work of Werner and Peherstorfer (2023) shows that one can stabilize linear noiseless systems with less data than would be necessary to learn the system parameters exactly. The work of Guo et al. (2022) shows how to search for stabilizing controllers and associated

Lyapunov functions for all continuous-time polynomial systems that are consistent with collected data. In a follow-up paper, Bisoffi et al. (2022) use Petersen's lemma to further develop the method and prove a necessary and sufficient condition for data-driven stabilization of linear systems. All these methods lead to stabilized systems wherein the state will remain bounded, however they do not consider explicit safety constraints. In another follow-up paper, Luppi et al. (2021) incorporate safety constraints into the framework of Guo et al. (2022) and Bisoffi et al. (2022) and show how to find approximations of the continuous time analogue of S_k^∞. This work is concerned with stabilization of polynomial vector fields and is focused on the $T = \infty$ case. The approach is specific to data observation models which lead to "matrix ellipsoidal" uncertainty sets (see Bisoffi et al. (2022) for a definition) and does not consider the problems of trajectory initialization and its cost. The work is instead focused on the design of controllers which produce invariant subsets of the safety region that are defined as sublevel sets of polynomials. By contrast, the invariant sets that our work produces (for a different class of nonlinear dynamics) are semidefinite representable and hence can be optimized over efficiently.

We end by noting that our work has some conceptual connections to the literature on experiment design (see, e.g., Pukelsheim, 2006; De Castro et al., 2019). However, this literature typically does not consider dynamical systems or notions of safety.

A much shorter version of this work containing preliminary results on one-step and two-step safe learning has appeared in Ahmadi et al. (2021).

3. One-Step Safe Learning of Linear Systems

In this section, we focus on characterizing one-step safe learning for linear systems. Here, the state evolves according to

$$x_{t+1} = A_\star x_t, \tag{3}$$

where $A_\star$ is an unknown $n \times n$ matrix. We assume we know that $A_\star$ belongs to a set $U_0 \subset \mathbb{R}^{n\times n}$ that represents our prior knowledge of $A_\star$. In this section, we take U_0 to be a polyhedron; i.e.,

$$U_0 = \left\{A \in \mathbb{R}^{n\times n} \mid \mathrm{Tr}(V_j^T A) \leq v_j \quad j = 1, \dots, s\right\} \tag{4}$$

for some matrices $V_1, \dots, V_s \in \mathbb{R}^{n\times n}$ and scalars $v_1, \dots, v_s \in \mathbb{R}$. We also work with a polyhedral representation of the safety region S; i.e.,

$$S = \left\{x \in \mathbb{R}^n \mid h_i^T x \leq b_i \quad i = 1, \dots, r\right\} \tag{5}$$

for some vectors $h_1, \dots, h_r \in \mathbb{R}^n$ and some scalars $b_1, \dots, b_r \in \mathbb{R}$. We assume that initializing the system at a point $x \in \mathbb{R}^n$ comes at a cost $c^T x$, for some given vector $c \in \mathbb{R}^n$. In practice, initialization costs are nonnegative. Since the set S is often compact in applications, one can add a constant term to $c^T x$ to ensure this requirement without changing any of our optimization problems. We ignore this constant term in our formulations and examples. Our algorithms tractably extend to any semidefinite-representable cost function (see Ben-Tal and Nemirovski (2001) for a definition) $c : \mathbb{R}^n \mapsto \mathbb{R}$ by replacing the objective function with a new variable β and adding the constraint $c(x) \leq \beta$.

We start by finding the minimum cost point that is one-step safe under all valid dynamics, i.e., a point $x \in S$ such that $Ax \in S$ for all $A \in U_0$. Once this is done, we gain further information by observing the action $y = A_\star x$ of system (3) on our point x, which further constrains the uncertainty set U_0. We then repeat this procedure with the updated uncertainty set to find the next minimum cost one-step safe point. More generally, after collecting k measurements, our uncertainty in the dynamics reduces to the set

$$U_k = \{A \in U_0 \mid Ax_j = y_j \quad j = 1, \dots, k\}. \tag{6}$$

Hence, the problem of finding the next cheapest one-step safe initialization point becomes:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^T x \\ \text{s.t.} \quad & x \in S \\ & Ax \in S \quad \forall A \in U_k. \end{aligned} \tag{7}$$

In Section 3.1, we show that problem (7) can be efficiently solved. We then use (7) as a subroutine in a one-step safe learning algorithm which we present in Section 3.2.

3.1. Reformulation via Duality

In this subsection, we reformulate problem (7) as a linear program. To do this, we introduce auxiliary variables $\mu_j^{(i)} \in \mathbb{R}$ and $\eta_k^{(i)} \in \mathbb{R}^n$ for $i = 1, \dots, r$, $j = 1, \dots, s$, and $k = 1, \dots, m$.

Theorem 1 *The feasible set of problem (7) is the projection to x-space of the feasible set of the following linear program:*

$$\begin{aligned} \min_{x,\mu,\eta} \quad & c^T x \\ \textit{s.t.} \quad & h_i^T x \le b_i \quad i = 1, \dots, r \\ & \sum_{k=1}^m y_k^T \eta_k^{(i)} + \sum_{j=1}^s \mu_j^{(i)} v_j \le b_i \quad i = 1, \dots, r \\ & x h_i^T = \sum_{k=1}^m x_k \eta_k^{(i)T} + \sum_{j=1}^s \mu_j^{(i)} V_j^T \quad i = 1, \dots, r \\ & \mu^{(i)} \ge 0 \quad i = 1, \dots, r. \end{aligned} \tag{8}$$

In particular, the optimal values of (7) and (8) are the same and the optimal solutions of (7) are the optimal solutions of (8) projected to x-space.

Proof Using the definitions of S and U_0, let us first rewrite (7) as a bilevel program:

$$\begin{aligned} \min_{x} \quad & c^T x \\ \text{s.t.} \quad & h_i^T x \le b_i \quad i = 1, \dots, r \\ & \begin{bmatrix} \max_A & h_i^T A x \\ \text{s.t.} & \mathrm{Tr}(V_j^T A) \le v_j \quad j = 1, \dots, s \\ & A x_k = y_k \quad k = 1, \dots, m \end{bmatrix} \le b_i \quad i = 1, \dots, r. \end{aligned} \tag{9}$$

We proceed by taking the dual of the r inner programs, treating the x variable as fixed. By introducing dual variables $\mu_j^{(i)}$ and $\eta_k^{(i)}$ for $i = 1, \dots, r$, $j = 1, \dots, s$, and $k = 1, \dots, m$, and by invoking strong duality of linear programming, we have

$$\begin{bmatrix} \max_A & h_i^T Ax \\ \text{s.t.} & \mathrm{Tr}(V_j^T A) \le v_j \; j = 1, \ldots, s \\ & Ax_k = y_k \; k = 1, \ldots, m \end{bmatrix} = \begin{bmatrix} \min_{\mu^{(i)}, \eta^{(i)}} & \sum_{k=1}^m y_k^T \eta_k^{(i)} + \sum_{j=1}^s \mu_j^{(i)} v_j \\ \text{s.t.} & x h_i^T = \sum_{k=1}^m x_k \eta_k^{(i)T} + \sum_{j=1}^s \mu_j^{(i)} V_j^T \\ & \mu^{(i)} \ge 0 \end{bmatrix} \tag{10}$$

for $i = 1, \ldots, r$. Thus by replacing the inner problem of (9) with the right-hand side of (10), the min-max problem (9) becomes a min-min problem. This min-min problem can be combined into a single minimization problem and be written as problem (8). Indeed, if x is feasible to (9), for that fixed x and for each i, there exist values of $\mu^{(i)}$ and $\eta^{(i)}$ that attain the optimal value for (10) and therefore the triple (x, μ, η) will be feasible to (8). Conversely, if some (x, μ, η) is feasible to (8), it follows that x is feasible to (9). This is because for any fixed x and for each i, the optimal value of the left-hand side of (10) is bounded from above by the objective value of the right-hand side evaluated at any feasible $\mu^{(i)}$ and $\eta^{(i)}$. ■

Remark 2 *We note that (8) can be modified so that one-step safety is achieved in the presence of disturbances. We can ensure, e.g., using linear programming, that $Ax + w \in S$ for all $A \in U_m$ and all w such that $\|w\| \le W$, where $\|\cdot\|$ is any norm whose unit ball is a polytope and W is a given scalar.*

3.2. An Algorithm for One-Step Safe Learning

We start by giving a mathematical definition of (exact) safe learning specialized to the case of one-step safety and linear dynamics. Recall the definition of the set U_k in (6).

Definition 3 (One-Step Safe Learning) *We say that one-step safe learning is possible if for some nonnegative integer m, we can sequentially choose vectors $x_k \in S$, for $k = 1, \ldots, m$, and observe measurements $y_k = A_\star x_k$ such that:*

1. ***(Safety)*** *for $k = 1, \ldots, m$, we have $Ax_k \in S \quad \forall A \in U_{k-1}$,*
2. ***(Learning)*** *the set of matrices U_m is a singleton.*

We now present our algorithm for checking the possibility of one-step safe learning (Algorithm 1). The proof of correctness of Algorithm 1 is given in Theorem 7.

Remark 4 *As Theorem 7 will demonstrate, the particular choice of the parameter $\varepsilon \in (0, 1]$ in the input to Algorithm 1 does not affect the detection of one-step safe learning by this algorithm. However, a smaller ε leads to a lower cost of learning. Therefore, in practice, ε should be chosen positive and as small as possible without causing the matrix X in line 25 to be ill conditioned.*

Algorithm 1 invokes two subroutines which we present next in Lemma 5 and Lemma 6.

Lemma 5 *Let $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{m \times p}$, $c \in \mathbb{R}^m$, and define the polyhedron*

$$P := \{x \in \mathbb{R}^n \mid \exists y \in \mathbb{R}^p \quad \text{s.t.} \quad Ax + By \le c\}.$$

The problem of checking if P is a singleton can be reduced to solving $2n$ linear programs.

Algorithm 1: One-Step Safe Learning Algorithm

Input : polyhedra $S \subset \mathbb{R}^n$ and $U_0 \subset \mathbb{R}^{n\times n}$, cost vector $c \in \mathbb{R}^n$, and a constant $\varepsilon \in (0, 1]$.
Output: A matrix $A_\star \in \mathbb{R}^{n\times n}$ or a declaration that one-step safe learning is impossible.

for $k = 0, \ldots, n-1$ **do**
  $D_k \leftarrow \{(x_j, y_j) \mid j = 1, \ldots, k\}$
  $U_k \leftarrow \{A \in U_0 \mid Ax_j = y_j, \quad j = 1, \ldots, k\}$
  if *U_k is a singleton (cf. Lemma 5)* **then**
    return the single element in U_k as $A_\star$
  end
  Let $x_k^\star$ be the projection to x-space of an optimal solution to problem (8) with data D_k
  if *$x_k^\star$ is linearly independent from $\{x_1, \ldots, x_k\}$* **then**
    $x_{k+1} \leftarrow x_k^\star$
  else
    Let S_k^1 be the projection to x-space of the feasible region of problem (8) with data D_k
    Compute a basis $B_k \subset S_k^1$ of span(S_k^1) (cf. Lemma 6)
    for $z_j \in B_k$ **do**
      if *z_j is linearly independent from $\{x_1, \ldots, x_k\}$* **then**
        $x_{k+1} \leftarrow (1-\varepsilon)x_k^\star + \varepsilon z_j$
        break
      end
    end
    if *no $z_j \in B_k$ is linearly independent from $\{x_1, \ldots, x_k\}$* **then**
      return one-step safe learning is impossible
    end
  end
  Observe $y_{k+1} \leftarrow A_\star x_{k+1}$
end
Define matrix $X = [x_1, \ldots, x_n]$
Define matrix $Y = [y_1, \ldots, y_n]$
return $A_\star = YX^{-1}$

Proof For each $i = 1, \ldots, n$, maximize and minimize the i-th coordinate of x over P. It is straightforward to check that P is a singleton if and only if the optimal values of these two linear programs coincide for every $i = 1, \ldots, n$. ■

In the next lemma, the notation span(P) denotes the set of all linear combinations of points in a set $P \subseteq \mathbb{R}^n$ (see the appendix for a proof of this lemma).

Lemma 6 *Let $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times p}$, $c \in \mathbb{R}^m$, and define the polyhedron*

$$P := \{x \in \mathbb{R}^n \mid \exists y \in \mathbb{R}^p \quad \text{s.t.} \quad Ax + By \leq c\}.$$

One can find a basis of span(P) *contained within P by solving at most $2n^2$ linear programs.*

Our next theorem is the main result of the section.

Theorem 7 *Given a safety region $S \subset \mathbb{R}^n$ and an uncertainty set $U_0 \subset \mathbb{R}^{n\times n}$, one-step safe learning is possible if and only if Algorithm 1 (with an arbitrary choice of $c \in \mathbb{R}^n$ and $\varepsilon \in (0, 1]$) returns a matrix.*

Proof ["If"] By construction, the sequence of initialization points chosen by Algorithm 1 satisfies the first condition of Definition 3, since the vectors $x_k^\star$ and z_j are both contained in S_k^1 and any vector in S_k^1 will remain in the safety region under the action of all matrices in U_k; i.e. all matrices in U_A that are consistent with the measurements made so far. If Algorithm 1 terminates early at line 5 for some iteration k, then clearly the uncertainty set U_k is a singleton. On the other hand, if we reach line 27, then we must have n linearly independent initialization points $\{x_1, \dots, x_n\}$. From this, it is clear that the set $\{A \in U_0 \mid Ax_j = y_j, j = 1, \dots, n\} = \{A_\star\}$.

["Only if"] Suppose Algorithm 1 chooses points $\{x_1, \dots, x_m\}$ where $m < n$ and terminates at line 20. Then it is clear from the algorithm that $\{x_1, \dots, x_m\}$ must form a basis of span(S_m^1) and that U_m is not a singleton. Take $\tilde{m}$ to be any nonnegative integer and $\{\tilde{x}_1, \dots, \tilde{x}_{\tilde{m}}\}$ to be any sequence that satisfies the first condition of Definition 3. For $k = 1, \dots, \tilde{m}$, let

$$\tilde{U}_k = \{A \in U_0 \mid A\tilde{x}_j = A_\star\tilde{x}_j, j = 1, \dots, k\} \,,$$
$$\tilde{S}_k^1 = \{x \in S \mid Ax \in S, \forall A \in \tilde{U}_k\} \,.$$

First we claim that $\tilde{x}_k \in S_m^1$ for $k = 1, \dots, \tilde{m}$. We show this by induction. It is clear that $\tilde{x}_1 \in S_m^1$ since $\tilde{x}_1 \in S_0^1$ and $S_0^1 \subseteq S_m^1$. Now we assume $\tilde{x}_1, \dots, \tilde{x}_k \in S_m^1$ and show that $\tilde{x}_{k+1} \in S_m^1$. Since $\{x_1, \dots, x_m\}$ forms a basis of span(S_m^1), it follows that for any matrix A, $Ax_j = A_\star x_j$ for $j = 1, \dots, m$ implies $Ax = A_\star x$ for all $x \in S_m^1$. In particular, for any matrix A, $Ax_j = A_\star x_j$ for $j = 1, \dots, m$ implies $A\tilde{x}_j = A_\star\tilde{x}_j$ for all $j = 1, \dots, k$. It follows that $U_m \subseteq \tilde{U}_k$ and therefore, $\tilde{S}_k^1 \subseteq S_m^1$. By the first condition of Definition 3, we must have $\tilde{x}_{k+1} \in \tilde{S}_k^1$, and thus, $\tilde{x}_{k+1} \in S_m^1$. This completes the inductive argument and shows that $\tilde{x}_k \in S_m^1$ for $k = 1, \dots, \tilde{m}$. From this, it follows that $U_m \subseteq \tilde{U}_{\tilde{m}}$. Recall that U_m is not a singleton, thus $\tilde{U}_{\tilde{m}}$ is not a singleton either. Therefore, the sequence $\{\tilde{x}_1, \dots, \tilde{x}_{\tilde{m}}\}$ does not satisfy the second condition of Definition 3. ■

Corollary 8 *Given a safety region $S \subset \mathbb{R}^n$ and an uncertainty set $U_0 \subset \mathbb{R}^{n\times n}$, if one-step safe learning is possible, then it is possible with at most n measurements.*

Note that if $A_\star$ belongs to the interior of U_0, any algorithm needs at least n measurements in order to learn $A_\star$.

3.3. The Value of Exploiting Information on the Fly

In addition to detecting the possibility of safe learning, Algorithm 1 attempts to minimize the overall cost of learning (i.e., $\sum_{k=1}^m c^T x_k$) by exploiting information gathered at every step. In order to demonstrate the value of using information online, we construct Algorithm 2 which chooses n initialization points $x_1, \dots, x_n$ ahead of time based solely on U_0 and S. This algorithm succeeds under the assumption that S_0^1 contains a basis of $\mathbb{R}^n$.

As ε tends to zero, the cost of Algorithm 2 approaches $nc^T x_0^\star$, where $x_0^\star$ is a minimum cost initialization point in S_0^1; therefore, $nc^T x_0^\star$ serves as an *upper bound* on the cost incurred by Algorithm 1. We note that $nc^T x_0^\star$ is also the minimum cost achievable by any one-step safe offline

Algorithm 2: Offline One-Step Safe Learning Algorithm

Input : polyhedra $S \subset \mathbb{R}^n$ and $U_0 \subset \mathbb{R}^{n \times n}$, cost vector $c \in \mathbb{R}^n$, and a constant $\varepsilon \in (0, 1]$.
Output: A matrix $A_\star \in \mathbb{R}^{n \times n}$ or failure.

if *S_0^1 does not contain a basis of $\mathbb{R}^n$ (cf. Lemma 6)* **then**
    return failure
end
Compute a basis $\{z_1, \ldots, z_n\} \subset S_0^1$ of $\mathbb{R}^n$
Let $x_0^\star$ be the projection to x-space of an optimal solution to problem (8) with data D_0
Set $x_k = (1 - \varepsilon)x_0^\star + \varepsilon z_k$ for $k = 1, \ldots, n$
Observe $y_k \leftarrow A_\star x_k$ for $k = 1, \ldots, n$
Define matrix $X = [x_1, \ldots, x_n]$
Define matrix $Y = [y_1, \ldots, y_n]$
return $A_\star = YX^{-1}$

algorithm that takes n measurements, since all initialization points $\{x_k\}$ of such an algorithm must come from S_0^1.

We refer the reader to Section 3.5 for a numerical example comparing Algorithm 1 and Algorithm 2, and to Section 3.6 for an example where exploiting online information is necessary for safe learning.

3.4. A Lower Bound on the Cost of Safe Learning

Consider a safety region $S \subset \mathbb{R}^n$, an initial uncertainty set $U_0 \subset \mathbb{R}^{n \times n}$, and an affine function $c : S \mapsto \mathbb{R}_+$. By assuming knowledge of the matrix $A_\star$ governing the true dynamics, we can express the minimum cost of safe learning (cf. the paragraph before Eq. (2)) over all possible (online or offline) algorithms as the optimal value of the following optimization problem:

$$
\begin{aligned}
\inf_{m \in \mathbb{N}, x_1, \ldots, x_m \in \mathbb{R}^n} \quad & \sum_{k=1}^{m} c(x_k) \\
\text{s.t.} \quad & x_k \in S \quad k = 1, \ldots, m \\
& Ax_1 \in S \quad \forall A \in U_0 \\
& Ax_2 \in S \quad \forall A \in \{A \in U_0 \mid Ax_1 = A_\star x_1\} \\
& Ax_3 \in S \quad \forall A \in \{A \in U_0 \mid Ax_1 = A_\star x_1, \quad Ax_2 = A_\star x_2\} \\
& \qquad \vdots \\
& Ax_m \in S \quad \forall A \in \{A \in U_0 \mid Ax_k = A_\star x_k \quad k = 1, \ldots, m-1\} \\
& \{A \in U_0 \mid Ax_k = A_\star x_k \quad k = 1, \ldots, m\} = \{A_\star\}.
\end{aligned}
\tag{11}
$$

For a fixed $m \in \mathbb{N}$, and assuming knowledge of $A_\star$, using a similar duality approach as in the proof of Theorem 1, the membership constraints in (11) can be written as bilinear constraints in $x_1, \ldots, x_m$ and additional dual variables. It is unclear however if (11) can be solved tractably (even for fixed m). Accordingly, we use the following easily computable lower bound on the minimum cost of safe learning for our numerical example in the next section. Suppose $A_\star$ is in the interior of U_0. Let

$$S^1(A_\star) = \{x \in S \mid A_\star x \in S\}$$

be the true one-step safety region of $A_\star$. Suppose $x^\star$ is an optimal solution to the linear program that minimizes $c(x)$ over $S^1(A_\star)$. Since one-step safe learning requires at least n measurements, we cannot achieve a cost lower than $nc(x^\star)$.

3.5. Numerical Example of One-Step Safe Learning

We present a numerical example with $n = 4$. Here, we take $U_0 = \{A \in \mathbb{R}^{4\times4} \mid |A_{ij}| \le 4 \quad \forall i, j\}$, $S = \{x \in \mathbb{R}^4 \mid \|x\|_\infty \le 1\}$, and $c = (-1, -1, 0, 0)^T$. We choose the matrix $A_\star$ uniformly at random among integer matrices in U_0

$$A_\star = \begin{bmatrix} 2 & 1 & 4 & 2 \\ 2 & -3 & -1 & -2 \\ -2 & -3 & 1 & 0 \\ 2 & 0 & -2 & 2 \end{bmatrix}.$$

In this example, Algorithm 1 takes four steps to safely recover $A_\star$. The projection to the first two dimensions of the four vectors that Algorithm 1 selects are plotted in Figure 1(*a*) (note that two of the points are very close to each other). Because of the cost vector c, points higher and further to the right in the plot have lower initialization cost. Also plotted in Figure 1(*a*) are the projections to the first two dimensions of the sets S_k^1 for $k \in \{0, 1, 2, 3\}$ and of the set $S^1(A_\star)$, the true one-step safety region of $A_\star$. In Figure 1(*b*), we plot U_k (the remaining uncertainty after making k measurements) for $k \in \{0, 1, 2, 3, 4\}$; we draw a two-dimensional projection of these sets of matrices by looking at the trace and the sum of the entries of each matrix in the set. Note that U_4 is a single point since we have recovered the true dynamics after the fourth measurement.

The cost of learning (i.e., $\sum_{i=1}^4 c_i^T x_i$) for the offline algorithm (Algorithm 2) approaches -1 as $\varepsilon \to 0$. The cost of learning for Algorithm 1 (with $\varepsilon = 0.01$) is -1.6385. The lower bound on the cost of learning is -2.2264 (cf. Section 3.4).[1] We can see that the value of exploiting information on the fly is significant.

3.6. Failure of Offline Learning

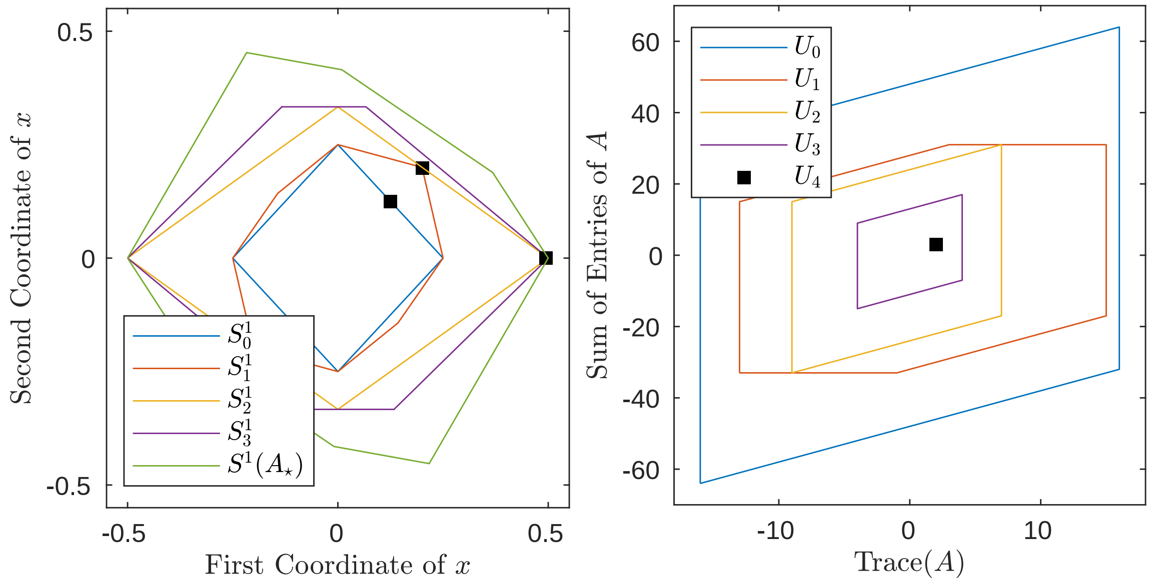
It is natural to ask if in every case that one-step safe learning is possible, whether it is also possible with an offline algorithm (i.e., an algorithm that can only sample points from S_0^1). In this subsection, we show that this is not the case, demonstrating the necessity of exploiting information on the fly.

Consider the following example with $n = 2$,

$$U_0 = \left\{ A \in \mathbb{R}^{2\times2} \middle| \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \le A \le \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \right\}, \quad S = [1, 3]^2,$$

an arbitrary cost vector c, and $A_\star = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. It is straightforward to see that no offline algorithm can recover $A_\star$. Indeed, one can check that (i) $S_0^1 = \{(1, 1)^T\}$, which does not contain a basis of

1. Note that adding a constant to the objective function to make it nonnegative over S, would shift all of our bounds by the same amount.



(a) S_k^1 grows with k.

(b) U_k shrinks with k.

Figure 1: One-step safe learning associated with the numerical example in Section 3.5.

$\mathbb{R}^2$, and (ii)

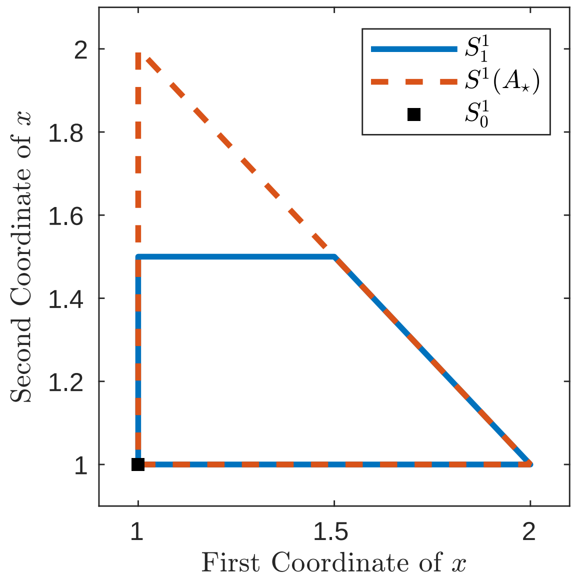
$$U_1 = \text{conv}\left(\left\{\begin{bmatrix}1 & 0\\0 & 2\end{bmatrix}, \begin{bmatrix}1 & 0\\1 & 1\end{bmatrix}\right\}\right)$$

which is not a singleton.

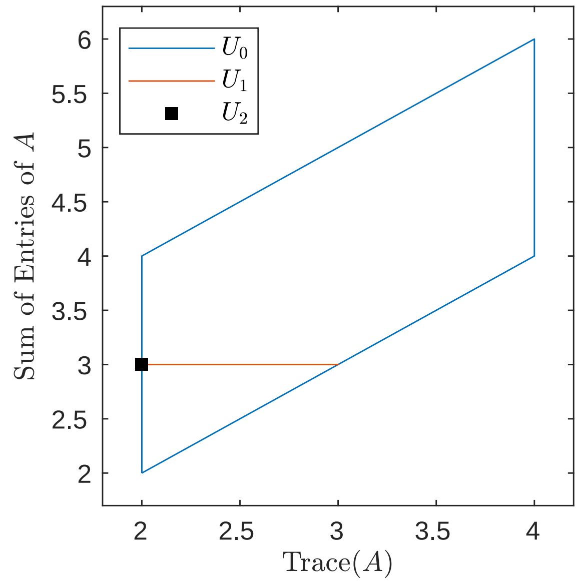
By contrast, Algorithm 1 takes two steps to safely recover $A_\star$, demonstrating that safe learning is possible. Plotted in Figure 2(a) are the sets S_k^1 for $k \in \{0,1\}$ and the set $S^1(A_\star)$, the true one-step safety region of $A_\star$. In Figure 2(b), we plot U_k (the remaining uncertainty after making k measurements) for $k \in \{0,1,2\}$; we draw a two-dimensional projection of these sets of matrices by plotting the trace and the sum of the entries of each matrix in the set. As noted previously, U_1 is not a singleton as we cannot exactly recover $A_\star$ from measuring its action on the single point in S_0^1. However, U_2 is a singleton since we have recovered the true dynamics after the second measurement. Thus, we see that the value of exploiting information on the fly is significant not just in terms of cost, but in terms of the possibility of learning as well.

4. Two-Step Safe Learning of Linear Systems

In this section, we again focus on learning the linear dynamics in (3). However, unlike the previous section, we are interested in making queries to the system that are two-step safe. An advantage of this formulation is that we may have fewer system resets and can potentially learn the dynamics with lower initialization cost. Moreover, it turns out that the robust optimization problem underlying the two-step safe learning problem remains tractable in the setting where the initial uncertainty set is an ellipsoid (in matrix space). We do not anticipate an exact tractable formulation of the T-step safe learning problem for $T \geq 3$. Our analysis of the $T = 2$ case (in addition to the limiting cases of $T = 1$ and $T = \infty$) is mainly motivated by the aforementioned tractability reason (see Theorem 9).



(a) S_k^1 grows with k.



(b) U_k shrinks with k.

Figure 2: One-step safe learning associated with the numerical example in Section 3.6.

We take the input to the two-step safe learning problem to be a polyhedral safety region $S \subset \mathbb{R}^n$ given in the form of (5), an objective function representing initialization cost which for simplicity we again take to be a linear function $c^T x$, and an uncertainty set $U_0 \subset \mathbb{R}^{n\times n}$ to which the matrix $A_\star$ belongs. We assume the set U_0 is an ellipsoid; this means that there is a strictly convex quadratic function $q : \mathbb{R}^{n\times n} \to \mathbb{R}$ such that

$$U_0 = \left\{A \in \mathbb{R}^{n\times n} \mid q(A) \leq 0\right\}.$$

An example of such an uncertainty set is $U_0 = \{A \in \mathbb{R}^{n\times n} \mid \|A - A_0\|_F \leq \gamma\}$, where A_0 is a nominal matrix, γ is a positive scalar, and $\|\cdot\|_F$ denotes to the Frobenius norm. Having safely collected k length-two trajectories $\{(x_j, A_\star x_j, A_\star^2 x_j)\}_{j=1}^k$, our uncertainty around $A_\star$ reduces to:

$$U_k = \{A \in U_0 \mid Ax_j = A_\star x_j, A^2 x_j = A_\star^2 x_j, j = 1, \dots, k\}. \tag{12}$$

The optimization problem we would like to solve to find the next best two-step safe initialization point is the following:

$$\begin{aligned} \min_x \quad & c^T x \\ \text{s.t.} \quad & x \in S \\ & Ax \in S \quad \forall A \in U_k \\ & A^2 x \in S \quad \forall A \in U_k. \end{aligned} \tag{13}$$

The feasible region of (13) is the set of two-step safe points with information available at step k and is denoted, using our convention, by S_k^2.

4.1. Reformulation via the S-Lemma

In this subsection, we derive a tractable reformulation of problem (13), which as a consequence results in an efficient semidefinite representation of the set S_k^2. Recall that n denotes the dimension of the state and r denotes the number of facets of the polytopic safety set S.

Theorem 9 *Problem (13) can be reformulated as a semidefinite program involving $3r$ scalar inequalities and $2r$ positive semidefinite constraints on matrices of size at most $(n^2+1)\times(n^2+1)$.*

Our proof makes use to the S-lemma (see, e.g., Pólik and Terlaky, 2007) which we recall next.

Lemma 10 (S-lemma) *For two quadratics functions q_a and q_b, if there exists a point $\bar{x}$ such that $q_a(\bar{x})<0$, then the implication*

$$\forall x, [q_a(x)\leq 0 \Rightarrow q_b(x)\leq 0]$$

holds if and only if there exists a scalar $\lambda\geq 0$ such that

$$\lambda q_a(x)-q_b(x)\geq 0 \quad \forall x.$$

Proof [of Theorem 9] Note that the set of equations

$$Ax_j = A_\star x_j, \quad A^2 x_j = A_\star^2 x_j \quad j=1,\dots,k$$

in the definition of U_k in (12) is equivalent to the set of linear equations

$$Ax_j = A_\star x_j, \quad A(A_\star x_j) = A_\star^2 x_j \quad j=1,\dots,k. \tag{14}$$

If there is only one matrix in U_0 that satisfies all of the equality constraints in (14) (a condition that can be checked via a simple modification of Lemma 5), then we have found $A_\star$ and (13) becomes a linear program. Therefore, let us assume that more than one matrix in U_0 satisfies the constraints in (14). In order to apply the S-lemma, we need to remove these equality constraints, a task that we accomplish via variable elimination. Let $\hat{n}$ be the dimension of the affine subspace of matrices that satisfy the constraints in (14) and let $\hat{A}\in\mathbb{R}^{n\times n}$ be an arbitrary member of this affine subspace. Let $A_1,\dots,A_{\hat{n}}\in\mathbb{R}^{n\times n}$ be a basis of the subspace

$$\{A\in\mathbb{R}^{n\times n} \mid Ax_j=0, \quad A(A_\star x_j)=0 \quad j=1,\dots,k\}.$$

Consider an affine function $g:\mathbb{R}^{\hat{n}}\to\mathbb{R}^{n\times n}$ defined as follows:

$$g(\hat{a}) := \hat{A}+\sum_{i=1}^{\hat{n}}\hat{a}_i A_i.$$

The function g has the properties that it is injective and that for each A that satisfies the equality constraints, there must be a vector $\hat{a}$ such that $A=g(\hat{a})$. In other words, the function g is simply parametrizing the affine subspace of matrices that satisfy the equality constraints. Now we can reformulate (13) as:

$$\begin{aligned} \min_{x} \quad & c^T x \\ \text{s.t.} \quad & x\in S \\ & g(\hat{a})x\in S \quad \forall \hat{a} \quad \text{s.t.} \quad q(g(\hat{a}))\leq 0 \\ & g(\hat{a})^2 x\in S \quad \forall \hat{a} \quad \text{s.t.} \quad q(g(\hat{a}))\leq 0. \end{aligned} \tag{15}$$

Let $\hat{q} := q \circ g$. Since q is a strictly convex quadratic function and g is an injective affine map, $\hat{q}$ is also a strictly convex quadratic function. Since we are under the assumption that there are multiple matrices in U_0 that satisfy the equality constraints, there must be a vector $\bar{a} \in \mathbb{R}^{\hat{n}}$ such that $\hat{q}(\bar{a}) < 0$. To see this, take $\bar{a}_1 \neq \bar{a}_2$ such that $\hat{q}(\bar{a}_1), \hat{q}(\bar{a}_2) \leq 0$. It follows from strict convexity of $\hat{q}$ that $\hat{q}(\frac{1}{2}(\bar{a}_1 + \bar{a}_2)) < 0$. Using the definition of S, problem (15) can be rewritten as:

$$\begin{aligned}
\min_{x} \quad & c^T x \\
\text{s.t.} \quad & h_i^T x \leq b_i \quad i = 1, \ldots, r \\
& \begin{bmatrix} \max_{\hat{a}} & h_i^T g(\hat{a}) x \\ \text{s.t.} & \hat{q}(\hat{a}) \leq 0 \end{bmatrix} \leq b_i \quad i = 1, \ldots, r \\
& \begin{bmatrix} \max_{\hat{a}} & h_i^T g(\hat{a})^2 x \\ \text{s.t.} & \hat{q}(\hat{a}) \leq 0 \end{bmatrix} \leq b_i \quad i = 1, \ldots, r.
\end{aligned} \tag{16}$$

Let $q_{1,i}(\hat{a}; x) = h_i^T g(\hat{a}) x - b_i$ and $q_{2,i}(\hat{a}; x) = h_i^T g(\hat{a})^2 x - b_i$. We consider these functions as quadratic functions of $\hat{a}$ parametrized by x. Note that the coefficients of $q_{1,i}$ and $q_{2,i}$ depend affinely on x. Using logical implications, problem (16) can be rewritten as:

$$\begin{aligned}
\min_{x} \quad & c^T x \\
\text{s.t.} \quad & h_i^T x \leq b_i \quad i = 1, \ldots, r \\
& \forall \hat{a}, [\hat{q}(\hat{a}) \leq 0 \Rightarrow q_{1,i}(\hat{a}; x) \leq 0] \quad i = 1, \ldots, r \\
& \forall \hat{a}, [\hat{q}(\hat{a}) \leq 0 \Rightarrow q_{2,i}(\hat{a}; x) \leq 0] \quad i = 1, \ldots, r.
\end{aligned} \tag{17}$$

Now we use the S-lemma to reformulate an implication between quadratic inequalities as a constraint on the global nonnegativity of a quadratic function. Note that as we have already argued for the existence of a vector $\bar{a}$ such that $\hat{q}(\bar{a}) < 0$, the condition of the S-lemma is satisfied. After introducing variables $\lambda_{1,i}$ and $\lambda_{2,i}$ for $i = 1, \ldots, r$, we apply the S-lemma $2r$ times to reformulate (17) as the following program:

$$\begin{aligned}
\min_{x,\lambda} \quad & c^T x \\
\text{s.t.} \quad & h_i^T x \leq b_i \quad i = 1, \ldots, r \\
& \lambda_{1,i} \hat{q}(\hat{a}) - q_{1,i}(\hat{a}; x) \geq 0 \quad \forall \hat{a} \quad i = 1, \ldots, r \\
& \lambda_{2,i} \hat{q}(\hat{a}) - q_{2,i}(\hat{a}; x) \geq 0 \quad \forall \hat{a} \quad i = 1, \ldots, r \\
& \lambda_{1,i} \geq 0, \quad \lambda_{2,i} \geq 0 \quad i = 1, \ldots, r.
\end{aligned} \tag{18}$$

It is a standard procedure to convert the constraint that a quadratic function of N variables is globally nonnegative into a semidefinite constraint on a matrix of size $(N+1) \times (N+1)$. Note that the coefficients of $q_{1,i}$ and $q_{2,i}$ depend affinely on x; this results in linear matrix inequalities when (18) is converted into a semidefinite program. ■

4.2. Number of Two-Step Trajectories Needed for Learning

In Section 3, we established that when one-step safe learning is possible, it can be done with at most n trajectories of length one (see Corollary 8). It is natural to ask how many trajectories might be required in the case of two-step safe learning. We show that generically, $\lceil \frac{n}{2} \rceil$ two-step trajectories suffice for learning.

Given m two-step trajectories, let $X = [x_1, \ldots, x_m]$ be a matrix whose columns are the vectors where our trajectories are initialized. Since our trajectories are of length two, we will observe $Y^{(1)} := A_\star X$ and $Y^{(2)} := A_\star^2 X$. We can write these measurements as the following linear system in A:

$$A[X, Y^{(1)}] = [Y^{(1)}, Y^{(2)}]. \tag{19}$$

From this, it is clear that A will be uniquely identifiable if the matrix $[X, Y^{(1)}]$ has rank n. This is only possible if $[X, Y^{(1)}]$ has at least n columns; in particular, this requires that $m \geq \lceil \frac{n}{2} \rceil$. This suggests that we may be able to learn A with only $\lceil \frac{n}{2} \rceil$ trajectories if we choose X correctly. Unfortunately, it is possible that no matter how we choose X, we may need more than $\lceil \frac{n}{2} \rceil$ trajectories in order to make $[X, Y^{(1)}]$ have rank n. This can be seen for example if A is the zero matrix or the identity matrix. Despite this, we can design an algorithm (Algorithm 3) for which $\lceil \frac{n}{2} \rceil$ trajectories suffice generically to make the matrix $[X, Y^{(1)}]$ have rank n, and hence for learning $A_\star$. Our algorithm relies on the following lemma as a subroutine (see the appendix for a proof).

Lemma 11 *If S_0^2 is full-dimensional, one can solve $2n$ semidefinite programs to find $2n$ vectors in S_0^2 whose convex hull is full-dimensional (if S_0^2 is not full-dimensional, the same process will prove that it is not full-dimensional). These semidefinite programs have the same variables and constraints as the program from Theorem 9, and in addition, at most n linear constraints.*

Our algorithm for two-step safe learning is Algorithm 3. We can prove the following theorem about it.

Theorem 12 *Suppose S_0^2 is full-dimensional. For any matrix $A_\star$ outside of a Lebesgue measure zero set in $\mathbb{R}^{n\times n}$, Algorithm 3 almost surely succeeds in safe learning using only $\lceil \frac{n}{2} \rceil$ trajectories.*

The proof of Theorem 12 relies on the following proposition whose proof can be found in the appendix.

Proposition 13 *Let λ^n denote the Lebesgue measure on $\mathbb{R}^n$. Let $\{\delta_t\}_{t=1}^m$ be a finite sequence of mutually independent random variables in $\mathbb{R}^n$. Suppose that for each t, the law of δ_t is absolutely continuous with respect to λ^n. Let $\{f_t\}_{t=1}^{m-1}$ be any sequence of functions mapping $\mathbb{R}^{n\times t}$ to $\mathbb{R}^n$. Define $z_1 = \delta_1$ and $z_t = f_{t-1}(z_1, ..., z_{t-1}) + \delta_t$ for $t = 2, \ldots, m$. For every λ^{nm} null-set N, we have $\mathbb{P}((z_1, ..., z_m) \in N) = 0$.*

Proof [of Theorem 12] First observe that by the convexity of S_k^2 (i.e., the feasible set of (13)), every initialization point chosen by Algorithm 3 is two-step safe.

Assume for simplicity that n is even and let $m = \lceil \frac{n}{2} \rceil$. Consider the set

$$\mathcal{V} := \{[A, X] \in \mathbb{R}^{n\times(n+m)} \mid \det(Z(A, X)) = 0\},$$

where $Z(A, X) \in \mathbb{R}^{n\times n}$ is the first n columns of $[X, AX]$. As a zero-set of a polynomial, $\mathcal{V}$ is either all of $\mathbb{R}^{n\times(n+m)}$ or has Lebesgue measure zero. It is not all of $\mathbb{R}^{n\times(n+m)}$ since, defining I_s to be the $s \times s$ identity matrix, we can take

$$X = \left[\begin{array}{c} I_m \\ \hline 0 \end{array}\right], \quad A = \left[\begin{array}{c|c} 0 & 0 \\ \hline I_{\lfloor \frac{n}{2} \rfloor} & 0 \end{array}\right]$$

Algorithm 3: Two-Step Safe Learning Algorithm

Require: S_0^2 full-dimensional

Input : polyhedron $S \subset \mathbb{R}^n$, ellipsoid $U_0 \subset \mathbb{R}^{n\times n}$, cost vector $c \in \mathbb{R}^n$, and a constant $\varepsilon \in (0, 1]$.

Output : A matrix $A_\star \in \mathbb{R}^{n\times n}$.

1 Compute $2n$ vectors $z_1, \dots, z_{2n} \in S_0^2$ such that $\operatorname{conv}\{z_1, \dots, z_{2n}\}$ is full-dimensional (cf. Lemma 11)
2 Define $m = \lceil \frac{n}{2} \rceil$
3 **for** $k = 0, \dots, m-1$ **do**
4     $U_k \leftarrow \{A \in U_0 \mid Ax_j = y_j^{(1)}, Ay_j^{(1)} = y_j^{(2)} \quad j = 1, \dots, k\}$
5   **if** *U_k is a singleton*[2] **then**
6   **return** the single element in U_k as $A_\star$
7   **end**
8   Let $x_k^\star$ be an optimal solution to problem (13) with the set U_k (cf. Theorem 9)
9   Pick a random vector $\lambda \in \mathbb{R}^{2n}$ from the $2n$-dimensional simplex[3]
10     $x_{k+1} \leftarrow (1-\varepsilon)x_k^\star + \varepsilon \sum_{i=1}^{2n} \lambda_i z_i$
11   Observe $y_{k+1}^{(1)} \leftarrow A_\star x_{k+1}$, $y_{k+1}^{(2)} \leftarrow A_\star y_{k+1}^{(1)}$
12 **end**
13 Define matrix $X = [x_1, \dots, x_m]$
14 Define matrix $Y^{(1)} = [y_1^{(1)}, \dots, y_m^{(1)}]$
15 Define matrix $Y^{(2)} = [y_1^{(2)}, \dots, y_m^{(2)}]$
16 **return** $A_\star = [Y^{(1)}, Y^{(2)}][X, Y^{(1)}]^T([X, Y^{(1)}][X, Y^{(1)}]^T)^{-1}$

and observe that

$$\det(Z(A, X)) = \det(I_n) = 1 \neq 0.$$

Therefore $\mathcal{V}$ must have Lebesgue measure zero. Since the Lebesgue measure on $\mathbb{R}^{n\times(n+m)}$ is the completion of the product measure of the the Lebesgue measures of $\mathbb{R}^{n\times n}$ and $\mathbb{R}^{n\times m}$, we have that for almost every A, the set

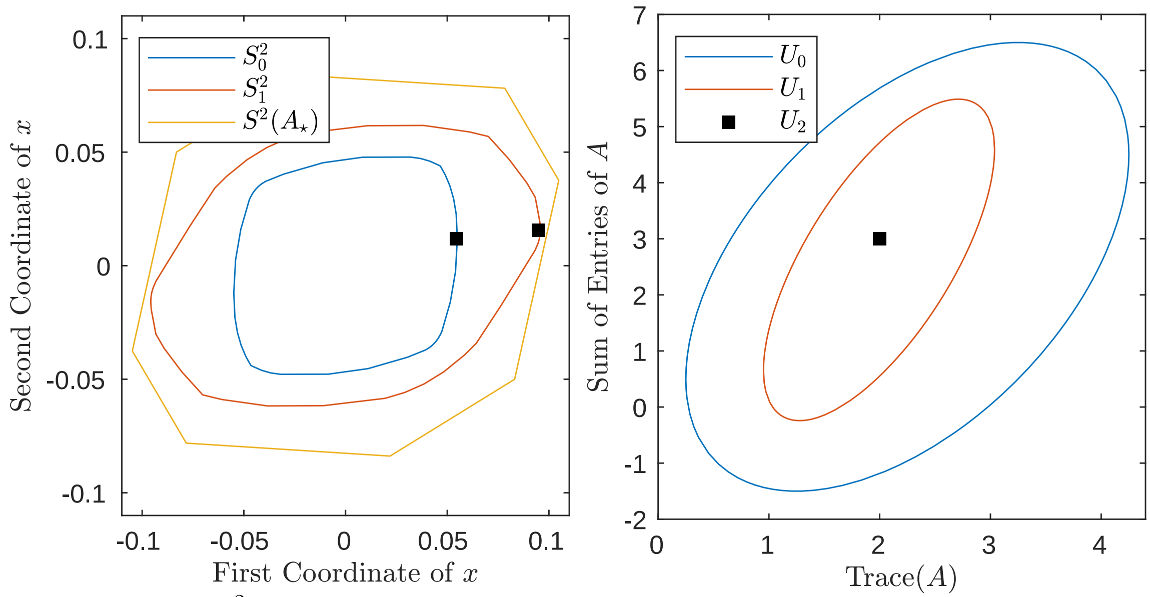
$$\mathcal{V}_A := \{X \in \mathbb{R}^{n\times m} \mid Z(A, X) = 0\}$$

has Lebesgue measure zero. Thus there must exist a set $\mathcal{A} \subset \mathbb{R}^{n\times n}$ of Lebesgue measure zero such that if $A \notin \mathcal{A}$, then $\mathcal{V}_A$ has Lebesgue measure zero.

Supposing $A_\star \notin \mathcal{A}$, we now apply Proposition 13 to the points $x_1, \dots, x_m$ produced by Algorithm 3. Assume that the algorithm does not return at Line 6, otherwise there is nothing to prove. Notice that for some choice of functions $f_1, \dots, f_{m-1} : \mathbb{R}^{n\times t} \to \mathbb{R}^n$, we can write for $k = 2, \dots, m$, $x_k = f_{k-1}(x_0^\star, \dots, x_{k-1}^\star) + \delta_k$ with $\delta_k = \varepsilon \sum_{i=1}^{2n} \lambda_i z_i$. It is clear that the law of δ_k is absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^n$ since $\operatorname{conv}(\{z_1, \dots, z_{2n}\})$ is full-dimensional. Hence, letting $X \in \mathbb{R}^{n\times m}$ be the matrix with $x_1, ..., x_m$ as columns, by Proposition 13, $\mathbb{P}(X \in \mathcal{V}_{A_\star}) = 0$. Therefore, almost surely, we have $\det(Z(A_\star, X)) \neq 0$. This proves that

2. The same approach as the proof of Lemma 5 can be used to perform this check.

3. Any distribution on the simplex that is absolutely continuous with respect to the Lebesgue measure would work, for example, the uniform distribution.



(a) S_k^2 grows with k.

(b) U_k shrinks with k.

Figure 3: Two-step safe learning associated with the numerical example in Section 4.3.

$[X, Y^{(1)}]$ almost surely has rank n and hence

$$[Y^{(1)}, Y^{(2)}][X, Y^{(1)}]^T([X, Y^{(1)}][X, Y^{(1)}]^T)^{-1} = A_\star.$$

■

4.3. Numerical Example

We present a numerical example of two-step safe learning, again with $n = 4$. Here we choose a nominal matrix

$$A_0 = \begin{bmatrix} 2.25 & 0.75 & 4.25 & 1.75 \\ 2.25 & -3.25 & -1.25 & -2.25 \\ -2.00 & -2.75 & 1.25 & 0.00 \\ 1.75 & -0.25 & -2.00 & 2.00 \end{bmatrix}$$

and let $U_0 = \{A \in \mathbb{R}^{4\times 4} \mid \|A - A_0\|_F \leq 1\}$. We let $S = \{x \in \mathbb{R}^4 \mid |x_i| \leq 1, i = 1, \ldots, 4\}$ and $c = (-1, 0, 0, 0)^T$. We choose the true matrix $A_\star$ to be the same matrix used in Section 3.5 (which belongs to U_0).

In this example, with Algorithm 3, we learn the true matrix $A_\star$ by choosing two initialization points that are each two-step safe. In other words, Algorithm 3 chooses $x_1 \in \mathbb{R}^4$, observes $A_\star x_1$ and $A_\star^2 x_1$, then chooses $x_2 \in \mathbb{R}^4$, and observes $A_\star x_2$ and $A_\star^2 x_2$. We can verify that we have recovered $A_\star$ if the vectors $\{x_1, A_\star x_1, x_2, A_\star x_2\}$ are linearly independent, which is the case. The projection to the first two dimensions of the two initialization points x_1 and x_2 are plotted in Figure 3(a).

Because of the cost vector c, points further to the right in the plot have lower initialization cost. Also plotted are the projections to the first two dimensions of the sets

$$\begin{aligned} S_0^2 &= \{x \in S \mid Ax \in S, A^2x \in S \quad \forall A \in U_0\}, \\ S_1^2 &= \{x \in S \mid Ax \in S, A^2x \in S \quad \forall A \in U_1\}, \\ S^2(A_\star) &= \{x \in S \mid A_\star x \in S, A_\star^2 x \in S\}. \end{aligned}$$

The sets S_0^2 and S_0^1 are the projections to x-space of the feasible regions of our two semidefinite programs (cf. Theorem 9). The set $S^2(A_\star)$ is the true two-step safety region of $A_\star$. In Figure 3(b), we plot U_k (the remaining uncertainty after observing k trajectories of length two) for $k \in \{0, 1, 2\}$; we draw a two-dimensional projection of these sets of matrices by looking at the trace and the sum of the entries of each matrix in the set. Note that U_2 is a single point since we have recovered the true dynamics after observing the second trajectory. The cost of learning (i.e., $c^T x_1 + c^T x_2$) is -0.1493.

We can construct an analogue of the offline Algorithm 2 by only making measurements from S_0^2. This approach would first pick the optimal point in S_0^2 (i.e., x_1), and then another vector in S_0^2 close to x_1, but linearly independent from it. The cost of learning for this offline approach would be $2c^T x_1 = -0.1099$. Finally, we can again find a lower bound on the cost of learning of any algorithm that chooses two two-step safe initialization points by assuming we know $A_\star$ ahead of time and optimizing $c^T x$ over $S^2(A_\star)$; in this example, the lower bound is -0.2097. Here, again, we see that by using information on the fly, we can succeed at safe learning at a considerably lower cost than the offline approach.

5. Infinite-Step Safe Learning of Linear Systems

In contrast to the previous two sections, in this section we consider the problem of safely learning the linear dynamical system in (3) from trajectories of unbounded length. This means that we are constrained to initializing the system at points whose entire future trajectories are guaranteed to remain in a specified safety region.

More formally, in the infinite-step safe learning problem, we have as input a polyhedral safety region $S \subset \mathbb{R}^n$ given in the form (5), an objective function representing initialization cost which for simplicity we take to be a linear function $c^T x$, and a polyhedral uncertainty set $U_0 \subset \mathbb{R}^{n \times n}$, given in the form (4), to which the matrix $A_\star$ governing the true dynamics belongs. Having collected k safe trajectories $\{(x_\ell, A_\star x_\ell, A_\star^2 x_\ell, \ldots)\}_{\ell=1}^k$, our uncertainty around $A_\star$ reduces to

$$U_k = \{A \in U_0 \mid Ax_\ell = A_\star x_\ell, A^2 x_\ell = A_\star^2 x_\ell, \ldots, A^n x_\ell = A_\star^n x_\ell, \ell = 1, \ldots, k\}.$$

Note that in the definition of U_k, information contained in the tail of the trajectories, beyond step n, is discarded. This is because of the following proposition, which we prove in the appendix using the Cayley-Hamilton theorem.

Proposition 14 *Let $A_\star \in \mathbb{R}^n$. For any vector $x \in \mathbb{R}^n$ and integer $k \geq n$, we have*

$$\begin{aligned} &\{A \in \mathbb{R}^{n \times n} \mid Ax = A_\star x, A^2 x = A_\star^2 x, \ldots, A^k x = A_\star^k x\} \\ =&\{A \in \mathbb{R}^{n \times n} \mid Ax = A_\star x, A^2 x = A_\star^2 x, \ldots, A^n x = A_\star^n x\}. \end{aligned}$$

Given the sets S, U_k, and the vector c, the optimization problem we would like to solve to find the next best infinite-step safe initialization point is the following:

$$\begin{aligned} \min_x \quad & c^T x \\ \text{s.t.} \quad & x \in S \\ & A^t x \in S \quad \forall A \in U_k, \quad t = 1, 2, \ldots . \end{aligned} \tag{20}$$

In keeping with the naming conventions of this work, we refer to the feasible region of (20) as S_k^∞; if U_0 is a single matrix A, we call this set $S^\infty(A)$.

Unlike problems (7) and (13), which as we showed admit a reformulation as tractable conic programs, problem (20) is in general intractable. In fact, even when the set U_0 is a singleton, deciding if a given vector x is feasible to (20) is NP-hard (Ahmadi and Günlük, 2018, Theorem 2.1). Therefore, our aim in this section is to find tractable inner approximations to the feasible region of (20).

We now describe our assumptions on (20) and their implications. We assume that our safety region S is compact, as this is typically the case in most applications. It is also natural to require S_0^∞ to be full-dimensional, as otherwise the implementation of a safe initialization would be impossible in presence of arbitrarily small quantization error and/or physical perturbations. Under these assumptions, we can make the following conclusions.

Proposition 15 *Suppose that S is compact and S_0^∞ is full-dimensional. Then, U_0 is bounded*[4] *and contains only matrices with spectral radius*[5] *less than or equal to one.*

Proof Suppose for the sake of contradiction that U_0 is unbounded. Because U_0 is a convex set, there must exist a matrix $A_0 \in U_0$ and a nonzero matrix $D \in \mathbb{R}^{n \times n}$ such that $A_0 + \lambda D \in U_0$ for all $\lambda \geq 0$. Since $D \neq 0$, the nullspace of D is not full-dimensional and therefore S_0^∞ cannot be contained within it. Therefore there exists a vector $x \in S_0^\infty$ such that $Dx \neq 0$. Now observe that $(A_0 + \lambda D)x = A_0 x + \lambda D x \in S$ for every $\lambda \geq 0$, which contradicts the compactness of S.

To prove that U_0 only contains matrices with spectral radius at most 1, suppose for the sake of contradiction that there exists a matrix $\bar{A} \in U_0$ with an eigenvalue $\lambda \in \mathbb{C}$ satisfying $|\lambda| > 1$. Because the spectral radius is dominated by the operator norm, we have that for every nonnegative integer k, $\|\bar{A}^k\| \geq \rho(\bar{A}^k) \geq |\lambda|^k$. Let R be a nonnegative scalar large enough such that $x \in S$ implies $\|x\| \leq R$. Let $g \in \mathbb{R}^n$ be a random vector with each entry an independent standard normal. First, we claim that $\mathbb{P}(g \in S_0^\infty) = 0$. Let $U \Sigma_k V^T$ be the singular value decomposition of $\bar{A}^k$, with the largest singular value placed on the first entry of Σ_k. By the rotational invariance of Gaussian random vectors, $\|\bar{A}^k g\|$ has the same distribution as $\|\Sigma_k g\|$. Therefore, letting g_1 denote the first

4. As the proof demonstrates, the claim that U_0 is bounded holds even under the weaker requirement that S is compact and S_0^1 is full-dimensional.

5. Recall that the spectral radius of a square matrix A is defined as $\rho(A) := \max_i |\lambda_i(A)|$, where $\lambda_i(A)$ is the i-th eigenvalue of A.

entry of the random vector g, we have

$$
\begin{aligned}
\mathbb{P}(g \in S_0^\infty) &\leq \mathbb{P}(\|\bar{A}^k g\| \leq R \quad \forall k \geq 0) \\
&\leq \inf_{k \geq 0} \mathbb{P}(\|\bar{A}^k g\| \leq R) \\
&= \inf_{k \geq 0} \mathbb{P}(\|\Sigma_k g\| \leq R) \\
&\leq \inf_{k \geq 0} \mathbb{P}(\|\Sigma_k\| |g_1| \leq R) \\
&\leq \inf_{k \geq 0} \mathbb{P}(|\lambda|^k |g_1| \leq R) \\
&= \inf_{k \geq 0} \mathbb{P}(|g_1| \leq R|\lambda|^{-k}) \\
&= \inf_{k \geq 0} \frac{1}{\sqrt{2\pi}} \int_{-R|\lambda|^{-k}}^{R|\lambda|^{-k}} \exp(-s^2/2)\, ds \\
&\leq \inf_{k \geq 0} \sqrt{\frac{2}{\pi}} R|\lambda|^{-k} = 0.
\end{aligned}
$$

Since S_0^∞ is full-dimensional, its Lebesgue measure is positive. Furthermore, we showed that the Gaussian measure of S_0^∞ is zero. Since the Lebesgue measure is absolutely continuous with respect to the Gaussian measure, this is a contradiction. ■

In view of Proposition 15, if we want S_0^∞ to be full-dimensional, we must assume that each matrix in U_0 has spectral radius less than or equal to one. We make the slightly stronger assumption that U_0 only contains matrices with spectral radius less than one. (Recall that a matrix with spectral radius less than one is called *stable* or *Schur stable*.) Under this assumption, for the set S_0^∞ to be nonempty, we need the origin to be in our safety region S (as otherwise, all initial conditions would converge to the origin under (3) and eventually leave S). We work with the slightly stronger assumption that the origin belongs to the interior of S. Under this assumption, our representation of the polytope S in (5) can be simplified (after potential rescaling) to:

$$
S = \left\{ x \in \mathbb{R}^n \mid h_i^T x \leq 1 \quad i = 1, \ldots, r \right\}. \tag{21}
$$

Before we state the main theorem of this section, we need to recall some basic definitions. Let $\mathbb{S}^{m \times m}$ denote the space of $m \times m$ real-valued symmetric matrices. We say that a matrix-valued function $M : \mathbb{R}^n \to \mathbb{S}^{m \times m}$ is a *polynomial matrix* if each entry M_{ij} is a polynomial.

Definition 16 (SOS Polynomial and SOS Matrix) *A polynomial $p : \mathbb{R}^n \to \mathbb{R}$ is said to be a* sum of squares *(SOS) if there exist some polynomials $q_1, \ldots, q_r \colon \mathbb{R}^n \mapsto \mathbb{R}$ such that $p = \sum_{i=1}^r q_i^2$. A polynomial matrix $M : \mathbb{R}^n \to \mathbb{S}^{m \times m}$ is said to be a* sum of squares matrix *(SOS matrix) if the scalar-valued polynomial $y^T M(x) y$ in the $n + m$ variables (x, y) is SOS.*

We can now present our main theorem of this section, which enables us to find infinite-step safe initialization points. Our arguments thus far justify the assumptions that this theorem places on the uncertainty set U_k.

Theorem 17 *Let the polyhedron $S \subseteq \mathbb{R}^n$ be as in (21), and the polyhedron $U_0 \subseteq \mathbb{R}^{n \times n}$ be as in (4). For $t = 0, \ldots, n$ and $\ell = 1, \ldots, k$, let $y_{t,\ell} \in \mathbb{R}^n$ be the tth vector in the ℓth observed* *trajectory; i.e., $y_{0,\ell}$ is the trajectory's initialization and $y_{t,\ell} = A^t y_{0,\ell}$. Let $\{e_p\}_{p=1}^n$ be the canonical basis vectors of $\mathbb{R}^n$. For an even integer d, let $\tilde{S}_{k,d}^\infty$ be the projection to x-space of the feasible region of the following optimization problem:*

$$\min_{x,Q,M_j,M_{t\ell p},\hat{M}_j,\hat{M}_{t\ell p},\sigma_{ij},\sigma_{it\ell p},\varepsilon} \quad c^T x \tag{22}$$

$$\text{s.t.} \quad Q(A), M_j(A) \text{ are } n \times n \text{ SOS matrices with degree} \leq d \quad j = 0, \ldots, s \tag{22a}$$

$$M_{t\ell p}(A) \text{ are } n \times n \text{ symmetric polynomial matrices with degree} \leq d \quad t = 1, \ldots, n \quad \ell = 1, \ldots, k \quad p = 1, \ldots, n \tag{22b}$$

$$\hat{M}_j(A) \text{ are } (n+1) \times (n+1) \text{ SOS matrices with degree} \leq d \quad j = 0, \ldots, s \tag{22c}$$

$$\hat{M}_{t\ell p}(A) \text{ are } (n+1) \times (n+1) \text{ symmetric polynomial matrices with degree} \leq d \quad t = 1, \ldots, n \quad \ell = 1, \ldots, k \quad p = 1, \ldots, n \tag{22d}$$

$$\sigma_{ij}(A) \text{ are SOS polynomials with degree} \leq d \quad i = 1, \ldots, r \quad j = 0, \ldots, s \tag{22e}$$

$$\sigma_{it\ell p}(A) \text{ are polynomials with degree} \leq d \quad i = 1, \ldots, r \quad t = 1, \ldots, n \quad \ell = 1, \ldots, k \quad p = 1, \ldots, n \tag{22f}$$

$$Q(A) - AQ(A)A^T = \varepsilon I + M_0(A) + \sum_{j=1}^{s} M_j(A)(v_j - \mathrm{Tr}(V_j^T A)) + \sum_{t=1}^{n}\sum_{\ell=1}^{k}\sum_{p=1}^{n} M_{t\ell p}(A) e_p^T (A y_{t-1,\ell} - y_{t,\ell}) \quad \forall A \in \mathbb{R}^{n \times n} \tag{22g}$$

$$1 - h_i^T Q(A) h_i = \sigma_{i0}(A) + \sum_{j=1}^{s} \sigma_{ij}(A)(v_j - \mathrm{Tr}(V_j^T A)) + \sum_{t=1}^{n}\sum_{\ell=1}^{k}\sum_{p=1}^{n} \sigma_{it\ell p}(A) e_p^T (A y_{t-1,\ell} - y_{t,\ell}) \quad i = 1, \ldots, r \quad \forall A \in \mathbb{R}^{n \times n} \tag{22h}$$

$$\begin{bmatrix} Q(A) & x \\ x^T & 1 \end{bmatrix} = \hat{M}_0(A) + \sum_{j=1}^{s} \hat{M}_j(A)(v_j - \mathrm{Tr}(V_j^T A)) + \sum_{t=1}^{n}\sum_{\ell=1}^{k}\sum_{p=1}^{n} \hat{M}_{t\ell p}(A) e_p^T (A y_{t-1,\ell} - y_{t,\ell}) \quad \forall A \in \mathbb{R}^{n \times n} \tag{22i}$$

$$\varepsilon > 0. \tag{22j}$$

Then,

(i) The program (22) can be reformulated as a semidefinite program of size polynomial in the size of the input (S,U_0, $\{y_{t,\ell}\}$ and c).

(ii) We have $\tilde{S}_{k,d}^\infty \subseteq S_k^\infty$ (i.e, any vector x feasible to this semidefinite program is infinite-step safe).

(iii) Furthermore, if U_k is compact and contains only stable matrices, then, for large enough d, the set $\tilde{S}_{k,d}^\infty$ is full-dimensional.

In words, Theorem 17 allows us to optimize initialization cost over semidefinite representable subsets of the set of infinite-step safe points. While the theorem guarantees full-dimensionality of these subsets for large d, in our experience, small values of d suffice for safe learning; see Section 5.5. We present the proof of this theorem in Section 5.3 after we review some results building up to it in Sections 5.1 and 5.2.

5.1. Review of a Result from Ahmadi and Günlük (2018)

The basis of (22) comes from the approach of Ahmadi and Günlük (2018). Let the safety set S be as in (21). For a single stable matrix A, this approach can be used to compute tractable inner approximations of $S^\infty(A)$.

Recall that a matrix $P \in \mathbb{S}^{n\times n}$ is positive definite (resp. positive semidefinite) if for every nonzero vector $x \in \mathbb{R}^n$ we have that $x^T P x > 0$ (resp. $x^T P x \geq 0$); we indicate such a matrix with the notation $P \succ 0$ (resp. $P \succeq 0$). Furthermore, we use the notation $P \succ Q$ (resp. $P \succeq Q$) if we have that $P - Q$ is positive definite (resp. positive semidefinite). Consider the following semidefinite program:

$$\begin{aligned} \min_{x\in\mathbb{R}^n, Q\in\mathbb{S}^{n\times n}} \quad & c^T x \\ \text{s.t.} \quad & Q \succ 0 \\ & Q \succeq AQA^T \\ & h_i^T Q h_i \leq 1 \quad i = 1, \dots, r \\ & \begin{bmatrix} Q & x \\ x^T & 1 \end{bmatrix} \succeq 0. \end{aligned} \tag{23}$$

The following lemma is a special case of Theorem 2.11 from Ahmadi and Günlük (2018). The proof carries some intuition behind the construction of (22) and therefore we include it here.

Lemma 18 *Let $S \subset \mathbb{R}^n$ be as in (21) and $A \in \mathbb{R}^{n\times n}$. Let $\tilde{S}^\infty(A)$ be the projection to x-space of the feasible region of (23). We have $\tilde{S}^\infty(A) \subseteq S^\infty(A)$.*

Proof Let $E := \{x \mid x^T Q^{-1} x \leq 1\}$; first we show that the constraints $h_i^T Q h_i \leq 1$ for $i = 1, \dots, r$ imply the set inclusion $E \subseteq S$. For a set $T \subseteq \mathbb{R}^n$, we define its *polar* T° as $T^\circ := \{y \mid y^T x \leq 1, \forall x \in T\}$. One can check that $E \subseteq S$ if and only if $S^\circ \subseteq E^\circ$, $S^\circ = \mathrm{conv}(\{h_i\}_{i=1}^r)$, and $E^\circ = \{x \mid x^T Q x \leq 1\}$. Thus, for each i, the constraint $h_i^T Q h_i \leq 1$ implies $h_i \in E^\circ$. By convexity, it follows that $S^\circ \subseteq E^\circ$ and therefore $E \subseteq S$ as desired.

Note that by the Schur complement lemma, the constraint $\begin{bmatrix} Q & x \\ x^T & 1 \end{bmatrix} \succeq 0$ implies that $x \in E$. Thus, x is in the safety region. To show that the trajectory remains safe for all time it suffices to show that the set E is invariant under the dynamics, i.e. that if $\bar{x}$ is in E, then so is $A\bar{x}$. Fix an arbitrary point $\bar{x} \in E$. By two applications of the Schur complement lemma, the constraint $Q \succeq AQA^T$ is equivalent to $Q^{-1} \succeq A^T Q^{-1} A$. This linear matrix inequality implies that $\bar{x}^T Q^{-1} \bar{x} \geq \bar{x}^T A^T Q^{-1} A \bar{x}$. Thus, we have $(A\bar{x})^T Q^{-1} (A\bar{x}) \leq \bar{x}^T Q^{-1} \bar{x} \leq 1$, and hence $A\bar{x} \in E$ as desired. ■

The approach of Ahmadi and Günlük (2018) and its extensions lead to infinite-safe sets for dynamics governed by a single matrix, or a group of matrices where the “joint spectral radius” is less than one. Our Theorem 17 extends their approach to the case where each individual matrix in U_k is stable, which is a weaker condition than the joint spectral radius of the matrices in U_k being less than one. This is the relevant setting for us which is not covered by Ahmadi and Günlük (2018).

We also note that the approach of Ahmadi and Günlük (2018) gives a hierarchy of inner approximations to $S^\infty(A)$. However, the first level of the hierarchy is sufficient for our goal of finding full-dimensional inner approximations.

5.2. Review of Putinar's Positivstellensatz

In this subsection, we briefly review Putinar's Positivstellensatz and its matrix generalization due to Scherer and Hol which, when combined with Lemma 18, help us approximate the feasible region of (20) with semidefinite programs. These theorems involve SOS polynomials and matrices (cf. Definition 16), and our interest in them stems from the following well-known fact: the constraint that an unknown polynomial or a polynomial matrix of a given degree be SOS and satisfy a set of affine inequalities can be cast as an semidefinite program of tractable size; see, e.g., Parrilo (2000).

Definition 19 (Archimedian Property) *We say that a set of n-variate polynomials $\mathcal{G} = \{g_1, \ldots, g_m\}$ satisfies the* Archimedian property *if there exists a scalar R and SOS polynomials $\sigma_0, \sigma_1, \ldots, \sigma_m$ such that*

$$R^2 - \sum_{i=1}^{n} x_i^2 = \sigma_0(x) + \sum_{j=1}^{m} \sigma_j(x) g_j(x) \quad \forall x \in \mathbb{R}^n.$$

Note that the Archimedian property implies that the set

$$K(\mathcal{G}) := \{x \in \mathbb{R}^n \mid g_i(x) \geq 0 \quad i = 1, \ldots, m\} \tag{24}$$

is compact. Furthermore, it is known that if $g_1, \ldots, g_m$ are affine polynomials and if $K(\mathcal{G})$ is compact, then $\mathcal{G}$ satisfies the Archimedian property (see, e.g., Laurent, 2009). Note that if we let $\mathcal{G} = \{A \mapsto v_j - \mathrm{Tr}(V_j^T A)\}_{j=1}^s$, then U_0 from (4) equals $K(\mathcal{G})$ and $\mathcal{G}$ satisfies the Archimedian property.

Theorem 20 (Putinar's Positivstellensatz (Putinar, 1993)) *Let $\mathcal{G} = \{g_1, \ldots, g_m\}$ be a set of n-variate polynomials satisfying the Archimedian property and let $K(\mathcal{G})$ be as in (24). For any polynomial $p : \mathbb{R}^n \to \mathbb{R}$, we have $p(x) > 0$ for all $x \in K(\mathcal{G})$ if and only if there exists a positive scalar ε and SOS polynomials $\sigma_0, \sigma_1, \ldots, \sigma_m$ such that*

$$p(x) = \varepsilon + \sigma_0(x) + \sum_{j=1}^{m} \sigma_j(x) g_j(x) \quad \forall x \in \mathbb{R}^n.$$

Theorem 21 (Matrix Putinar's Positivstellensatz (Scherer and Hol, 2006)) *Let $\mathcal{G} = \{g_1, \ldots, g_m\}$ be a set of n-variate polynomials satisfying the Archimedian property and let $K(\mathcal{G})$ be as in (24). For any polynomial matrix $M : \mathbb{R}^n \to \mathbb{S}^{r \times r}$, we have $M(x) \succ 0$ for all $x \in K(\mathcal{G})$ if and only if there exists a positive scalar ε and SOS matrices $S_0, S_1, \ldots, S_m$ such that*

$$M(x) = \varepsilon I + S_0(x) + \sum_{j=1}^{m} S_j(x) g_j(x) \quad \forall x \in \mathbb{R}^n.$$

5.3. Proof of Theorem 17

In addition to Theorems 20 and 21, the proof of claim (iii) in Theorem 17 also relies on the following technical lemma.

Lemma 22 *Let $U \subset \mathbb{R}^{n\times n}$ be a compact set of matrices. Then, every matrix $A \in U$ is stable if and only if there exists a $n \times n$ SOS matrix $P : \mathbb{R}^{n\times n} \mapsto \mathbb{S}^{n\times n}$ such that*

1. $P(A) \succ 0 \quad \forall A \in U$,
2. $P(A) - A^T P(A)A \succ 0 \quad \forall A \in U$.

Proof ["If"] It is straightforward to check that the conditions imply that for any matrix $A \in U$, the positive definite Lyapunov function $V_A(x) = x^T P(A)x$ satisfies $V_A(Ax) < V(x)$ for all $x \neq 0$. The stability of A then follows from Lyapunov's stability theorem; see, e.g., (Żak, 2003, Theorem 4.3).

["Only if"] For a positive integer N, let the SOS matrix $P_N : \mathbb{R}^{n\times n} \to \mathbb{S}^{n\times n}$ be defined as follows:

$$P_N(A) = \sum_{k=0}^{N} (A^k)^T (A^k).$$

Clearly we have $P_N(A) \succ 0$ for each matrix $A \in U$ since each summand is positive semidefinite and the zeroth summand is the identity matrix. We claim that for sufficiently large N, $P_N(A)$ will satisfy $P_N(A) - A^T P_N(A)A \succ 0$ for all $A \in U$. Observe

$$P_N(A) - A^T P_N(A)A = I - (A^{N+1})^T (A^{N+1}).$$

To show that $P_N(A) - A^T P_N(A)A \succ 0$ for each $A \in U$, we prove

$$\|(A^{N+1})^T (A^{N+1})\| < 1 \quad \forall A \in U.$$

For a matrix B, let $\|B\|_\infty := \max_{ij} |B_{ij}|$. Define the scalars R, M as

$$R := \max_{A\in U} \rho(A), \qquad M := \max_{A\in U} \|A\|_\infty.$$

Since each matrix $A \in U$ is stable and U is compact, $R < 1$. Since U is compact, $M < \infty$. Now fix a matrix $A \in U$ and write $A = QTQ^{-1}$, where $Q \in \mathbb{C}^{n\times n}$ is unitary and $T \in \mathbb{C}^{n\times n}$ is upper triangular (this "Schur decomposition" always exists). Observe that $\|A^N\| = \|T^N\|$. We can bound the norm of powers of a triangular matrix as follows (see Corollary 3.15 of Dowler (2013)):

$$\|T^N\| \leq \sqrt{n} \sum_{j=0}^{n-1} \binom{n-1}{j} \binom{N}{j} \|T\|_\infty^j \rho(T)^{N-j}.$$

In particular, we have

$$\|A^N\| \leq \sqrt{n} \sum_{j=0}^{n-1} \binom{n-1}{j} \binom{N}{j} M^j R^{N-j} \leq \sqrt{n} \sum_{j=0}^{n-1} \binom{n-1}{j} N^j M^j R^{N-j}. \tag{25}$$

Inequality (25) implies that $\lim_{N\to\infty} \|A^N\| = 0$. Therefore, we can choose N large enough such that

$$\|(A^{N+1})^T(A^{N+1})\| \le \|A^{N+1}\|^2 < 1.$$

■

We are now able to present the proof of the main result of this section.

Proof [of Theorem 17]

(i) Recall that for any fixed degree d, the SOS constraints in (22a), (22c), (22e) can be reformulated as semidefinite programming constraints of size polynomial in n; see, e.g., Parrilo (2000). The constraints in (22g), (22h), (22i) can be imposed by coefficient matching via a number of linear equations bounded by a polynomial function of the size of the input $(S, U_0, \{y_{t,\ell}\})$. The constraint that $\varepsilon > 0$ can be rewritten as the constraint $\begin{bmatrix} \varepsilon & 1 \\ 1 & \delta \end{bmatrix} \succeq 0$ for a new variable δ. Therefore, for any fixed degree d, (22) is a semidefinite program of size polynomial in the size of the input $(S, U_0, \{y_{t,\ell}\}, c)$.

(ii) Let $(x, Q, M_j, M_{t\ell p}, \hat{M}_j, \hat{M}_{t\ell p}, \sigma_{ij}, \sigma_{it\ell p}, \varepsilon)$ be feasible to (22). It is straightforward to check that for every $A \in U_k$, the tuple $(x, Q(A))$ satisfies the following constraints

$$\begin{aligned}
&Q(A) \succ 0 \\
&Q(A) \succeq AQ(A)A^T \\
&h_i^T Q(A) h_i \le 1 \quad i = 1, \dots, r \\
&\begin{bmatrix} Q(A) & x \\ x^T & 1 \end{bmatrix} \succeq 0.
\end{aligned} \tag{26}$$

Therefore, by Lemma 18, we have $x \in \tilde{S}^\infty(A) \subseteq S^\infty(A)$ for every $A \in U_k$. Hence,

$$x \in \bigcap_{A \in U_k} S^\infty(A) = S_k^\infty.$$

This implies that $\tilde{S}_{k,d}^\infty \subseteq S_k^\infty$.

(iii) Suppose that U_k is compact and contains only stable matrices. It follows that the set

$$U_k^T := \{A^T \mid A \in U_k\}$$

is also compact and contains only stable matrices. By Lemma 22 applied to U_k^T, there exists an SOS matrix $P(A)$ which satisfies

$$\begin{aligned}
&P(A) \succ 0 \quad \forall A \in U_0^T \\
&P(A) \succ A^T P(A) A \quad \forall A \in U_0^T.
\end{aligned}$$

Now by defining $Q(A) := P(A^T)$, we observe that

$$\begin{aligned}
&Q(A) \succ 0 \quad \forall A \in U_0 \\
&Q(A) \succ AQ(A)A^T \quad \forall A \in U_0.
\end{aligned}$$

Analogously to how we derived linear constraints in (14), we can rewrite the description of U_k as

$$U_k = \{A \in U_0 \mid Ay_{t-1,\ell} = y_{t,\ell}, t = 1, \ldots, n, \ell = 1, \ldots, k\}.$$

Since U_k is a compact polyhedron, the Archimedean property is satisfied for the polynomials $\{A \mapsto v_j - \mathrm{Tr}(V_j^T A)\}$ and $\{A \mapsto \pm e_p^T (Ay_{t-1,\ell} - y_{t,\ell})\}$. By Theorem 21, since $Q(A) - AQ(A)A^T \succ 0$ for every $A \in U_k$, there exists a positive scalar ε and SOS matrices $M_j(A)$ and $M^{\pm}_{t\ell p}(A)$ that satisfy

$$\begin{aligned} Q(A) - AQ(A)A^T =& \varepsilon I + M_0(A) + \sum_{j=1}^{s} M_j(A)(v_j - \mathrm{Tr}(V_j^T A)) \\ &+ \sum_{t=1}^{n}\sum_{\ell=1}^{k}\sum_{p=1}^{n} M^{+}_{t\ell p}(A) e_p^T (Ay_{t-1,\ell} - y_{t,\ell}) \\ &- \sum_{t=1}^{n}\sum_{\ell=1}^{k}\sum_{p=1}^{n} M^{-}_{t\ell p}(A) e_p^T (Ay_{t-1,\ell} - y_{t,\ell}) \quad \forall A \in \mathbb{R}^{n\times n}. \end{aligned}$$

By letting $M_{t\ell p}(A) = M^{+}_{t\ell p}(A) - M^{-}_{t\ell p}(A)$, we can satisfy (22g).

Now observe that for each $i \in \{1, \ldots, r\}$, the function $A \to h_i^T Q(A) h_i$ is continuous. Therefore, since U_k is compact, there exists a positive scalar α satisfying

$$\max_{i\in\{1,\ldots,r\}, A\in U_k} h_i^T Q(A) h_i < \alpha.$$

Observe that the tuple $(\varepsilon/\alpha, Q(A)/\alpha, M_j(A)/\alpha, M_{t\ell p}/\alpha)$ still satisfies (22a), (22g), and (22j). Furthermore, for each $i \in \{1, \ldots, r\}$, $1 - \alpha^{-1} h_i^T Q(A) h_i > 0$ for all $A \in U_k$. Therefore, by Theorem 20, and by a similar argument as in the case of constraint (22g), there exist SOS polynomials $\sigma_{ij}(A)$ and polynomials $\sigma_{it\ell p}(A)$ satisfying (22h).

Since $Q(A)/\alpha$ is a continuous function of A and since $Q(A)/\alpha$ is positive definite for each A in the compact set U_k, there exists a scalar $\beta > 0$ such that $Q(A)/\alpha \succ \beta I$ for all $A \in U_k$. Then, we have

$$\begin{bmatrix} Q(A)/\alpha - \beta I & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \succ 0 \quad \forall A \in U_k.$$

It follows from Theorem 21, and by a similar argument as in the case of constraint (22g), that there exist some SOS matrices $\hat{M}_j(A)$ and symmetric polynomial matrices $\hat{M}_{t\ell p}(A)$ satisfying

$$\begin{aligned} \begin{bmatrix} Q(A)/\alpha - \beta I & 0 \\ 0 & \frac{1}{2} \end{bmatrix} =& \hat{M}_0(A) + \sum_{j=1}^{s} \hat{M}_j(A)(v_j - \mathrm{Tr}(V_j^T A)) \\ &+ \sum_{t=1}^{n}\sum_{\ell=1}^{k}\sum_{p=1}^{n} \hat{M}_{t\ell p}(A) e_p^T (Ay_{t-1,\ell} - y_{t,\ell}) \quad \forall A \in \mathbb{R}^{n\times n}. \end{aligned}$$

Observe that for any $x \in \mathbb{R}^n$ satisfying $\|x\| \le \sqrt{\frac{1}{2\beta}}$, we have $\begin{bmatrix} \beta I & x \\ x^T & \frac{1}{2} \end{bmatrix} \succeq 0$. Therefore, $A \mapsto \hat{M}_0(A) + \begin{bmatrix} \beta I & x \\ x^T & \frac{1}{2} \end{bmatrix}$ is still an SOS matrix of the same degree as $\hat{M}_0(A)$. Hence, for any x satisfying

$\|x\| \leq \sqrt{\frac{1}{2\beta}}$, we have that the tuple

$$\left(x, Q/\alpha, M_j/\alpha, M_{t\ell p}/\alpha, \hat{M}_0 + \begin{bmatrix} \beta I & x \\ x^T & \frac{1}{2} \end{bmatrix}, \hat{M}_1, \ldots, \hat{M}_s, \hat{M}_{t\ell p}, \sigma_{ij}, \sigma_{it\ell p}, \varepsilon/\alpha\right)$$

is feasible to (22) for some degree d large enough.

■

5.4. Number of Trajectories Needed to Learn

In Corollary 8, we established that we need no more than n one-step trajectories to safely learn the matrix $A_\star \in \mathbb{R}^{n\times n}$ governing the true linear dynamical system of interest. Then in Theorem 12, we proved that generically, it is possible to safely learn $A_\star$ using only $\lceil \frac{n}{2} \rceil$ trajectories of length two. We now show that generically, we can safely learn $A_\star$ from a single trajectory of length n.

Theorem 23 *There exists a set $\mathcal{A} \subset \mathbb{R}^{n\times n}$ of Lebesgue measure zero such that if $A_\star \notin \mathcal{A}$, then by observing a single trajectory of length n initialized at random*[6] *from any full-dimensional infinite-step safe set (for example, the set $\tilde{S}^{\infty}_{0,d}$ defined in Theorem 17 for large enough d), we will almost surely safely learn $A_\star$.*

Proof Consider the set

$$\mathcal{V} := \{[A, x] \in \mathbb{R}^{n\times(n+1)} \mid \det([x, Ax, A^2x, \ldots, A^{n-1}x]) = 0\}.$$

This is the zero-set of a polynomial, therefore it is either the entire space or has Lebesgue measure zero. It is not the entire space since we can take A to be the matrix with ones on its first subdiagonal and zeros elsewhere and x to be the vector with one as its first entry and zeros elsewhere. With A and x defined this way, we have

$$\det([x, Ax, A^2x, \ldots, A^{n-1}x]) = \det(I) = 1 \neq 0.$$

Therefore, $\mathcal{V}$ must have Lebesgue measure zero. Since the Lebesgue measure on $\mathbb{R}^{n\times(n+1)}$ is the completion of the product measure of the Lebesgue measures of $\mathbb{R}^{n\times n}$ and $\mathbb{R}^n$, we have that for almost every A, the set

$$\mathcal{V}_A := \{x \in \mathbb{R}^n \mid \det([x, Ax, A^2x, \ldots, A^{n-1}x]) = 0\}$$

has Lebesgue measure zero. Thus, there must exist a set $\mathcal{A} \subset \mathbb{R}^{n\times n}$ of Lebesgue measure zero such that if $A \notin \mathcal{A}$, then $\mathcal{V}_A$ has Lebesgue measure zero. Now assume that $A_\star \notin \mathcal{A}$ and let x be the initialization of our observed trajectory. Because x is sampled at random[6] from a full-dimensional infinite-step safe set, we have $\mathbb{P}(x \notin \mathcal{V}_{A_\star}) = 1$. When $x \notin \mathcal{V}_{A_\star}$, we have $\det([x, A_\star x, A_\star^2 x, \ldots, A_\star^{n-1}x])$ is nonzero, and therefore $[x, A_\star x, A_\star^2 x, \ldots, A_\star^{n-1}x]$ is invertible. Since we observe $[A_\star x, A_\star^2 x, \ldots, A_\star^n x]$, we can now recover $A_\star$ by solving a linear system

$$A_\star[x, A_\star x, A_\star^2 x, \ldots, A_\star^{n-1}x] = [A_\star x, A_\star^2 x, \ldots, A_\star^n x]$$
$$\Rightarrow A_\star = [A_\star x, A_\star^2 x, \ldots, A_\star^n x] \left([x, A_\star x, A_\star^2 x, \ldots, A_\star^{n-1}x]\right)^{-1}.$$

■

6. Any distribution that is absolutely continuous with respect to the Lebesgue measure would work; for example, the uniform distribution.

5.5. Numerical Examples

In this section, we present two numerical examples of infinite-step safe learning.

5.5.1. INNER AND OUTER APPROXIMATIONS OF THE INFINITE-STEP SAFE SET

In our first example, we take $n = 2$,

$$U_0 = \left\{ A \in \mathbb{R}^{2\times 2} \middle| A_{1,1} = A_{2,2} = \frac{1}{2}, A_{1,2}, A_{2,1} \geq 0, A_{1,2} + A_{2,1} = \frac{9}{5} \right\},$$

and

$$S = \left\{ x \in \mathbb{R}^2 \middle| \begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} x \leq \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}.$$

One can check that every matrix in U_0 is stable, though there are products of matrices in U_0 that have spectral radius greater than one and hence the techniques of Ahmadi and Günlük (2018) do not apply. We solve the semidefinite program (22) with degree $d = 4$ (in this example, the semidefinite program with $d = 2$ is infeasible). In Figure 4(a), we plot S, $\tilde{S}^{\infty}_{0,4}$, and $\tilde{S}^{\infty}(A)$ for various matrices A in U_0. We also plot $\bar{S}^{\infty}_0$ which is the intersection of $S^{10}(A)$ for various matrices A in U_0; in particular, $\bar{S}^{\infty}_0$ is an outer approximation of S^{∞}_0. This outer approximation is not too much larger than $\tilde{S}^{\infty}_{0,4}$, our inner approximation of S^{∞}_0.

In this example, we also observe that

$$\tilde{S}^{\infty}_{0,4} = \bigcap_{A \in U_0} \tilde{S}^{\infty}(A).$$

From the proof of Theorem 17, we have that $\tilde{S}^{\infty}_{0,d} \subseteq \bigcap_{A\in U_0} \tilde{S}^{\infty}(A)$ for all d. Therefore, this example shows not only that $d = 4$ is high enough to get a full-dimensional inner approximation of S^{∞}_0, but also that $d = 4$ is sufficient to get the largest possible infinite-step safe set based on our approach.

5.5.2. COMPARING ONE, TWO, AND INFINITE-STEP SAFETY

In our second example, we take $n = 2$,

$$U_0 = \left\{ A \in \mathbb{R}^{n\times n} \middle| \left\| A - \begin{bmatrix} 1 & .5 \\ -.5 & .5 \end{bmatrix} \right\|_F \leq 0.1 \right\},$$

and the same safety region S as in the previous example. We take $A_\star = \begin{bmatrix} 1.05 & .5 \\ -.5 & .5 \end{bmatrix} \in U_0$ and the initialization cost function to be given by the affine function $c(x) = (-1, 0)^T x + 3$, which is nonnegative over S.

Since the initial uncertainty set U_0 is not polyhedral, we replace the linear programs in Algorithm 1 for one-step safe learning with semidefinite programs. This is done by taking (13), discarding the two-step constraint, and then converting the resulting problem to a semidefinite program by the same method as in Theorem 9. Our algorithm then learns $A_\star$ with two one-step safe trajectories with a total initialization cost of 3.1489.

For two-step safe learning, we use Algorithm 3. We learn $A_\star$ with one two-step safe trajectory with an initialization cost of 1.9252.

For infinite-step safe learning, we adapt the method of Theorem 17 to the non-polyhedral set U_0 by multiplying the SOS matrices and polynomials in semidefinite program (22) by the polynomial

$$\left\{ A \mapsto 0.1^2 - \left\| A - \begin{bmatrix} 1 & .5 \\ -.5 & .5 \end{bmatrix} \right\|_F^2 \right\}$$

instead of $\{A \mapsto v_j - \mathrm{Tr}(V_j^T A)\}_{j=1}^s$. We learn $A_\star$ with one infinite-step safe trajectory with an initialization cost of 2.0080.

In this example, we see that two-step learning incurs the lowest total initialization cost. This is because a single two-step safe trajectory is sufficient for learning $A_\star$. Therefore, the initialization cost is incurred once as opposed to twice for one-step safe learning. Additionally, requiring two-step safety is less restrictive than requiring infinite-step safety resulting in lower cost compared to infinite-step safe learning.

In Figure 4(*b*), we plot S, S_0^1, S_1^1, S_0^2, $\tilde{S}_{0,2}^\infty$ and the initialization points chosen by each algorithm. We observe the inclusion relationships $\tilde{S}_{0,2}^\infty \subseteq S_0^2 \subseteq S_0^1 \subseteq S$ as expected. Note that S_0^1 and S_0^2 are exact characterizations of the one-step and two-step safety sets, respectively, while $\tilde{S}_{0,2}^\infty$ is an inner approximation of S_0^∞, the true infinite-step safety set. Since $\tilde{S}_{0,2}^\infty$ is not much smaller than S_0^2, which is a superset of S_0^∞, we see that our semidefinite representable set $\tilde{S}_{0,d}^\infty$ with $d = 2$ closely approximates the true infinite-step safety set S_0^∞.

6. One-Step Safe Learning of Nonlinear Systems

In this section, we turn our attention to the problem of safely learning a dynamical system of the form $x_{t+1} = f_\star(x_t)$, where

$$f_\star(x) = A_\star x + g_\star(x), \tag{27}$$

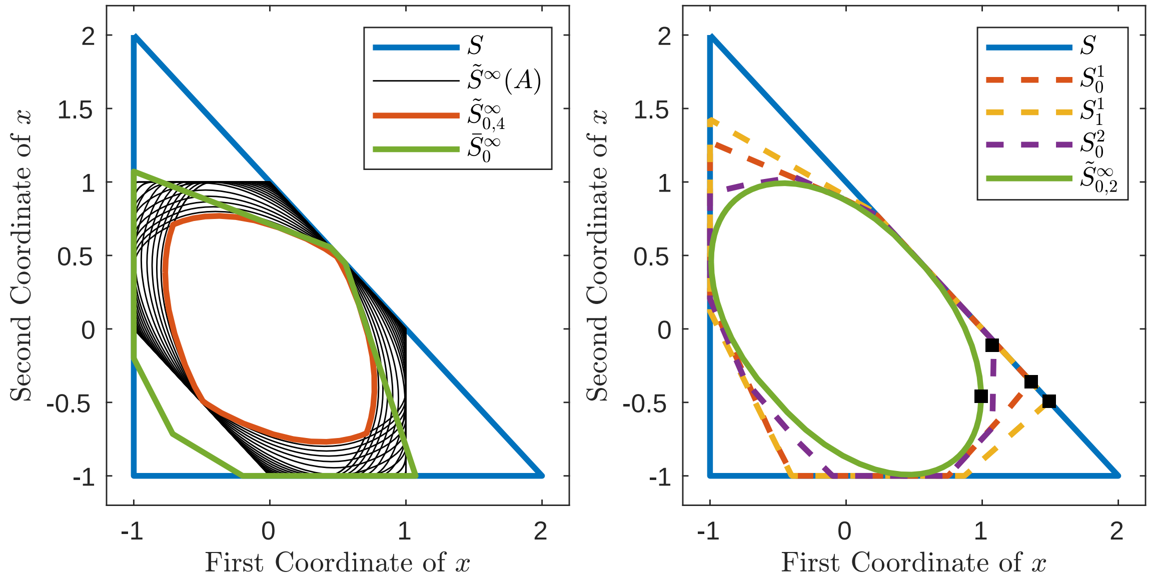
for some matrix $A_\star \in \mathbb{R}^{n\times n}$ and some possibly nonlinear map $g_\star : \mathbb{R}^n \to \mathbb{R}^n$. We take our safety region $S \subset \mathbb{R}^n$ to be the same as (5). Our initial knowledge about $A_\star, g_\star$ is membership in the sets

$$\begin{aligned} U_{0,A} &:= \left\{ A \in \mathbb{R}^{n\times n} \mid \mathrm{Tr}(V_j^T A) \le v_j \quad j = 1, \ldots, s \right\}, \\ U_{0,g} &:= \{ g : \mathbb{R}^n \to \mathbb{R}^n \mid \|g(x)\|_\infty \le \gamma \|x\|_p^d \quad \forall x \in S \}. \end{aligned}$$

Here, $p \ge 1$ is either $+\infty$ or a rational number, γ is a given positive constant, and d is a given nonnegative integer. The use of the $\|\cdot\|_\infty$ on g in the definition of $U_{0,g}$ simplifies some of the following analysis, though an extension to other semidefinite representable norms is possible. Note that by taking $d = 0$ e.g., our model of uncertainty captures any map f which is bounded on S. Again for simplicity, we assume a linear initialization cost $c^T x$ for some vector $c \in \mathbb{R}^n$.

Our goal in this section is to demonstrate how to safely collect one-step safe trajectories for (27) at minimum cost. By doing so, we reduce our uncertainty on $A_\star$ and are able to fit a parametric model to g that respects the constraints in $U_{0,g}$. Having collected k safe measurements $\{(x_j, y_j)\}_{j=1}^k$ with $y_j = f_\star(x_j)$, we can reduce our uncertainty around $A_\star$ as follows:

$$U_{k,A} = \{ A \in U_0 \mid \|Ax_j - y_j\|_\infty \le \gamma \|x_j\|_p^d \quad j = 1, \ldots, k \}.$$



(*a*) Inner approximation of the infinite-step safety set of a linear system associated with the example in Section 5.5.1.

(*b*) One-step and two-step safety sets and an inner approximation of the infinite-step safety set associated with the example in Section 5.5.2.

Figure 4: Infinite-step safe learning associated with the numerical examples in Section 5.5.

The optimization problem to find the next cheapest one-step safe initialization point is then:

$$\begin{aligned} \min_{x} \quad & c^T x \\ \text{s.t.} \quad & x \in S \\ & f(x) \in S \quad \forall\, f \in \{x \mapsto Ax + g(x) \mid A \in U_{k,A}, g \in U_{0,g}\}. \end{aligned} \tag{28}$$

We show next that an exact reformulation of this problem can be solved in an efficient manner.

6.1. Reformulation as a Second-Order Cone Program

Our main result of this section is to derive a tractable reformulation of problem (28).

Theorem 24 *Problem* (28) *can be reformulated as a second-order cone program.*

Proof We start by rewriting problem (28) using the definition of S:

$$\begin{aligned} \min_{x} \quad & c^T x \\ \text{s.t.} \quad & h_i^T x \le b_i \quad i = 1, \ldots, r \\ & \begin{bmatrix} \max_{A,g} & h_i^T(Ax + g(x)) \\ \text{s.t.} & \mathrm{Tr}(V_j^T A) \le v_j \quad j = 1, \ldots, s \\ & \|g(x)\|_\infty \le \gamma \|x\|_p^d \quad \forall x \in S \\ & Ax_k + g(x_k) = y_k \quad k = 1, \ldots, m \end{bmatrix} \le b_i \quad i = 1, \ldots, r. \end{aligned} \tag{29}$$

Note that in the inner maximization problem in (29), the variable x is fixed. We claim that if $x \notin \{x_1, \dots, x_k\}$, then

$$\begin{bmatrix} \max_{A,g} & h_i^T(Ax + g(x)) \\ \text{s.t.} & \mathrm{Tr}(V_j^T A) \leq v_j \quad j = 1, \dots, s \\ & \|g(x)\|_\infty \leq \gamma \|x\|_p^d \quad \forall x \in S \\ & Ax_k + g(x_k) = y_k \quad k = 1, \dots, m \end{bmatrix} \tag{30}$$
$$= \begin{bmatrix} \max_{A,g} & h_i^T Ax \\ \text{s.t.} & \mathrm{Tr}(V_j^T A) \leq v_j \quad \forall j \\ & Ax_k + g(x_k) = y_k \quad \forall k \\ & \|g(x)\|_\infty \leq \gamma \|x\|_p^d \quad \forall x \in S \end{bmatrix} + \begin{bmatrix} \max_{A,g} & h_i^T g(x) \\ \text{s.t.} & \mathrm{Tr}(V_j^T A) \leq v_j \quad \forall j \\ & Ax_k + g(x_k) = y_k \quad \forall k \\ & \|g(x)\|_\infty \leq \gamma \|x\|_p^d \quad \forall x \in S \end{bmatrix}.$$

It is clear that the left-hand side is upper bounded by the right-hand side. To show the reverse inequality, let (A_1, g_1) (resp. (A_2, g_2)) be feasible to the first (resp. second) problem on the right-hand side (if any of these of these problems is infeasible, then the inequality we are after is immediate). Now let

$$\hat{g}_2(x) = \begin{cases} g_2(x) & \text{if } x \notin \{x_1, \dots, x_k\} \\ y_k - A_1 x_k & \text{if } x = x_k. \end{cases}$$

It is straightforward to check that the pair $(A_1, \hat{g}_2)$ is feasible to the left-hand side of (30), therefore proving (30).

We now focus on reformulating each term on the right-hand side of (30), again under the assumption that $x \notin \{x_1, \dots, x_m\}$. Using the constraint on g, the first term can be rewritten as follows:

$$\begin{aligned} \max_{A} \quad & h_i^T Ax \\ \text{s.t.} \quad & \mathrm{Tr}(V_j^T A) \leq v_j \quad j = 1, \dots, s \\ & \|Ax_k - y_k\|_\infty \leq \gamma \|x_k\|_p^d \quad k = 1, \dots, m. \end{aligned} \tag{31}$$

Note that (31) is a linear program as it is equivalent to:

$$\begin{aligned} \max_{A} \quad & h_i^T Ax \\ \text{s.t.} \quad & \mathrm{Tr}(V_j^T A) \leq v_j \quad j = 1, \dots, s \\ & (Ax_k - y_k)_l \leq \gamma \|x_k\|_p^d \quad k = 1, \dots, m \quad l = 1, \dots, n \\ & -(Ax_k - y_k)_l \leq \gamma \|x_k\|_p^d \quad k = 1, \dots, m \quad l = 1, \dots, n. \end{aligned} \tag{32}$$

Here, the notation $(Ax_k - y_k)_l$ represents the l-th coordinate of the vector $(Ax_k - y_k)$. Following the same approach as in Section 3, we proceed by taking the dual of this linear program. For $j = 1, \dots, s$, $k = 1, \dots, m$, and $l = 1, \dots, n$, let $\mu_j, \eta_{kl}^+, \eta_{kl}^- \in \mathbb{R}$ be dual variables. The dual of problem (32) reads:

$$\begin{aligned} \min_{\mu, \eta^+, \eta^-} \quad & \sum_j \mu_j v_j + \sum_{kl} \eta_{kl}^+ (\gamma \|x_k\|_p^d + (y_k)_l) + \sum_{kl} \eta_{kl}^- (\gamma \|x_k\|_p^d - (y_k)_l) \\ \text{s.t.} \quad & x h_i^T = \sum_j \mu_j V_j^T + \sum_{kl} \eta_{kl}^+ x_k e_l^T - \sum_{kl} \eta_{kl}^+ x_k e_l^T \\ & \mu \geq 0, \quad \eta^+ \geq 0, \quad \eta^- \geq 0, \end{aligned} \tag{33}$$

where e_l is the l-th coordinate vector. Now we turn our attention to the second term on the right-hand side of (30). After eliminating the irrelevant constraints, the problem can be rewritten as:

$$\begin{aligned} \max_{g} \quad & h_i^T g(x) \\ \text{s.t.} \quad & \|g(x)\|_\infty \leq \gamma \|x\|_p^d. \end{aligned} \tag{34}$$

Recall that the dual norm of $\|\cdot\|_\infty$ is $\|\cdot\|_1$. Therefore, the optimal value of this optimization problem is simply $\gamma\|h_i\|_1 \cdot \|x\|_p^d$.

Now consider the optimization problem:

$$\begin{aligned} \min_{x,\mu,\eta^+,\eta^-} \quad & c^T x \\ \text{s.t.} \quad & h_i^T x \leq b_i \quad i = 1, \dots, r \\ & \sum_j \mu_j v_j + \sum_{kl} \eta_{kl}^+ (\gamma \|x_k\|_p^d + (y_k)_l) \\ & \qquad + \sum_{kl} \eta_{kl}^- (\gamma \|x_k\|_p^d - (y_k)_l) + \gamma \|h_i\|_1 \cdot \|x\|_p^d \leq b_i \quad i = 1, \dots, r \\ & \mu \geq 0, \quad \eta^+ \geq 0, \quad \eta^- \geq 0. \end{aligned} \tag{35}$$

If $d = 0$, or if $d = 1$ and $p \in \{1, +\infty\}$, then (35) is a linear program. Otherwise, the rationality of p ensures that $\|x\|_p^d$ is *second-order cone representable* (see Ben-Tal and Nemirovski, 2001, Sect. 2.3; Lobo et al., 1998, Sect. 2.5). This means that (35) is indeed a second-order cone program.

Let $F \subset \mathbb{R}^n$ denote the projection of the feasible set of (35) to x-space. We claim that the feasible set of (28) equals $F \cup \{x_1, \dots, x_k\}$. Indeed, since the vectors x_k are one-step safe initialization points, we have that $x_k \in S$ and $y_k \in S$. This implies that x_k is feasible to (28). Furthermore, for $x \in F \setminus \{x_1, \dots, x_k\}$, we have shown that x satisfies the constraints of (29) if and only if x satisfies the constraints of (35).

Therefore, optimizing an objective function over the feasible set of (28) is equivalent to optimizing the same objective function over $F \cup \{x_1, \dots, x_k\}$. ■

6.2. Numerical Example

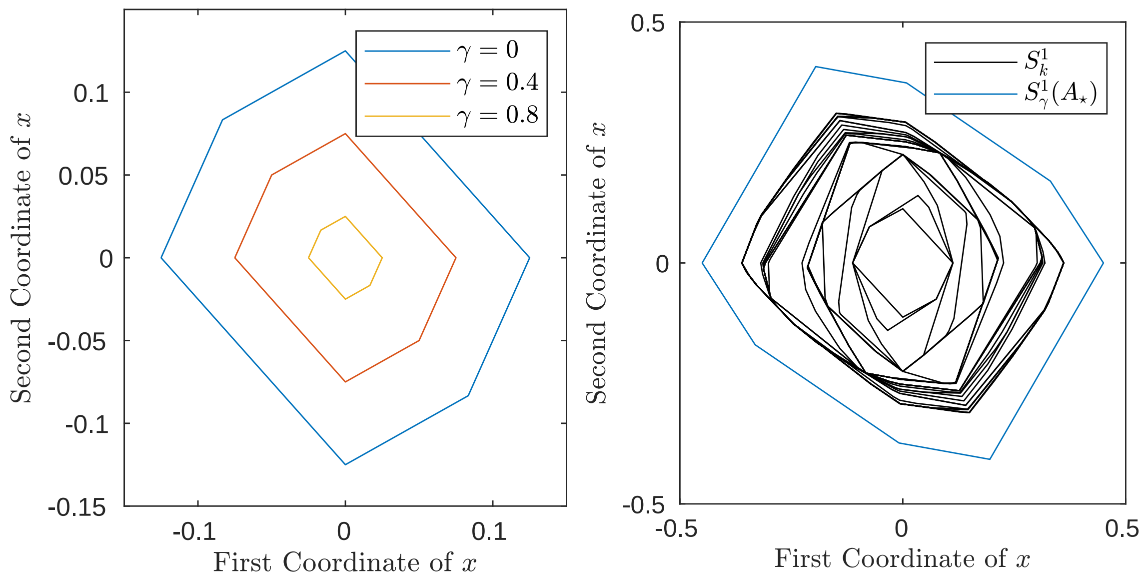
We present a numerical example with $n = 4$. We take

$$\begin{aligned} S &= \{x \in \mathbb{R}^4 \mid |x_i| \leq 1, i = 1, \dots, 4\}, \\ U_{0,A} &= \{A \in \mathbb{R}^{4\times 4} \mid -4 \leq A_{ij} \leq 8, i = 1, \dots, 4, j = 1, \dots, 4\}, \\ U_{0,g} &= \{g : \mathbb{R}^4 \to \mathbb{R}^4 \mid \|g(x)\|_\infty \leq \gamma \quad \forall x \in S\}. \end{aligned}$$

In Figure 5(*a*), we plot S_0^1 (the one-step safety region without any measurements) projected to the first two dimensions of x for $\gamma \in \{0, 0.4, 0.8\}$. As expected, larger values of γ result in smaller one-step safety regions.

For our next experiment, we choose the matrix $A_\star$ in (27) to be the same matrix used in the example in Section 3.5. We let $\gamma = 0.1$, and

$$g_\star(x) = \frac{\gamma}{2} \left(x_2^2 - x_3 x_4, \quad \sqrt{x_1^4 + x_3^4}, \quad x_3 \sin^2(x_1), \quad \sin^2(x_2) \right)^T \in U_{0,g}.$$



(a) Dependence of S_0^1 on γ.

(b) S_k^1 grows with k.

Figure 5: One-step safe learning of a nonlinear system associated with the example in Section 6.2.

Since the true system is not linear, we cannot hope to learn the dynamics in n steps as we did in the linear case. We instead pick 30 one-step safe points $x_1, \ldots, x_{30}$ (by sequentially solving the second-order cone program from Theorem 24) and observe $y_k = f_\star(x_k)$ for each $k = 1, \ldots, 30$. In order to encourage exploration of the state space, we optimize in random directions in every iteration (instead of optimizing the same cost function throughout the process). In Figure 5(b), we plot S_k^1 (the one-step safety region after k measurements) projected to the first two dimensions of x for $k = 0, \ldots, 30$. Note that S_k^1 is the projection of the feasible set of (35) to x-space. We also plot the projection of $S_\gamma^1(A_\star)$, which we define as the set of one-step safe points if we knew $A_\star$, but not $g_\star$

$$S_\gamma^1(A_\star) := \{x \in S \mid A_\star x + g(x) \in S \quad \forall g \in U_{0,g}\}.$$

Note that this set is an outer approximation to S_k^1. Here we see that S_k^1 comes close to $S_\gamma^1(A_\star)$ already in thirty iterations.

Finally, we undertake the task of learning the unknown nonlinear dynamics. We only use information from our first 8 data points in order to make the fitting task more challenging. We fit a function of the form

$$\hat{f}(x) = \hat{A}x + \hat{g}(x),$$

where $\hat{A} \in \mathbb{R}^{4\times 4}$ and each entry of $\hat{g} : \mathbb{R}^4 \to \mathbb{R}^4$ is a homogeneous quadratic function of x. Our regression is done by minimizing the least-squares loss function

$$L(\hat{f}) = \sum_{k=1}^{8} \|\hat{f}(x_k) - y_k\|^2.$$

We train two models. The first model, $\hat{f}_{\text{ls}}$, minimizes the least-squares loss with no constraints. The second model, $\hat{f}_{\text{SOS}}$, minimizes the least-squares loss subject to the constraints that $\hat{A} \in U_{0,A}$,

$\|\hat{A}x_k - y_k\|_\infty \leq \gamma$ for $k = 1, \ldots, 8$, and $\hat{g} \in U_{0,g}$. The constraint that $\hat{g} \in U_{0,g}$ is imposed via sum of squares constraints (see, e.g., Parrilo, 2000; Ahmadi and El Khadir, 2023 for details). More specifically, we require that for $j = 1, \ldots, 4$,

$$\gamma \pm \hat{g}_j(x) = \sigma_0^{j,\pm}(x) + \sum_{i=1}^{r} \sigma_i^{j,\pm}(x)(b_i - h_i^T x) \quad \forall x \in \mathbb{R}^4.$$

Here, $\hat{g}_j(x)$ is the j-th entry of the vector $\hat{g}(x)$, and the functions $\sigma_i^{j,\pm}$, for $i = 0, \ldots, r$ and $j = 1, \ldots, 4$, are sum of squares quadratic functions of x. These constraints can be imposed by semidefinite programming.

We sample test points $z_1, \ldots, z_{1000}$ uniformly at random in S in order to estimate the generalization error. The root-mean-square error (RMSE) is computed as

$$\text{RMSE}(\hat{f}) = \sqrt{\frac{1}{1000} \sum_{j=1}^{1000} \|\hat{f}(z_i) - f_\star(z_i)\|^2}.$$

The RMSE($\hat{f}_{\text{SOS}}$) of the constrained model is 0.0851 and the RMSE($\hat{f}_{\text{ls}}$) of the unconstrained model is 0.2567. We see that imposing prior knowledge with sum of squares constraints results in a significantly better fit.

7. Infinite-Step Safe Learning of Nonlinear Systems

In our final technical section, we consider the problem of safely learning a dynamical system of the same form as in Section 6, i.e.,

$$x_{t+1} = A_\star x_t + g_\star(x_t) \tag{36}$$

involving some matrix $A_\star \in \mathbb{R}^{n \times n}$ and some possibly nonlinear map $g_\star : \mathbb{R}^n \to \mathbb{R}^n$. We take the safety region $S \subset \mathbb{R}^n$ to be the same as (5). We take our initial knowledge about $A_\star, g_\star$ to be membership in the following sets:

$$\begin{aligned} U_{0,A} &:= \left\{ A \in \mathbb{R}^{n \times n} \mid \text{Tr}(V_j^T A) \leq v_j \quad j = 1, \ldots, s \right\}, \\ U_{0,g} &:= \{ g : \mathbb{R}^n \to \mathbb{R}^n \mid \|g(x)\| \leq \gamma \|x\| \quad \forall x \in S \}, \end{aligned}$$

where γ is a given nonnegative constant. In the definition of $U_{0,g}$, it is convenient to use the ℓ_2 norm because of some technical reasons that will become clear in the proofs of the statements in this section. Again for simplicity, we assume that for some vector $c \in \mathbb{R}^n$, initializing the system at a point $x \in S$ comes at the cost $c^T x$.

Just as in Section 5, the notion of safety in this section is that of infinite-step safety; i.e., we can only initialize the system at points whose entire trajectory will remain in S under all dynamical systems consistent with the information at hand. By observing these trajectories, we can safely reduce our uncertainty on $A_\star$ and fit a parametric model to $g_\star$ that respects the constraints in $U_{0,g}$ (in the same way that we did in Section 6). Unlike the setting of infinite-step safe learning of linear systems (Section 5), it might be useful to observe trajectories of length greater than n. Assuming there is some limitation on memory, it is sensible to truncate each trajectory after some time. Suppose that we have collected k trajectories and that the ℓth trajectory is of length n_ℓ. Let $y_{t,\ell} \in \mathbb{R}^n$ be the

tth observed vector in the ℓth trajectory with $y_{0,\ell}$ being the trajectory's initialization. With these observations, we can reduce our uncertainty around $A_\star$ as follows:

$$U_{k,A} = \{A \in U_0 \mid \|Ay_{t-1,\ell} - y_{t,\ell}\| \leq \gamma \|y_{t-1,\ell}\| \quad t = 1, \dots, n_\ell \quad \ell = 1, \dots, k\}.$$

Since $U_{0,g}$ contains the zero map (i.e., one possibility for the unknown dynamics is $x_{t+1} = Ax_t$ for some matrix $A \in U_{0,A}$), the problem considered in this section is at least as hard as that of Section 5. Therefore, we make the same assumptions as Section 5 to have a full-dimensional infinite-step safety set. In particular, we assume that the origin is in the interior of S, which means that S can be described as (21), and that all matrices in $U_{0,A}$ are stable. Having collected k infinite-step safe trajectories, the optimization problem we are interested in solving to find the next cheapest infinite-step safe initialization point is:

$$\begin{aligned} \min_{x} \quad & c^T x \\ \text{s.t.} \quad & x \in S \\ & f^t(x) \in S \quad \forall\, f \in \{x \mapsto Ax + g(x) \mid A \in U_{k,A}, g \in U_{0,g}\} \quad t = 1, 2, \dots. \end{aligned} \tag{37}$$

In keeping with the naming conventions of this work, we refer to the feasible region of (37) as S_k^∞, and if $U_{0,A}$ is a single matrix A, we call it $S_\gamma^\infty(A)$. We can now present the main theorem of this section, which enables us to find infinite-step safe initialization points.

Theorem 25 *Let the polyhedron $S \subseteq \mathbb{R}^n$ be as in (21), and the polyhedron $U_{0,A} \subseteq \mathbb{R}^{n \times n}$ be as in (4). For $\ell = 1, \dots, k$ and $t = 0, \dots, n_\ell$, let $y_{t,\ell}$ be the tth vector in the ℓth observed trajectory. For an even integer d, let $\tilde{S}_{k,d}^\infty$ be the projection to x-space of the feasible region of the following optimization problem:*

$$\min_{x,Q,M_j,M_{t\ell},\hat{M}_j,\hat{M}_{t\ell},\sigma_{ij},\sigma_{it\ell},\varepsilon,\lambda} \quad c^T x \tag{38}$$

s.t. $\quad Q(A)$ is an $n \times n$ SOS matrix with degree $\leq d$ (38a)

$M_j(A)$ are $2n \times 2n$ SOS matrices with degree $\leq d \quad j = 0, \dots, s$ (38b)

$M_{t\ell}(A)$ are $2n \times 2n$ SOS matrices with degree $\leq d$ (38c)

$\quad \ell = 1, \dots, k \quad t = 1, \dots, n_\ell$

$\hat{M}_j(A)$ are $(n+1) \times (n+1)$ SOS matrices with degree $\leq d \quad j = 0, \dots, s$ (38d)

$\hat{M}_{t\ell}(A)$ are $(n+1) \times (n+1)$ SOS matrices with degree $\leq d$ (38e)

$\quad \ell = 1, \dots, k \quad t = 1, \dots, n_\ell$

$\sigma_{ij}(A)$ are SOS polynomials with degree $\leq d \quad i = 1, \dots, r \quad j = 0, \dots, s$ (38f)

$\sigma_{it\ell}(A)$ are SOS polynomials with degree $\leq d$ (38g)

$\quad i = 1, \dots, r \quad \ell = 1, \dots, k \quad t = 1, \dots, n_\ell$

$$\begin{bmatrix} Q(A) - AQ(A)A^T & -AQ(A) \\ -Q(A)A^T & -Q(A) \end{bmatrix} - \lambda \begin{bmatrix} \gamma^2 I & 0 \\ 0 & -I \end{bmatrix} = \varepsilon I + M_0(A) + \sum_{j=1}^{s} M_j(A)(v_j - \mathrm{Tr}(V_j^T A)) \tag{38h}$$

$$+ \sum_{\ell=1}^{k} \sum_{t=1}^{n_\ell} M_{t\ell}(A)(\gamma^2 \|y_{t-1,\ell}\|^2 - \|Ay_{t-1,\ell} - y_{t,\ell}\|^2) \quad \forall A \in \mathbb{R}^{n \times n}$$

$$1 - h_i^T Q(A) h_i = \sigma_{i0}(A) + \sum_{j=1}^{s} \sigma_{ij}(A)(v_j - \mathrm{Tr}(V_j^T A)) \tag{38i}$$

$$+ \sum_{\ell=1}^{k} \sum_{t=1}^{n_\ell} \sigma_{it\ell}(A)(\gamma^2 \|y_{t-1,\ell}\|^2 - \|Ay_{t-1,\ell} - y_{t,\ell}\|^2) \quad i = 1, \dots, r \quad \forall A \in \mathbb{R}^{n \times n}$$

$$\begin{bmatrix} Q(A) & x \\ x^T & 1 \end{bmatrix} = \hat{M}_0(A) + \sum_{j=1}^{s} \hat{M}_j(A)(v_j - \mathrm{Tr}(V_j^T A)) \tag{38j}$$

$$+ \sum_{\ell=1}^{k} \sum_{t=1}^{n_\ell} \hat{M}_{t\ell}(A)(\gamma^2 \|y_{t-1,\ell}\|^2 - \|Ay_{t-1,\ell} - y_{t,\ell}\|^2) \quad \forall A \in \mathbb{R}^{n \times n}$$

$$\varepsilon > 0 \tag{38k}$$

$$\lambda \geq 0. \tag{38l}$$

Then,

(i) The program (38) can be reformulated as a semidefinite program of size polynomial in the size of the input (S, U_0, $\{y_{t,\ell}\}$, c, γ).

(ii) We have $\tilde{S}^{\infty}_{k,d} \subseteq S^{\infty}_k$ (i.e, any vector x feasible to this semidefinite program is infinite-step safe).

(iii) *Furthermore, if $U_{0,A}$ is compact and contains only stable matrices, and if γ is smaller than some positive threshold depending on $U_{0,A}$, then, for large enough d, the set $\tilde{S}^{\infty}_{k,d}$ is full-dimensional.*

In words, Theorem 25 allows us to optimize initialization cost over semidefinite representable subsets of the set of points which are infinite-step safe under all nonlinear dynamics consistent with information at hand. While the theorem guarantees full-dimensionality of these subsets for sufficiently small γ and large d, in our experience, even when γ is relatively large, small values of d suffice for safe learning; see Section 6.2.

Before we present a proof of Theorem 25, we introduce a "nonlinear version" of (23), which applies to the case of a fixed matrix A:

$$\begin{aligned}
\min_{x\in\mathbb{R}^n, Q\in\mathbb{S}^{n\times n}, \lambda\in\mathbb{R}} \quad & c^T x \\
\text{s.t.} \quad & Q \succ 0 \\
& \begin{bmatrix} Q - AQA^T & -AQ \\ -QA^T & -Q \end{bmatrix} \succeq \lambda \begin{bmatrix} \gamma^2 I & 0 \\ 0 & -I \end{bmatrix} \\
& h_i^T Q h_i \leq 1 \quad i = 1, \ldots, r \\
& \begin{bmatrix} Q & x \\ x^T & 1 \end{bmatrix} \succeq 0 \\
& \lambda \geq 0.
\end{aligned} \tag{39}$$

We now prove a nonlinear version of Lemma 18. Recall the definition of the set $S^{\infty}_{\gamma}(A)$ from the paragraph after (37).

Lemma 26 *Let $\tilde{S}^{\infty}_{\gamma}(A)$ be the projection to x-space of the feasible region of (39). Then, we have $\tilde{S}^{\infty}_{\gamma}(A) \subseteq S^{\infty}_{\gamma}(A)$.*

Proof Let x, Q, and λ be feasible to (39). As in the proof of Lemma 18, the constraints $h_i^T Q h_i \leq 1$, $i = 1, \ldots, r$, imply

$$\{x \mid x^T Q^{-1} x \leq 1\} \subseteq S,$$

and the constraint $\begin{bmatrix} Q & x \\ x^T & 1 \end{bmatrix} \succeq 0$ implies $x^T Q^{-1} x \leq 1$. Thus x is in the safety region S. To show that the trajectory remains safe for all time, it suffices to show that the set $\{x \mid x^T Q^{-1} x \leq 1\}$ is invariant under all valid dynamics, i.e., for any vector $\bar{x}$ in this set and any vector $g(\bar{x})$ satisfying $\|g(\bar{x})\| \leq \gamma\|\bar{x}\|$, the vector $A\bar{x} + g(\bar{x})$ is also in the set.

Let $B \in \mathbb{R}^{n\times n}$ be an arbitrary matrix and let $\|B\|$ denote its spectral norm. We first claim that if $\|B\| \leq \gamma$, then $(A + B)^T Q^{-1}(A + B) \preceq Q^{-1}$. Fix an arbitrary vector $\hat{x}$ and let $\hat{y} = B^T\hat{x}$. By the bound on the spectral norm of B, we have $\|\hat{y}\| \leq \gamma\|\hat{x}\|$. By the second linear matrix inequality in (39), we have

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}^T \begin{bmatrix} Q - AQA^T & -AQ \\ -QA^T & -Q \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} \geq \lambda \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}^T \begin{bmatrix} \gamma^2 I & 0 \\ 0 & -I \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}.$$

Rearranging, we have

$$\hat{x}^T Q\hat{x} - (\hat{x}^T AQA^T\hat{x} + 2\hat{x}^T AQ\hat{y} + \hat{y}^T Q\hat{y}) \geq \lambda(\gamma^2\hat{x}^T\hat{x} - \hat{y}^T\hat{y}).$$

Since $\lambda \geq 0$ and $\|\hat{y}\| \leq \gamma\|\hat{x}\|$, we have $\hat{x}^T Q\hat{x} \geq \hat{x}^T AQA^T\hat{x} + 2\hat{x}^T AQ\hat{y} + \hat{y}^T Q\hat{y}$, which implies

$$\hat{x}^T(A+B)Q(A+B)^T\hat{x} \leq \hat{x}^T Q\hat{x}.$$

This is equivalent to the claimed linear matrix inequality by two applications of the Schur complement lemma.

Now we show invariance of the set $\{x \mid x^T Q^{-1}x \leq 1\}$ under all valid dynamics. Let $\bar{x}$ be any vector such that $\bar{x}^T Q^{-1}\bar{x} \leq 1$, and let $B_{\bar{x}} := \frac{g(\bar{x})\bar{x}^T}{\|\bar{x}\|^2}$. From the definition of $U_{0,g}$, we have $\|B_{\bar{x}}\| \leq \gamma$. By the claim we established above, we have $\bar{x}^T(A+B_{\bar{x}})^T Q^{-1}(A+B_{\bar{x}})\bar{x} \leq \bar{x}^T Q^{-1}\bar{x}$. Since $(A+B_{\bar{x}})\bar{x} = A\bar{x} + g(\bar{x})$, it follows that

$$\big(A\bar{x} + g(\bar{x})\big)^T Q^{-1}\big(A\bar{x} + g(\bar{x})\big) \leq \bar{x}^T Q^{-1}\bar{x} \leq 1.$$

Thus, $A\bar{x} + g(\bar{x})$ is in the set $\{x \mid x^T Q^{-1}x \leq 1\}$ as desired. ■

We can now present the proof of the main result of this section.

Proof [Proof of Theorem 25]

(i) We make a similar argument as in the proof of (i) in Theorem 17. Recall that for any fixed degree d, the SOS constraints can be reformulated as semidefinite programming constraints of size polynomial in n. The "$\forall A$" constraints can be imposed by coefficient matching via a number of linear equations bounded by a polynomial function of the size of the input $(S, U_{0,A}, \{y_{t,\ell}\}, c, \gamma)$. The constraint that $\varepsilon > 0$ can be rewritten as the constraint $\begin{bmatrix}\varepsilon & 1\\ 1 & \delta\end{bmatrix} \succeq 0$ for a new variable δ. Therefore, for any fixed degree d, (22) is a semidefinite program of size polynomial in the size of the input $(S, U_{0,A}, \{y_{t,\ell}\}, c, \gamma)$.

(ii) Let $(x, Q, M_j, M_{t\ell}, \hat{M}_j, \hat{M}_{t\ell}, \sigma_{ij}, \sigma_{it\ell}, \varepsilon, \lambda)$ be feasible to (38). Then, it is straightforward to check that for every $A \in U_{k,A}$, the tuple $(x, Q(A), \lambda)$ satisfies the following constraints:

$$\begin{aligned}
&Q(A) \succ 0\\
&\begin{bmatrix} Q(A) - AQ(A)A^T & -AQ(A)\\ -Q(A)A^T & -Q(A)\end{bmatrix} \succeq \lambda\begin{bmatrix}\gamma^2 I & 0\\ 0 & -I\end{bmatrix}\\
&h_i^T Q(A)h_i \leq 1 \quad i = 1,\dots,r\\
&\begin{bmatrix} Q(A) & x\\ x^T & 1\end{bmatrix} \succeq 0,
\end{aligned} \tag{40}$$

and therefore, by Lemma 26, we have $x \in \tilde{S}^\infty_\gamma(A) \subseteq S^\infty_\gamma(A)$. Hence,

$$x \in \bigcap_{A\in U_{k,A}} S^\infty_\gamma(A) = S^\infty_k,$$

implying that $\tilde{S}^\infty_{k,d} \subseteq S^\infty_k$.

(iii) Suppose that $U_{0,A}$ is compact and contains only stable matrices. By Lemma 22 and the arguments in the beginning of the proof of (iii) in Theorem 17, we can find an SOS matrix $\hat{Q}(A)$ satisfying

$$\begin{aligned}
&\hat{Q}(A) \succ 0 \quad \forall A \in U_{0,A}\\
&\hat{Q}(A) \succ A\hat{Q}(A)A^T \quad \forall A \in U_{0,A}.
\end{aligned}$$

In particular, there must be a positive constant δ such that $\hat{Q}(A) - A\hat{Q}(A)A^T \succeq \delta I$ for all $A \in U_{0,A}$. Let $\hat{\lambda} := 1 + \max_{A \in U_{0,A}} \|\hat{Q}(A)A^T(\hat{Q}(A) - A\hat{Q}(A)A^T - \frac{\delta}{2}I)^{-1}A\hat{Q}(A) + \hat{Q}(A)\|$, and take γ to be a positive scalar less than $\sqrt{\frac{\delta}{2\hat{\lambda}}}$. By the Schur complement lemma and the fact that $\frac{\delta}{2} > \hat{\lambda}\gamma^2$, it follows that for every $A \in U_{0,A}$,

$$\begin{bmatrix} \hat{Q}(A) - A\hat{Q}(A)A^T & -A\hat{Q}(A) \\ -\hat{Q}(A)A^T & -\hat{Q}(A) \end{bmatrix} \succeq \begin{bmatrix} \frac{\delta}{2}I & 0 \\ 0 & -(\hat{\lambda}-1)I \end{bmatrix} \succ \hat{\lambda} \begin{bmatrix} \gamma^2 I & 0 \\ 0 & -I \end{bmatrix}.$$

Since $U_{k,A}$ is compact, there exists a scalar $\alpha > 0$ satisfying $\max_{i \in \{1,\ldots,r\}, A \in U_{k,A}} h_i^T \hat{Q}(A) h_i < \alpha$. Let us define $Q := \hat{Q}/\alpha$ and $\lambda := \hat{\lambda}/\alpha$. Since $Q(A) \succ 0$ for all $A \in U_{k,A}$, we can find a scalar $\beta > 0$ such that $\begin{bmatrix} Q(A) - \beta I & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \succ 0$ for all $A \in U_{k,A}$. Summarizing, so far we have:

$$\begin{aligned} \begin{bmatrix} Q(A) - AQ(A)A^T & -AQ(A) \\ -Q(A)A^T & -Q(A) \end{bmatrix} - \lambda \begin{bmatrix} \gamma^2 I & 0 \\ 0 & -I \end{bmatrix} &\succ 0 \quad \forall A \in U_{k,A} \\ 1 - h_i^T Q(A) h_i &> 0 \quad \forall A \in U_{k,A} \quad i = 1, \ldots, r \\ \begin{bmatrix} Q(A) - \beta I & 0 \\ 0 & \frac{1}{2} \end{bmatrix} &\succ 0 \quad \forall A \in U_{k,A}. \end{aligned} \tag{41}$$

Since $U_{0,A}$ is a bounded polyhedron, the set of inequalities that define it (i.e., $\{A \to v_j - \mathrm{Tr}(V_j^T A)\}_{j=1}^s$) satisfy the Archimedian property. Consider the following set of polynomials:

$$\mathcal{G} := \{A \to v_j - \mathrm{Tr}(V_j^T A)\}_{j=1}^s \cup \bigcup_{\ell=1}^{k} \bigcup_{t=1}^{n_\ell} \{A \to \gamma^2 \|y_{t-1,\ell}\|^2 - \|Ay_{t-1,\ell} - y_{t,\ell}\|^2\}.$$

As a superset of a set satisfying the Archimedian property, this set also satisfies the Archimedian property. By applying Theorem 20 and Theorem 21 to (41), we can conclude the existence of SOS matrices, $M_j, M_{t\ell}, \hat{M}_j, \hat{M}_{t\ell}$, and SOS polynomials, $\sigma_{ij}, \sigma_{it\ell}$ of some degree d, and a positive scalar ε satisfying (38h), (38i), (38k), and the following:

$$\begin{aligned} \begin{bmatrix} Q(A) - \beta I & 0 \\ 0 & \frac{1}{2} \end{bmatrix} &= \hat{M}_0(A) + \sum_{j=1}^{s} \hat{M}_j(A)(v_j - \mathrm{Tr}(V_j^T A)) \\ &\quad + \sum_{\ell=1}^{k} \sum_{t=1}^{n_\ell} \hat{M}_{t\ell}(A)(\gamma^2 \|y_{t-1,\ell}\|^2 - \|Ay_{t-1,\ell} - y_{t,\ell}\|^2) \quad \forall A \in \mathbb{R}^{n \times n}. \end{aligned}$$

Note that for any x satisfying $\|x\| \leq \sqrt{\frac{1}{2\beta}}$, we have that $\begin{bmatrix} \beta I & x \\ x^T & \frac{1}{2} \end{bmatrix} \succeq 0$. Therefore,

$$A \mapsto \hat{M}_0(A) + \begin{bmatrix} \beta I & x \\ x^T & \frac{1}{2} \end{bmatrix}$$

is still an SOS matrix of the same degree as $\hat{M}_0(A)$. Hence, for any x satisfying $\|x\| \leq \sqrt{\frac{1}{2\beta}}$, we have that the tuple

$$\left(x, Q, M_j, M_{t\ell}, \hat{M}_0(A) + \begin{bmatrix} \beta I & x \\ x^T & \frac{1}{2} \end{bmatrix}, \hat{M}_j, \hat{M}_{t\ell}, \sigma_{ij}, \sigma_{it\ell}, \varepsilon, \lambda\right)$$

is feasible to (38). ■

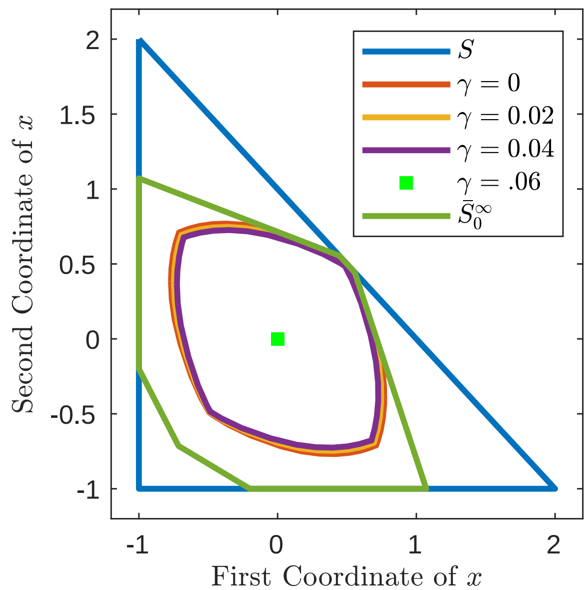


Figure 6: The numerical example in Section 7.1: the safety set S, the sets $\tilde{S}_{0,4}^\infty$ for four different values of γ, and the set $\bar{S}_0^\infty$, which is an outer approximation to S_0^∞ for any value of γ.

7.1. Numerical Example

We present a numerical example with $n = 2$. Here we take S and $U_{0,A}$ to be the same as S and U_0 in Section 5.5.1. We solve the semidefinite program in (38) with degree $d = 4$ (the program with $d = 2$ is infeasible). In Figure 7.1, we plot the safety region S, and our semidefinite programming based inner approximations $\tilde{S}_{0,4}^\infty$ of the infinite-step safe set S_0^∞ for $\gamma = 0, 0.02, 0.04, 0.06$. We also plot a set $\bar{S}_0^\infty$, which is the same outer approximation of S_0^∞ as in Section 5.5.1. Note that $\bar{S}_0^\infty$ is an outer approximation of S_0^∞ for any value of γ.

For $\gamma = 0.06$, our semidefinite program is infeasible and therefore we can only certify that the origin is infinite-step safe. This is intended behavior since for $\gamma = 0.06$, the true infinite-step safe set is just the origin. To see why, observe that if $A_\star = \begin{bmatrix} 0.5 & 0.45 \\ 0.45 & 0.5 \end{bmatrix} \in U_{0,A}$, and $g_\star(x) = 0.055 * x \in U_{0,g}$, then we have $f_\star(x) = \begin{bmatrix} 0.555 & 0.45 \\ 0.45 & 0.555 \end{bmatrix} x$ which is unstable since $\rho\left(\begin{bmatrix} 0.555 & 0.45 \\ 0.45 & 0.555 \end{bmatrix}\right) > 1$. This means that the true infinite-step safe set is not full-dimensional (see Preposition 15). By slightly perturbing $g_\star$ within $U_{0,g}$, we can obtain another valid unstable linear map $\hat{f}$ whose lower-dimensional stable subspace is different than that of $f_\star$. Therefore, when $\gamma = 0.06$, the true infinite-step safe set is indeed just the origin.

For $\gamma = 0.02$ or 0.04 for example, and for any nonlinear system of the type (36), with $A_\star \in U_{0,A}$ and $g_\star \in U_{0,g}$, we can choose initialization points within our full-dimensional sets $\tilde{S}_{0,4}^\infty$ and safely observe their trajectories. Having safely collected trajectory data, following the same exact procedure as in Section 6.2, we can narrow the uncertainty on the linear part of the dynamics and use semidefinite programming to fit a polynomial map to the nonlinear part of the dynamics in such a way that the information in $U_{0,g}$ is respected and the error on the observations is minimized.

8. Future Research Directions

We conclude with a discussion of potential directions for future research:

- While we considered safely learning autonomous systems, a natural direction is to extend our model and techniques to controlled systems as well. Specifically, in our models, how would our conic optimization problems have to change if the goal is to select a sequence of admissible control inputs that simultaneously maintains the safety of the system while providing the necessary excitation to reduce uncertainty? One can show that in the case of linear systems, where $x_{t+1} = Ax_t + Bu_t$ with given initial uncertainty sets for matrices A and B and given safety regions for both the state x and the control u, the problem of one-step safe learning of the dynamics reduces to the uncontrolled setting studied in Section 3 by state augmentation. Extensions of the other settings considered in this paper are left for future work.

- Can one bound the suboptimality of our greedy online algorithm for minimizing the cost of safe learning against the idealized minimum cost of safe learning (cf. the paragraph above Eq. (2) and e.g., problem (11))? How would this bound depend on the input parameters S, U_0, T, and the true dynamics $f_\star$? Furthermore, since it is not clear if any possible algorithm can achieve the idealized minimum cost of safe learning for every $f_\star \in U_0$, one could instead consider comparing to the best valid algorithm for safe learning that achieves the lowest worst-case (minimax) cost over all $f_\star \in U_0$. How would the greedy algorithm compare to this minimax optimal algorithm?

- In Sections 6 and 7, we studied systems which consist of a linear term plus a nonlinear term with bounded growth. While this description is fairly general, further specialization to practical nonlinear systems, such as piecewise affine systems or systems parameterized with a known set of basis functions, could potentially allow one to safely recover the nonlinear part of the dynamics.[7]

- In Sections 5 and 7, we approximated infinite-step safe sets by deriving semidefinite programs whose size depend on the maximum allowed degree of certain SOS polynomials and matrices. While we found small degrees to suffice empirically, it would be interesting to bound, for some class of problem instances, the degree one must choose in order for the proposed approximation to the safe sets to be full-dimensional (assuming the true set is full-dimensional).

- In this work, we considered the problem of safe learning for deterministic discrete-time dynamical systems. Extending our framework and techniques to stochastic and/or continuous-time systems would broaden the scope of our work.

7. Under suitable growth assumptions, one could apply the results in Section 6 (resp. Section 7) to derive inner approximations of the one (resp. infinite) step safe set of e.g. a piecewise affine system. However, by specializing the description of the system, it may be possible to derive less conservative inner approximations (or even exact characterizations).

Acknowledgments

AAA and AC were partially supported by the MURI award of the AFOSR, the DARPA Young Faculty Award, the CAREER Award of the NSF, the Google Faculty Award, the Innovation Award of the School of Engineering and Applied Sciences at Princeton University, and the Sloan Fellowship.

References

Amir Ali Ahmadi and Bachir El Khadir. Learning dynamical systems with side information. *SIAM Review*, 65(1):183–223, 2023.

Amir Ali Ahmadi and Oktay Günlük. Robust-to-dynamics optimization. *arXiv:1805.03682*, 2018.

Amir Ali Ahmadi, Abraar Chaudhry, Vikas Sindhwani, and Stephen Tu. Safely learning dynamical systems from short trajectories. In *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, volume 144 of *Proceedings of Machine Learning Research*, pages 498–509. PMLR, 07 – 08 June 2021.

Anayo K. Akametalu, Jaime F. Fisac, Jeremy H. Gillula, Shahab Kaynama, Melanie N. Zeilinger, and Claire J. Tomlin. Reachability-based safe learning with Gaussian processes. In *53rd IEEE Conference on Decision and Control*, 2014.

Zvi Artstein and Saša V. Raković. Feedback and invariance under uncertainty via set-iterates. *Automatica*, 44(2):520–525, 2008.

Karl Johan Åström and Peter Eykhoff. System identification—a survey. *Automatica*, 7(2):123–162, 1971.

Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on Modern Convex Optimization*. Society for Industrial and Applied Mathematics, 2001.

Felix Berkenkamp and Angela P. Schoellig. Safe and robust learning control with Gaussian processes. In *2015 European Control Conference (ECC)*, 2015.

Felix Berkenkamp, Matteo Turchetta, Angela P. Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In *Neural Information Processing Systems*, 2017.

Homanga Bharadhwaj, Aviral Kumar, Nicholas Rhinehart, Sergey Levine, Florian Shkurti, and Animesh Garg. Conservative safety critics for exploration. In *International Conference on Learning Representations*, 2021.

Andrea Bisoffi, Claudio De Persis, and Pietro Tesi. Data-driven control via Petersen's lemma. *Automatica*, 145:110537, 2022. ISSN 0005-1098.

Grigoriy Blekherman, Pablo A. Parrilo, and Rekha Thomas. *Semidefinite Optimization and Convex Algebraic Geometry*. SIAM Series on Optimization, 2013.

Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear Matrix Inequalities In System And Control Theory*. SIAM, 1994.

Lukas Brunke, Melissa Greeff, Adam W. Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P. Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Reviews of Control, Robotics, and Autonomous Systems*, 2021.

Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.

Shaoru Chen, Nikolai Matni, Manfred Morari, and Victor M. Preciado. System level synthesis-based robust model predictive control through convex inner approximation. *arXiv:2111.05509*, 2021.

Shaoru Chen, Victor M. Preciado, Manfred Morari, and Nikolai Matni. Robust model predictive control with polytopic model uncertainty through system level synthesis. *arXiv:2203.11375*, 2022.

Richard Cheng, Gábor Orosz, Richard M. Murray, and Joel W. Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, 2019.

Yohann De Castro, Fabrice Gamboa, Didier Henrion, Roxana Hess, and Jean-Bernard Lasserre. Approximate optimal designs for multivariate polynomial regression. *Ann. Statist.*, 47(1):127–155, 2019.

Sarah Dean, Stephen Tu, Nikolai Matni, and Benjamin Recht. Safely learning to control the constrained linear quadratic regulator. In *Proceedings of the American Control Conference*, 2019.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, 20(4): 633–679, 2020. ISSN 1615-3383.

Daniel Ammon Dowler. Bounding the norm of matrix powers. In *BYU ScholarsArchive*, 2013.

Gerald B. Folland. *Real Analysis: Modern Techniques and Their Applications*. Wiley, 1999.

Meichen Guo, Claudio De Persis, and Pietro Tesi. Data-driven stabilization of nonlinear polynomial systems with noisy data. *IEEE Transactions on Automatic Control*, 67(8):4210–4217, 2022.

Thomas Gurriet, Mark Mote, Andrew Singletary, Eric Feron, and Aaron D. Ames. A scalable controlled set invariance framework with practical safety guarantees. In *Proceedings of the 58th IEEE Conference on Decision and Control*, 2019.

Karel J. Keesman. *System Identification: An Introduction*, volume 2. Springer, 2011.

Torsten Koller, Felix Berkenkamp, Matteo Turchetta, Joschka Boedecker, and Andreas Krause. Learning-based model predictive control for safe exploration and reinforcement learning. *arXiv:1906.12189*, 2019.

Jean Bernard Lasserre. *Moments, Positive Polynomials And Their Applications*, volume 1. World Scientific, 2010.

Monique Laurent. *Sums of Squares, Moment Matrices and Optimization Over Polynomials*, pages 157–270. Springer New York, New York, NY, 2009. ISBN 978-0-387-09686-5.

Miguel Sousa Lobo, Lieven Vandenberghe, Stephen Boyd, and Hervé Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284(1):193–228, 1998.

Tyler Lu, Martin Zinkevich, Craig Boutilier, Binz Roy, and Dale Schuurmans. Safe exploration for identifying linear systems via robust optimization. *arXiv:1711.11165*, 2017.

Wenhao Luo, Wen Sun, and Ashish Kapoor. No-regret safe learning for online nonlinear control with control barrier functions. In *ICRA Workshop on Safe Robot Control with Learned Motion and Environment Models*, 2021.

Alessandro Luppi, Andrea Bisoffi, Claudio De Persis, and Pietro Tesi. Data-driven design of safe control for polynomial systems. *arXiv:2112.12664*, 2021.

Tommaso Mannucci, Erik-Jan van Kampen, Cornelis de Visser, and Qiping Chu. Safe exploration algorithms for reinforcement learning controllers. *IEEE Transactions on Neural Networks and Learning Systems*, 29(4):1069–1081, 2018.

Pablo Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, 2000.

Friedrich Pukelsheim. *Optimal Design of Experiments*. Society for Industrial and Applied Mathematics, 2006.

Mihai Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42(3):969–984, 1993.

Imre Pólik and Tamás Terlaky. A survey of the S-lemma. *SIAM Review*, 49(3):371–418, 2007.

C. W. Scherer and C. W. J. Hol. Matrix sum-of-squares relaxations for robust semi-definite programs. *Math. Program.*, 107(1–2):189–211, 2006.

Andrew Taylor, Andrew Singletary, Yisong Yue, and Aaron Ames. Learning for safety-critical control with control barrier functions. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, 2020.

Kim Peter Wabersich and Melanie N. Zeilinger. A predictive safety filter for learning-based control of constrained nonlinear dynamical systems. *Automatica*, 129:109597, 2021.

Steffen W. R. Werner and Benjamin Peherstorfer. On the sample complexity of stabilizing linear dynamical systems from data. *Foundations of Computational Mathematics*, 2023. ISSN 1615-3383.

S.H. Żak. *Systems and Control*. Oxford Series in Electrical and Computer Engineering. Oxford University Press, 2003.

Jesse Zhang, Brian Cheung, Chelsea Finn, Sergey Levine, and Dinesh Jayaraman. Cautious adaptation for reinforcement learning in safety-critical settings. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2020.

Appendix A. (Proofs Omitted)

A.1. Proof of Lemma 6

Proof We form the desired basis $\{e_i\}$ iteratively and with an inductive argument. Let e_1 be any nonzero vector in P (existence of such a vector can be checked by the argument in the proof of Lemma 5); if there is no such vector, we return the empty set. Let $\{e_1, \ldots, e_k\}$ be a linearly independent set in P. We will either find an additional linearly independent vector $e_{k+1} \in P$, or show that the dimension of the span of P is k. Let x, x^+, and x^- be variables in $\mathbb{R}^n$, y^+ and y^- be variables in $\mathbb{R}^p$, and λ^+ and λ^- be variables in $\mathbb{R}$. Consider the following linear programming feasibility problem:

$$\begin{aligned}
&e_i^T x = 0 \quad i = 1, \ldots, k \\
&x = x^+ - x^- \\
&Ax^+ + By^+ \leq \lambda^+ c \\
&Ax^- + By^- \leq \lambda^- c \\
&\lambda^+ \geq 0 \\
&\lambda^- \geq 0.
\end{aligned} \tag{42}$$

Let $F \subseteq \mathbb{R}^n$ be the projection to x-space of the feasible region of this problem. We claim that $F = \{0\}$ if and only if the dimension of span(P) is k. Moreover, if there is solution to (42) with $x \neq 0$, then there is also a solution $(x, x^\pm, y^\pm, \lambda^\pm)$ where $\lambda^+, \lambda^- \neq 0$. In this case, either $\frac{x^+}{\lambda^+}$ or $\frac{x^+}{\lambda^+}$ can be taken as e_{k+1}.

Suppose first that the dimension span(P) is at least $k+1$; then there is a vector $\tilde{x} \in$ span(P) that is linearly independent from $\{e_1, \ldots, e_k\}$. By subtracting the projection of $\tilde{x}$ to span$(\{e_1, \ldots, e_k\})$, we will find a nonzero vector $x \in$ span(P) that is orthogonal to the vectors $e_1, \ldots, e_k$. We claim this vector x is feasible to (42) for some choice of $(x^\pm, y^\pm, \lambda^\pm)$. Indeed, since $x \in$ span(P), then

$$x = \sum_{j=1}^{r} \lambda_j x_j,$$

for some vectors $x_1, \ldots, x_r \in P$ and some nonzero scalars $\lambda_1, \ldots, \lambda_r$. For each j, as $x_j \in P$, there exists a vector y_j such that $Ax_j + By_j \leq c$. Let J denote the set of indices j such that $\lambda_j > 0$. It is easy to check that the assignment

$$\begin{aligned}
(x^+, y^+, \lambda^+) &= (\sum_{j \in J} \lambda_j x_j, \sum_{j \in J} \lambda_j y_j, \sum_{j \in J} \lambda_j)\,, \\
(x^-, y^-, \lambda^-) &= (-\sum_{j \notin J} \lambda_j x_j, -\sum_{j \notin J} \lambda_j y_j, -\sum_{j \notin J} \lambda_j)
\end{aligned} \tag{43}$$

satisfies system (42). Hence, we have shown that if $F = \{0\}$ then the dimension of span(P) is k.

To see the converse implication, suppose $x \neq 0$, and that the tuple $(x, x^\pm, y^\pm, \lambda^\pm)$ is feasible to system (42). Without loss of generality we assume $\lambda^\pm \geq 1$; if not, we replace the tuple with

$$(x, x^\pm + \hat{x}, y^\pm + \hat{y}, \lambda^\pm + 1), \tag{44}$$

where $\hat{x}$ and $\hat{y}$ are any vectors satisfying $A\hat{x} + B\hat{y} \leq c$. Then, since $A\frac{x^+}{\lambda^+} + B\frac{y^+}{\lambda^+} \leq c$, the vector $\frac{x^+}{\lambda^+} \in P$. By the same argument, $\frac{x^-}{\lambda^-} \in P$. It follows from the orthogonality constraint of (42) that at least one of the vectors $\frac{x^+}{\lambda^+}$ and $\frac{x^-}{\lambda^-}$ is linearly independent from $\{e_1, \ldots, e_k\}$ and can be taken as e_{k+1}, also proving that the dimension of span(P) is at least $k+1$.

Note that the condition $F = \{0\}$ can be checked by solving $2n$ linear programs (cf. the proof of Lemma 5); if $F \neq \{0\}$, then at least one of these $2n$ linear programs will return a tuple $(x, x^\pm, y^\pm, \lambda^\pm)$ where $x \neq 0$. We then transform this tuple via (44) to ensure that both $\lambda^+, \lambda^- \neq 0$ (we can take $\hat{x} = e_1$ and $\hat{y}$ to be any vector such that $Ae_1 + B\hat{y} \leq c$). Since we cannot have more than n linearly independent vectors in span(P), this procedure is repeated at most n times. ■

A.2. Proof of Lemma 11

Proof Let $e_i \in \mathbb{R}^n$ be the i-th canonical basis vector. We construct $2n$ points $\{x_1^\pm, \ldots, x_n^\pm\}$ using the following iterative procedure.

To construct $x_1^\pm$, we first solve (13) with $c = \pm e_1$ and set x_1^+, x_1^{-1} to be optimal solutions for $+e_1, -e_1$, respectively. Now to construct $x_{k+1}^\pm$ given $x_1^\pm, \ldots, x_k^\pm$, we solve (13) with $c = \pm e_{k+1}$ and with the additional constraints that $e_i^T x = \frac{e_i^T x_i^+ + e_i^T x_i^-}{2}$ for each $i = 1, \ldots, k$; call the resulting optimal points $x_{k+1}^\pm$. By Theorem 9, every $x_k^\pm$, for $k = 1, \ldots, n$ is the solution to a semidefinite program.

We now prove that $\mathrm{conv}(x_1^\pm, \ldots, x_n^\pm)$ is a full-dimensional subset of S_0^2. For a vector $x \in \mathbb{R}^n$ and a positive scalar r, let $B(x, r)$ represent the closed ℓ_2 ball centered at x of radius r. Let x_0 and r_0 be such that $B(x_0, r_0) \subseteq S_0^2$; such a point exists by the assumption that S_0^2 is full-dimensional.

We first show by induction that for each $k = 0, \ldots, n$, there exist some x_k and $r_k > 0$ such that $B(x_k, r_k) \subseteq S_0^2$ and $e_i^T x_k = \frac{e_i^T x_i^+ + e_i^T x_i^-}{2}$ for each $i = 1, \ldots, k$. The base case $k = 0$ holds by assumption. Assume for $k < n$ we have such an x_k and $r_k > 0$, and let us show the corresponding statement for $k+1$. By the properties assumed of x_k and r_k, and by the definition of $x_{k+1}^\pm$, we have $e_{k+1}^T x_{k+1}^+ \leq e_{k+1}^T x_k - r_k$ and $e_{k+1}^T x_{k+1}^- \geq e_{k+1}^T x_k + r_k$. Therefore, we have $e_{k+1}^T x_{k+1}^+ < e_{k+1}^T x_{k+1}^-$. Assume without loss of generality that $\frac{e_{k+1}^T x_{k+1}^+ + e_{k+1}^T x_{k+1}^-}{2} \leq e_{k+1}^T x_k$ (if the inequality is reversed, swap x_{k+1}^+ with x_{k+1}^-). Let $\lambda \in [0, 1)$ be such that

$$\frac{e_{k+1}^T x_{k+1}^+ + e_{k+1}^T x_{k+1}^-}{2} = \lambda e_{k+1}^T x_{k+1}^+ + (1-\lambda) e_{k+1}^T x_k.$$

Now we define $x_{k+1} := \lambda x_{k+1}^+ + (1-\lambda) x_k$ and $r_{k+1} = (1-\lambda) r_k$. It is clear by this definition that x_{k+1} satisfies the constraints $e_i^T x_{k+1} = \frac{e_i^T x_i^+ + e_i^T x_i^-}{2}$ for each $i = 1, \ldots, k$ since it is a convex combination of the vectors x_{k+1}^+ and x_k which also satisfy those constraints. It is also clear by the choice of λ that x_{k+1} satisfies $\frac{e_{k+1}^T x_{k+1}^+ + e_{k+1}^T x_{k+1}^-}{2} = e_{k+1}^T x_{k+1}$. Since S_0^2 is a convex set and since we have $x_{k+1}^+ \in S_0^2$ and $B(x_k, r_k) \subseteq S_0^2$, it follows that S_0^2 contains the convex hull of x_{k+1}^+ and $B(x_k, r_k)$. Observe that $B(x_{k+1}, r_{k+1})$ lies inside this convex hull and therefore also S_0^2, by the following Minkowski arithmetic:

$$\begin{aligned}
\mathrm{conv}(\{x_{k+1}^+\}, B(x_k, r_k)) &\supseteq \lambda x_{k+1}^+ + (1-\lambda) B(x_k, r_k) \\
&= \lambda x_{k+1}^+ + B((1-\lambda) x_k, (1-\lambda) r_k) \\
&= B(\lambda x_{k+1}^+ + (1-\lambda) x_k, (1-\lambda) r_k) \\
&= B(x_{k+1}, r_{k+1}).
\end{aligned}$$

This establishes the statement for $k+1$ and concludes the inductive argument.

Thus, for each $k = 0, \dots, n$, there exist some x_k and $r_k > 0$ such that $B(x_k, r_k) \subseteq S_0^2$ and $e_i^T x_k = \frac{e_i^T x_i^+ + e_i^T x_i^-}{2}$ for each $i = 1, \dots, k$. It now follows that $e_k^T x_k^+ < e_k^T x_k^-$ for each $k = 1, \dots, n$. Let T be the $n \times n$ matrix whose i-th column is $x_i^+ - x_i^-$. Then $T_{ii} = e_i^T x_i^+ - e_i^T x_i^- \neq 0$, and for $k > i$ we have $T_{ik} = 0$ since $e_i^T x_k^+ = e_i^T x_k^- = \frac{e_i^T x_i^+ + e_i^T x_i^-}{2}$. Therefore T is invertible, because it is a lower triangular with nonzero entries on its diagonal. Define

$$x_c := \sum_{i=1}^{n} \frac{1}{2n}(x_i^+ + x_i^-)$$

and the set

$$M := \{x_c + u \mid \|T^{-1}u\|_\infty \leq \frac{1}{2n}\}.$$

Observe that M is full-dimensional. To conclude the proof, we show that $M \subseteq \text{conv}(\{x_i^\pm\}_{i=1}^n)$. Indeed, for any $x_c + u \in M$,

$$\begin{aligned} x_c + u &= \sum_{i=1}^{n} \frac{1}{2n}(x_i^+ + x_i^-) + T \sum_{i=1}^{n} e_i e_i^T T^{-1} u \\ &= \sum_{i=1}^{n} (\frac{1}{2n} + e_i^T T^{-1} u) x_i^+ + (\frac{1}{2n} - e_i^T T^{-1} u) x_i^- \\ &\in \text{conv}(\{x_i^\pm\}_{i=1}^n). \end{aligned}$$

Note that if S_0^2 is not full-dimensional, the points $\{x_i^\pm\}_{i=1}^n$ supplied by this algorithm would satisfy $e_k^T x_k^+ = e_k^T x_k^-$ for at least one $k = 1, \dots, n$. ■

A.3. Proof of Proposition 13

Proof We proceed by induction. For the base case, let N_1 be an arbitrary λ^n null-set. We clearly have $\mathbb{P}(z_1 \in N_1) = \mathbb{P}(\delta_1 \in N_1) = 0$ by the absolute continuity of the law of δ_1 w.r.t. λ^n.

For the inductive hypothesis, let k be an integer satisfying $1 \leq k \leq m-1$. Suppose that for every λ^{nk} null-set N_k, we have $\mathbb{P}((z_1, \dots, z_k) \in N_k) = 0$. Now let N_{k+1} be an arbitrary $\lambda^{n(k+1)}$ null-set. For any $\bar{z}_{1:k} \in \mathbb{R}^{n \times k}$, define the slice $N_{k+1}(\bar{z}_{1:k}) = \{\bar{z}_{k+1} \in \mathbb{R}^n \mid (\bar{z}_{1:k}, \bar{z}_{k+1}) \in N_{k+1}\}$. Next, define the set:

$$N_{k+1}^0 = \{\bar{z}_{1:k} \in \mathbb{R}^{n \times k} \mid N_{k+1}(\bar{z}_{1:k}) \text{ is } \lambda^n\text{-measurable and } \lambda^n(N_{k+1}(\bar{z}_{1:k})) = 0\}.$$

By the Fubini-Tonelli theorem for complete measures (see e.g. Theorem 2.39 of Folland (1999)), N_{k+1}^0 is λ^{nk}-measurable and $\lambda^{nk}((N_{k+1}^0)^c) = 0$. Abbreviating $z_{1:k} = (z_1, \dots, z_k)$,

$$\begin{aligned}
\mathbb{P}((z_1, \dots, z_{k+1}) \in N_{k+1}) &= \mathbb{P}(\{(z_{1:k}, z_{k+1}) \in N_{k+1}\} \cap \{z_{1:k} \in N_{k+1}^0\}) \\
&\qquad + \mathbb{P}(\{(z_{1:k}, z_{k+1}) \in N_{k+1}\} \cap \{z_{1:k} \in (N_{k+1}^0)^c\}) \\
&\leq \mathbb{P}(\{(z_{1:k}, z_{k+1}) \in N_{k+1}\} \cap \{z_{1:k} \in N_{k+1}^0\}) + \mathbb{P}(z_{1:k} \in (N_{k+1}^0)^c) \\
&\overset{(a)}{=} \mathbb{P}(\{(z_{1:k}, z_{k+1}) \in N_{k+1}\} \cap \{z_{1:k} \in N_{k+1}^0\}) \\
&= \mathbb{P}(\{z_{k+1} \in N_{k+1}(z_{1:k})\} \cap \{z_{1:k} \in N_{k+1}^0\}) \\
&= \mathbb{P}(\{\delta_{k+1} \in N_{k+1}(z_{1:k}) - f_k(z_{1:k})\} \cap \{z_{1:k} \in N_{k+1}^0\}) \\
&\overset{(b)}{=} 0.
\end{aligned}$$

Above, (a) follows by the inductive hypothesis and the fact that $(N_{k+1}^0)^c$ is a λ^{nk} null-set. Furthermore, (b) follows since when $z_{1:k} \in N_{k+1}^0$, then $N_{k+1}(z_{1:k})$ is a λ^n null-set, and hence by the translation invariance of λ^n, $N_{k+1}(z_{1:k}) - f_k(z_{1:k})$ is also a λ^n null-set. Therefore, by the absolute continuity of the law of δ_{k+1} w.r.t. λ^n and the independence of δ_{k+1} from $\delta_1, \dots, \delta_k$,

$$\mathbb{P}(\delta_{k+1} \in N_{k+1}(z_{1:k}) - f_k(z_{1:k}) \mid z_{1:k} \in N_{k+1}^0) = \mathbb{P}_{\delta_{k+1}}(\delta_{k+1} \in N_{k+1}(z_{1:k}) - f_k(z_{1:k})) = 0.$$

■

A.4. Proof of Proposition 14

Proof It is sufficient to show that for each integer k satisfying $k \geq n$,

$$\left[Ax = A_\star x, A^2 x = A_\star^2 x, \dots, A^n x = A_\star^n x\right] \Rightarrow A^k x = A_\star^k x.$$

Clearly this statement holds for $k = n$. We now assume the statement holds for some $k \geq n$ and show that it also holds for $k+1$. By the Cayley-Hamilton theorem, we have

$$A_\star^k \in \text{span}(I, \dots, A_\star^{n-1})$$

and from this it follows that

$$A_\star^k x \in \text{span}(x, \dots, A_\star^{n-1} x).$$

Therefore, there exist scalars λ_i, $i = 0, \dots, n-1$, such that $A_\star^k x = \sum_{i=0}^{n-1} \lambda_i A_\star^i x$. Now we have:

$$\begin{aligned}
A^{k+1} x &= A A^k x \overset{(a)}{=} A A_\star^k x \\
&= A\left(\sum_{i=0}^{n-1} \lambda_i A_\star^i x\right) = \sum_{i=0}^{n-1} \lambda_i A A_\star^i x \overset{(b)}{=} \sum_{i=0}^{n-1} \lambda_i A^{i+1} x \overset{(c)}{=} \sum_{i=0}^{n-1} \lambda_i A_\star^{i+1} x \\
&= A_\star\left(\sum_{i=0}^{n-1} \lambda_i A_\star^i x\right) = A_\star A_\star^k x = A_\star^{k+1} x,
\end{aligned}$$

where (a) follows from the inductive hypothesis and (b) and (c) follow from the assumption that $A^i x = A_\star^i x$ for $i = 1, \dots, n$. ■