

Learning Barrier-Certified Polynomial Dynamical Systems for Obstacle Avoidance with Robots

Martin Schonger^{1,†}, Hugo T. M. Kussaba^{1,†}, Lingyun Chen¹, Luis Figueredo²,
 Abdalla Swikir¹, Aude Billard³, and Sami Haddadin¹

Abstract—Established techniques that enable robots to learn from demonstrations are based on learning a stable dynamical system (DS). To increase the robots’ resilience to perturbations during tasks that involve static obstacle avoidance, we propose incorporating barrier certificates into an optimization problem to learn a stable and barrier-certified DS. Such optimization problem can be very complex or extremely conservative when the traditional linear parameter-varying formulation is used. Thus, different from previous approaches in the literature, we propose to use polynomial representations for DSs, which yields an optimization problem that can be tackled by sum-of-squares techniques. Finally, our approach can handle obstacle shapes that fall outside the scope of assumptions typically found in the literature concerning obstacle avoidance within the DS learning framework. Supplementary material can be found at the project webpage: <https://martinschonger.github.io/abc-ds>

I. INTRODUCTION

The future of robotics will be challenged with novel and complex tasks, involving close proximity to people. In particular, it will be essential to attain two key objectives. Firstly, since programming such complex tasks requires a high level of expertise and time, enabling the end-user to teach tasks along with preferences of how to perform those tasks is critical. Secondly, robots should be able to navigate around obstacles and avoid collisions with objects to perform their tasks efficiently and safely.

One way to accomplish the first objective is by the *Learning from Demonstration* (LfD) paradigm, that enables robots to learn a task by generalizing from demonstrations, rather than simply recording and replaying [1]. The fundamental principle of this paradigm is that it allows end-users to teach robots how to perform new tasks by providing them with examples of a task being performed, eliminating the need for coding on their part. An established method to implement

This work was funded by the European Union’s Horizon 2020 program (grant agreement No. 101070596 - euROBIN and Marie Skłodowska-Curie grant agreement no. 899987), by the German Research Foundation (DFG) as part of Germany’s Excellence Strategy, EXC 2050/1, Project ID 390696704 – Cluster of Excellence “Centre for Tactile Internet with Human-in-the-Loop” (CeTI) of Technische Universität Dresden, and by the Bavarian State Ministry for Economic Affairs, Regional Development and Energy (StMWi) (the Lighthouse Initiative KI.FABRIK Bayern Phase 1: Aufbau Infrastruktur). The authors would also like to thank Katharina Bieker, Philipp Scholl, and Bachir El Khadir for insightful discussions

¹Munich Institute of Robotics and Machine Intelligence (MIRMI), Technical University of Munich (TUM), Germany. Abdalla Swikir is also with Omar Al-Mukhtar University (OMU), Albaida, Libya.

²School of Computer Science, University of Nottingham, UK. Luis Figueredo is also an Associated Fellow at the MIRMI, TUM.

³Learning Algorithms and Systems Laboratory, EPFL, Switzerland.

[†]The first two authors contributed equally to this work.

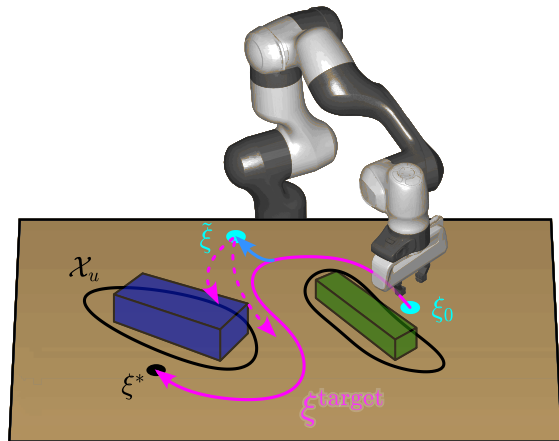


Fig. 1. In a pick-and-place task a robot can be taught how to move from ξ_0 to ξ^* by encoding this trajectory in a dynamical system (DS). While the reference trajectory avoids the two obstacles, enclosed by the unsafe set \mathcal{X}_u , a disturbance (blue) can push the robot to a state $\tilde{\xi}$ where there is no reference velocity available. Nonetheless, it is crucial that the trajectory starting in $\tilde{\xi}$ also does not collide with any obstacle, i.e. does not enter \mathcal{X}_u . For example, the left dashed partial trajectory is unsafe, whereas the right one is safe (up to where it is shown). It is desired from the DS to generate safe trajectories for regions of the state space that go beyond the reference trajectories.

LfD in robots is by encoding demonstrations in a stable *dynamical system* (DS) [2]. More precisely, the position and velocity of the end effector (or of the joints) are recorded for each demonstration, and optimization algorithms are used to find a globally stable dynamical system which reproduces these demonstrations as closely as possible.

Concurrently, as introduced by the second objective, it is important that robots learn how to realize these demonstrated tasks while also avoiding obstacles. Encoding demonstrations in a globally stable dynamical system aims to accurately reproduce these for the robot to follow while ensuring that it will still converge towards the desired equilibrium point even if perturbations occur along the trajectory [3]–[5]. Note that the trajectory followed after a perturbation is not necessarily close to the reference trajectories. This becomes particularly problematic if this new trajectory encounters an obstacle (as illustrated in Fig. 1). While methods have been proposed to make solutions tend toward the reference trajectory in certain parts of the state space [6], these do not take into account obstacles directly in the learning formulation.

Therefore, when dealing with scenarios involving obstacles, additional methods or techniques should be considered to address the challenges arising from perturbations and to ensure safe obstacle avoidance during the robot’s motion.

Considering this, [7]–[10] proposed techniques to modulate DSs to deal with dynamic obstacles. These techniques are designed to achieve real-time performance. However, they involve altering the original DS, which can potentially result in deviations from the user’s initial demonstration. Many scenarios, however, only require dealing with static obstacles—which allows using offline optimization techniques that *simultaneously* learn a DS encoding the user’s demonstrations, and a certificate that ensures that a specific set of trajectories of this DS will avoid user-defined obstacles. Since the certified DS results from a single-step optimization problem, it is expected that this DS will better preserve the user’s demonstrations compared to two-step methods that first optimize the DS for the user demonstrations and then modify it for avoiding a static obstacle, without taking into account the cost function of the initial step.

One method for incorporating obstacle avoidance as a constraint in the DS optimization problem is by utilizing barrier certificates [11]. However, naively combining barrier certificates with the usual Linear Parameter-Varying DS (LPV-DS) formulation [3], [6], [12] results in complex optimization problems, even when dealing with simple obstacles. Thus, instead of using the typical LPV-DS formulation, we propose to use a polynomial representation for DSs: this approach yields an optimization problem that can be tackled by sum-of-squares (SOS) techniques, and also facilitates the modelling of complex obstacles. In particular, this approach enables dealing with obstacles that violate the necessary assumptions of all algorithms previously proposed in the literature on obstacle-avoidance within the DS framework. For instance, [9] assumes that individual obstacles must be star-shaped, while the more recent work [10] requires that obstacles do not have any holes. Our method, on the other hand, only requires a basic semi-algebraic description of the obstacles (which can be systematically generated [13]). Notably, our method can be used with obstacles that are non-star-shaped (e.g. Fig. 6) and even with shapes that have holes.

In summary, in this paper we propose using polynomial dynamical systems as a way to combine and synergize the methodology of learning stable dynamical systems with barrier certificates. We call our algorithm *ABC-DS*, which stands for *obstacle Avoidance with Barrier-Certified polynomial Dynamical Systems*, and to demonstrate its effectiveness, we conducted numerical simulations and experiments. Finally, the source code for this research is publicly accessible.¹

II. PRELIMINARIES

We use the following notation: lowercase bold letters represent vectors in \mathbb{R}^n , e.g. \mathbf{x} , uppercase bold letters represent matrices in $\mathbb{R}^{n \times m}$, e.g. \mathbf{A} , and $\|\mathbf{x}\|$ denotes the 2-norm of \mathbf{x} . Given a differentiable function $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$, ∇f denotes the gradient of f , i.e. $\nabla f(\mathbf{x}) = (\partial f / \partial x_1(\mathbf{x}), \dots, \partial f / \partial x_n(\mathbf{x}))^\top$. Given a polynomial f , $\deg(f)$ denotes the degree of f . Lastly, $f^{-1}(0)$

denotes the 0-level set of a function $f: \mathbb{R}^n \rightarrow \mathbb{R}$, that is, $f^{-1}(0) = \{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) = 0\}$.

A. Learning stable dynamical systems

Learning a DS with a globally asymptotically stable equilibrium point in $\xi^* \in \mathbb{R}^n$ can be expressed as an optimization problem whose variables are the dynamics² $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ and a radially unbounded Lyapunov function $V: \mathbb{R}^n \rightarrow \mathbb{R}$. More precisely, given reference data in the form of N tuples of position and velocity, $(\xi_{\text{ref}}^{(i)}, \dot{\xi}_{\text{ref}}^{(i)})$, $i = 1, \dots, N$, one needs to solve the following optimization problem (cf. [3]):³

$$\min_f \sum_{i=1}^N \|\dot{\xi}_{\text{ref}}^{(i)} - f(\xi_{\text{ref}}^{(i)})\|^2 \quad (1a)$$

$$\text{s.t. } f(\mathbf{0}) = 0, \quad (1b)$$

$$V(\mathbf{0}) = 0, \quad V(\xi) > 0 \text{ for all } \xi \in \mathbb{R}^n \setminus \{\mathbf{0}\}, \quad (1c)$$

$$\dot{V}(\mathbf{0}) = 0, \quad \dot{V}(\xi) < 0 \text{ for all } \xi \in \mathbb{R}^n \setminus \{\mathbf{0}\}. \quad (1d)$$

Finding $f(\xi)$ and $V(\xi)$ for each $\xi \in \mathbb{R}^n$ is an infinite-dimensional optimization problem and, as such, not computationally tractable unless a suitable parameterization of f and V is chosen such that (1) becomes a finite-dimensional approximation. In particular, previous approaches in the literature propose *linear parameter-varying* (LPV) dynamics f :

$$f(\xi) = \sum_{k=1}^K \gamma_k(\xi) \mathbf{A}_k \xi, \quad (2)$$

where $\mathbf{A}_k \in \mathbb{R}^{n \times n}$ and $\gamma_k: \mathbb{R}^n \rightarrow \mathbb{R}$ is the K -component Gaussian mixture model (GMM) defined as

$$\gamma_k(\mathbf{x}) := \frac{\pi_k p(\mathbf{x}|k)}{\sum_j \pi_j p(\mathbf{x}|j)}, \quad (3)$$

with $\pi_k \geq 0$ being the mixing weights satisfying $\sum_{k=1}^K \pi_k = 1$, and $p(\mathbf{x}|k)$, $k = 1, \dots, K$, being the probability density function of a multivariate normal distribution. The DS corresponding to (2) is given by

$$\dot{\xi}(t) = \sum_{k=1}^K \gamma_k(\xi) \mathbf{A}_k \xi, \quad (4)$$

and the stability of the origin of (4) is certified by either (i) using the Lyapunov function $V(\xi) = \|\xi\|^2$, as in [3]; or (ii) searching for a generic quadratic Lyapunov function, i.e. $V(\xi) = \xi^\top \mathbf{P} \xi$, where $\mathbf{P} \in \mathbb{R}^{n \times n}$ is positive-definite, as in [12].

B. Barrier certificates

Barrier certificates were introduced in [11] with the purpose of verifying the safety of control systems. Given an *initial set* $\mathcal{X}_0 \subset \mathbb{R}^n$ and an *unsafe set* $\mathcal{X}_u \subset \mathbb{R}^n$, the DS is *safe* if $\xi_0 \in \mathcal{X}_0$ implies that the solution of the initial value

²Hereafter we assume that $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is such that the DS $\dot{\xi} = f(\xi)$ has a unique solution for each initial condition $\xi(0) \in \mathbb{R}^n$.

³We assume that the sequence $(\xi_{\text{ref}}^{(i)})_{i=1}^N$ is made of distinct elements, and that $(\xi_{\text{ref}}^{(N)}, \dot{\xi}_{\text{ref}}^{(N)}) = (\xi^*, \mathbf{0})$. Moreover, hereafter it is assumed that $\xi^* = \mathbf{0}$ without loss of generality since one can use the reference data $(\xi_{\text{ref}}^{(i)} - \xi^*, \dot{\xi}_{\text{ref}}^{(i)})_{i=1}^N$ instead of the original data, and translate the obtained DS in such way that its equilibrium point is ξ^* .

¹Available at <https://github.com/martinschonger/abc-ds>

problem $\dot{\tilde{\xi}}(t) = f(\tilde{\xi}(t))$ with $\tilde{\xi}(0) = \xi_0$ satisfies $\tilde{\xi}(t) \notin \mathcal{X}_u$ for all $t \geq 0$ (see Fig. 2).

To verify if a given system is safe, one can compute a barrier certificate. Similar to Lyapunov functions, barrier certificates do not necessitate the explicit computation of system trajectories, offering a convenient way of integrating safety constraints as additional constraints in optimization problems. The next theorem is adapted from [14] and shows how to employ barrier certificates:⁴

Theorem 1. Let $\mathcal{X}_0, \mathcal{X}_u \subset \mathbb{R}^n$, and $\varepsilon_1 > 0$. If there exists a continuously differentiable function $B: \mathbb{R}^n \rightarrow \mathbb{R}$ such that

$$B(\xi) \leq 0, \quad \text{for all } \xi \in \mathcal{X}_0, \quad (5a)$$

$$B(\xi) > 0, \quad \text{for all } \xi \in \mathcal{X}_u, \quad (5b)$$

$$\nabla B(\xi)^\top f(\xi) \leq 0, \quad \text{for all } \xi \text{ with } |B(\xi)| \leq \sqrt{\varepsilon_1}, \quad (5c)$$

then for all trajectories $\xi(t)$ of the system $\dot{\xi}(t) = f(\xi(t))$ such that $\xi(0) \in \mathcal{X}_0$, one has that $\xi(t) \notin \mathcal{X}_u$ for all $t \geq 0$.

Proof. Suppose $\xi_0 \in \mathcal{X}_0$ and there exists B satisfying (5). If the DS is not safe, there exist a trajectory $\tilde{\xi}: [0, +\infty) \rightarrow \mathbb{R}^n$ starting at ξ_0 and $t' > 0$ such that $B(\tilde{\xi}(t')) > 0$ by (5b). By (5a), $B(\tilde{\xi}(0)) \leq 0$. Since the function $t \mapsto B(\tilde{\xi}(t))$ is continuous, it follows from the intermediate value theorem that there exists t_1 with $0 \leq t_1 < t'$ and $-\sqrt{\varepsilon_1} \leq B(\tilde{\xi}(t_1)) \leq 0$, and exists t_2 with $t_1 < t_2 \leq t'$ and $0 < B(\tilde{\xi}(t_2)) \leq \sqrt{\varepsilon_1}$. Since the function $t \mapsto B(\tilde{\xi}(t))$ is differentiable, the mean value theorem assures there exists a t'' in (t_1, t_2) such that $\frac{d}{dt}[B(\tilde{\xi}(t))]$ evaluated at t'' is strictly positive. But this contradicts (5c). ■

A function B satisfying conditions (5a)–(5c) of Theorem 1 is called a *barrier certificate* for f with respect to \mathcal{X}_0 and \mathcal{X}_u . While this theorem only makes a direct statement regarding safety of trajectories starting in \mathcal{X}_0 , the safe region of the state space actually amounts to all points ξ where $B(\xi) \leq 0$ and can be much larger than \mathcal{X}_0 (see Fig. 2). We define the *certified safe set* as $\mathcal{X}_s := \{\xi \in \mathbb{R}^n : B(\xi) \leq 0\}$. By (5b) and (5c), any trajectory starting in \mathcal{X}_s will never cross the level set $B^{-1}(0)$ and, therefore, never reach \mathcal{X}_u . The reason for taking \mathcal{X}_0 as a strict subset of \mathcal{X}_s is to increase the search space of the optimization problem.

While Theorem 1 is very flexible and gives many options for searching a barrier certificate B , it is desired to restrict B to a set of functions that turns conditions (5a)–(5c) of Theorem 1 into a computationally tractable optimization problem. A common approach for achieving this is by the use of sum-of-squares (SOS) optimization [11].

III. PROPOSED METHOD

The optimization framework (1) offers the flexibility of combining the constraints presented in (1b)–(1d) and (5)

⁴Instead of checking a strict inequality for the term \dot{B} over the 0-level set of B as in Proposition 3 of [14], we propose to check a non-strict inequality over a larger set containing the 0-level set of B . While Theorem 1 of the previous work [11] proposed checking this non-strict inequality only over the 0-level set of B , a variation of the counterexample in [15] shows that [11, Theorem 1] is incorrect.

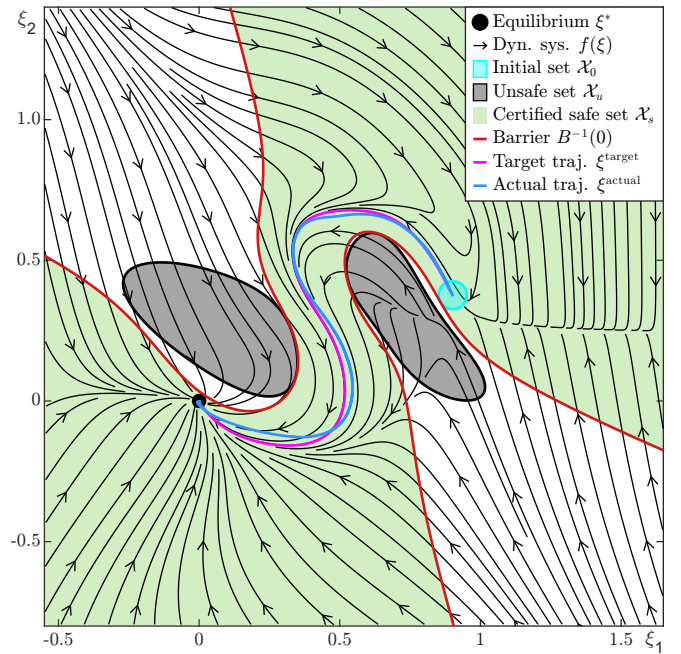


Fig. 2. A dynamical system f is safe if none of its trajectories starting from a state in the initial set \mathcal{X}_0 reach a state in the unsafe set \mathcal{X}_u . In the particular case of obstacle avoidance, \mathcal{X}_u can be specified to enclose any obstacles. The certified safe set \mathcal{X}_s of f with respect to a barrier certificate B is a superset of \mathcal{X}_0 and amounts to all states ξ for which $B(\xi) \leq 0$. When starting at any state in \mathcal{X}_s , the trajectory generated by f is guaranteed to not enter \mathcal{X}_u . The specific DS shown in the figure was generated by our proposed approach, with reference data obtained from a robot. The trajectory generated by the DS when starting in the center of \mathcal{X}_0 is plotted in pink. The trajectory executed by the robot when controlled by this DS is shown in blue.

into a single optimization problem minimizing the objective (1a). However, the computational feasibility of the resulting optimization problem depends on the f used to represent the DS. For instance, using an LPV dynamics such as in (2) leads to a highly complex optimization problem due to the exponential terms γ_k in (3).

Here, we propose to use a polynomial dynamics f for parameterizing DSs to enable the formulation of an optimization problem computationally tractable by sum-of-squares techniques. Simultaneously, this parameterization has the capability to capture complex dynamics: as proved by [16], trajectories of a continuously differentiable vector field that pass through pre-fixed points can be arbitrarily approximated by trajectories of polynomial vector fields that pass through the same points. Furthermore, experiments presented in [17], as well as those detailed in Section IV of our paper,⁵ demonstrate that when applied in the context of learning DSs, their performance is on par with LPV-DS.

We show now how to transform Theorem 1 into a SOS optimization problem. For that, we assume that the initial set \mathcal{X}_0 and the unsafe set \mathcal{X}_u are *basic semialgebraic sets*, that

⁵The results of [17] regarding polynomial DSs as an alternative to LPV-DS were developed concurrently with ours. However, the primary focus of this paper is using polynomial DSs as a way to leverage the use of barrier certificates, while [17] focused on the comparison of polynomial DSs with LPV-DS (without barrier certificates).

is, they can be described as

$$\mathcal{X}_0 = \{\boldsymbol{\xi} \in \mathbb{R}^n : \tilde{g}_1(\boldsymbol{\xi}) \geq 0, \dots, \tilde{g}_p(\boldsymbol{\xi}) \geq 0\}, \quad (6a)$$

$$\mathcal{X}_u = \{\boldsymbol{\xi} \in \mathbb{R}^n : g_1(\boldsymbol{\xi}) \geq 0, \dots, g_m(\boldsymbol{\xi}) \geq 0\}, \quad (6b)$$

where $\tilde{g}_i: \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, p$, and $g_j: \mathbb{R}^n \rightarrow \mathbb{R}$, $j = 1, \dots, m$, are polynomials. It is important to emphasize that even if the initial and unsafe sets are non-semialgebraic, one can use lightweight algorithms (e.g. [13]) to generate basic semialgebraic sets that overapproximate the non-semialgebraic sets. Thus, the SOS optimization in the next proposition can handle initial and unsafe sets with very complicated shapes, possibly non-convex and non-connected.

Proposition 1. *If there exist a polynomial function $B: \mathbb{R}^n \rightarrow \mathbb{R}$, SOS polynomials⁶ $\phi: \mathbb{R}^n \rightarrow \mathbb{R}$, $\tau_i: \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, p$, and $\sigma_j: \mathbb{R}^n \rightarrow \mathbb{R}$, $j = 1, \dots, m$, and $\varepsilon_1, \varepsilon_2 > 0$ satisfying the following conditions⁷*

$$-B(\boldsymbol{\xi}) - \sum_{i=1}^p \tau_i(\boldsymbol{\xi}) \tilde{g}_i(\boldsymbol{\xi}) \text{ is SOS}, \quad (7a)$$

$$B(\boldsymbol{\xi}) - \varepsilon_2 - \sum_{j=1}^m \sigma_j(\boldsymbol{\xi}) g_j(\boldsymbol{\xi}) \text{ is SOS}, \quad (7b)$$

$$-\nabla B(\boldsymbol{\xi})^\top f(\boldsymbol{\xi}) - \phi(\boldsymbol{\xi})(\varepsilon_1 - B^2(\boldsymbol{\xi})) - \varepsilon_2 \|\boldsymbol{\xi}\|^2 \text{ is SOS}, \quad (7c)$$

then B is a barrier certificate for the system f with respect to the initial set \mathcal{X}_0 and the unsafe set \mathcal{X}_u .

Proof. First, suppose that $\boldsymbol{\xi} \in \mathcal{X}_0$. Then $\tilde{g}_i(\boldsymbol{\xi}) \geq 0$ and this implies that $\sum_{i=1}^p \tau_i(\boldsymbol{\xi}) \tilde{g}_i(\boldsymbol{\xi}) \geq 0$. This last inequality and the non-negativity of (7a) imply that $B(\boldsymbol{\xi}) \leq 0$. Thus, $B(\boldsymbol{\xi}) \leq 0$ for all $\boldsymbol{\xi} \in \mathcal{X}_0$ and (5a) is satisfied. A similar argument can be applied to show that (7b) implies $B(\boldsymbol{\xi}) \geq \varepsilon_2 > 0$ for all $\boldsymbol{\xi} \in \mathcal{X}_u$, fulfilling (5b). Finally, if $|B(\boldsymbol{\xi})| \leq \sqrt{\varepsilon_1}$ then (7c) implies that $\nabla B(\boldsymbol{\xi})^\top f(\boldsymbol{\xi}) \leq -\varepsilon_2 \|\boldsymbol{\xi}\|^2 \leq 0$, satisfying (5c). ■

Based on Proposition 1, we propose *ABC-DS* in Algorithm 1. The constraints (7) represent the safety requirements, while the constraints (1b), (8b) and (8c) represent the global stability of the origin (for more details, see [19]). The sum-of-squares constraints can be easily recast as linear and bilinear⁸ matrix inequalities (LMI and BMI, respectively) by using specialized parsers [23]–[25]. The constraint (8d) is used to exclude local optima where the barrier certificate cuts through the reference trajectories and as such, to bias the solver toward a useful solution. It is worthy to note that this constraint is linear in the variables of B and, thus, adds only marginal complexity to the problem even for large N .

As the degree of the polynomial variables increases, the search space of the optimization problem grows, increasing

⁶A polynomial is SOS if it can be written as a sum of polynomials, each raised to an even power [18].

⁷The terms with ε_2 in (7) are introduced to improve the numerical stability of the optimization problem and/or to be able to deal with the strict inequalities using SOS.

⁸Bilinear matrix inequality constraints can be viewed as bilinear extensions of LMIs. Differently from the latter, the former is a non-convex problem. Nevertheless, there are specialized solvers for solving them [20], and many of these problems can be approached by bisection techniques [21], [22]. For instance, optimization problems with quadratic objectives and BMI constraints have been solved in [12] and [6] to learn DSs without obstacles.

Algorithm 1 ABC-DS

Input: Samples of demonstrated positions $\boldsymbol{\xi}_{\text{ref}}^{(i)}$ and corresponding velocities $\dot{\boldsymbol{\xi}}_{\text{ref}}^{(i)}$ from reference trajectory, basic semi-algebraic initial set (6a), unsafe set (6b), and positive scalars $\varepsilon_1, \varepsilon_2$ and ε_3 .

Output: Dynamics f , Lyapunov function V , and barrier B .

Search polynomial functions $f, V, B: \mathbb{R}^n \rightarrow \mathbb{R}^n$ and sum-of-squares polynomials $\phi, (\tau_i)_{i=1}^p, (\sigma_j)_{j=1}^m: \mathbb{R}^n \rightarrow \mathbb{R}$ satisfying:

$$\min_f \sum_{i=1}^N \|\dot{\boldsymbol{\xi}}_{\text{ref}}^{(i)} - f(\boldsymbol{\xi}_{\text{ref}}^{(i)})\|^2 \quad (8a)$$

$$\text{s.t. } V(\boldsymbol{\xi}) - \varepsilon_3 \|\boldsymbol{\xi}\|^2 \text{ is SOS}, \quad (8b)$$

$$-\nabla V(\boldsymbol{\xi})^\top f(\boldsymbol{\xi}) - \varepsilon_3 \|\boldsymbol{\xi}\|^2 \text{ is SOS}, \quad (8c)$$

$$\sum_{i=1}^N \max\{0, B(\boldsymbol{\xi}_{\text{ref}}^{(i)})\} \leq 0, \quad (8d)$$

and (1b) and (7) hold.

the chance of finding a solution. The price of increasing the degree is that it makes the problem more complex and, for very high degrees, the resulting problem could be numerically intractable for current solvers. Nevertheless, recent advances in the SOS literature have been enhancing and pushing forward the scalability of SOS-based optimization approaches [26]. It is also important to note that employing very high-degree polynomials in sum-of-squares optimization problems makes them more prone to numerical issues such as floating-point errors and ill-conditioned problem formulations, which can influence the trustworthiness of the final numerical results to varying degrees [27]. Nevertheless, post-analysis procedures can be performed to certify that the DSs generated by the optimization problem indeed satisfy the required properties (see e.g. [28], [29]). In this work, the DSs were verified using the SMT solvers cvc5 [30] and dReal [31].

IV. SIMULATIONS

In this section we illustrate the results of this paper by computing DSs for a representative subset of the LASA Handwriting Dataset⁹ [3]. We subsample the data to 100 samples per reference trajectory and, without loss of generality, we assume the attractor point to coincide with the origin. When using polynomials of higher degrees, the spread of coefficients of the decision variables in the optimization problem can reach values in the 10^6 -range and above, which may cause the solver to behave unfavorably. In order to reduce this spread, we normalize the reference trajectories and corresponding reference velocities. All the simulations are run with MATLAB R2023a, using the numerical solvers PENBMI [20] and MOSEK [32] to solve the optimization problems. First, we show the competitiveness of ABC-DS in encoding diverse motions with polynomial DSs. Then, we show the capability to produce DSs that are safe in the

⁹Available at <https://bitbucket.org/khansari/lasahandwritingdataset>

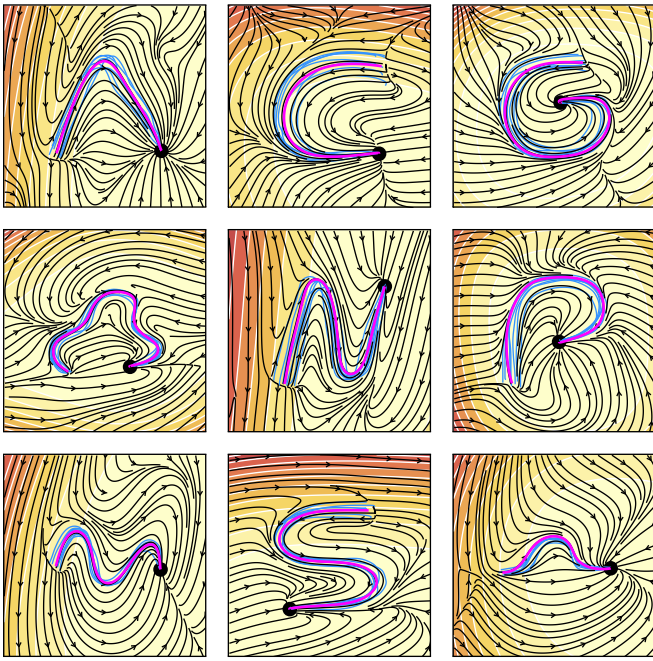


Fig. 3. Polynomial DS generated by our proposed method (without barrier) on a representative subset of the LASA handwriting dataset. Black stream lines depict the vector field with a globally asymptotically stable equilibrium (black dot). The yellow contours depict the values of the Lyapunov function, where lighter areas indicate lower values. The reference trajectories are shown in blue, and a trajectory starting from the references' mean initial point and simulated by the DS is plotted in pink.

presence of obstacles but still closely resemble the reference trajectories.

A. Comparison of polynomial DSs with LPV-DS

Using ABC-DS without the barrier-related constraints, i.e. without (7) and (8d), we are able to generate polynomial DSs that accurately encode many demonstrations of the LASA dataset, including highly nonlinear ones like *Leaf2* (see Fig. 3). All shown results are obtained with the same optimization parameters and solver settings; in particular, $\deg(f) = 6$, $\deg(V) = 4$. We also warm-start some variables of the optimization problem of Algorithm 1 with the solution of the convex optimization problem obtained by enforcing $\deg(f) = 1$ and $V(\xi) = \xi^T \xi$.

To quantify the overall similarity of a dynamical system to its underlying reference trajectories, we report the mean squared error (MSE) between the reference velocities $\dot{\xi}^{\text{ref}}$ and the corresponding velocities output by the DS, $f(\xi^{\text{ref}})$. Fig. 4 gives a statistical overview of the MSE of our method on the scenarios from Fig. 3. It also conveys the competitive performance of our method compared to LPV-DS.¹⁰

Nevertheless, one current limitation of our approach is that for certain datasets the solver may have difficulty finding feasible solutions. However, in several cases a more extensive hyperparameter search on the solver settings or the optimization parameters along with heuristics for initializing the variables can solve this issue.

¹⁰We use the LPV-DS code from <https://github.com/nbfigueroa/ds-opt> with default GMM fitting from the provided demo script.

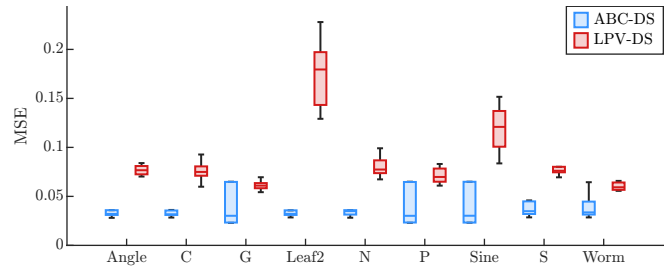


Fig. 4. Mean and standard deviation of the MSE of our polynomial approach (without barrier) versus LPV-DS on a subset of the LASA handwriting dataset. We evaluate each one 10 times with different random seeds for the random number generation, which influences the solver. The reported values are for normalized reference data.

B. Encoding demonstrations using ABC-DS

First, we select four motions from the LASA handwriting dataset and place an obstacle (in the shape of an ellipse) near the reference trajectories in each one of these scenarios. For simplicity, we define any \mathcal{X}_0 as a hypersphere large enough to enclose all respective reference trajectories' starting points. The solutions computed by our approach are plotted in Fig. 5. Note that we use slightly different optimization parameters for each scenario, with $\deg(f) \in \{4, 5\}$, $\deg(V) \in \{2, 4\}$, $\deg(B) \in \{3, 4\}$, and the solver settings differing mostly in the number of iterations. It can be seen that the trajectories generated by our DS when starting at the reference trajectories' initial points follow the reference trajectories very closely. At the same time, the computed barrier certificates define a non-conservative safe region: in practice, the certified safe sets (as introduced in Section II) are not limited to \mathcal{X}_0 , but typically encompass a large region of the state space.

Second, we demonstrate the capability of ABC-DS to work with complex concave obstacles (see Fig. 6). Specifically, we select a scenario where the reference trajectories enter the convex hull of such obstacle, because then the concavity cannot simply be mitigated by replacing the obstacle by its convex hull. Therefore, the state-of-the-art method [9] cannot be used. Also, ABC-DS generates a DS that reproduces the demonstrations faithfully. It is important to remark that the barrier's 0-level set is fit tightly between demonstrations and obstacle.

V. EXPERIMENTS

For this experiment,¹¹ the goal is to learn a non-trivial S-shaped motion in the presence of two static box-like obstacles, as illustrated in Figs. 1 and 7. The motion passes through a narrow passage formed by these obstacles. A video demonstration is available at <https://youtu.be/iei-hVL-pfc>.

Recording of reference trajectories and computing the DS: To obtain the reference trajectories, we use Cartesian impedance control with stiffness in X and Y axis set to 0, while stiffness in the Z axis is set to 500 N/m. This allows us to move the robot around in the X-Y plane by hand

¹¹Experiments are conducted with a 7-DoF Franka Emika robot [33]. The robot is controlled using the Franka Control Interface (FCI) at 1 kHz. The control loop runs on a computer (Intel Core i7-10700 @ 2.90GHz) installed with Ubuntu 18.04 LTS and real-time kernel (5.4.138-rt62).

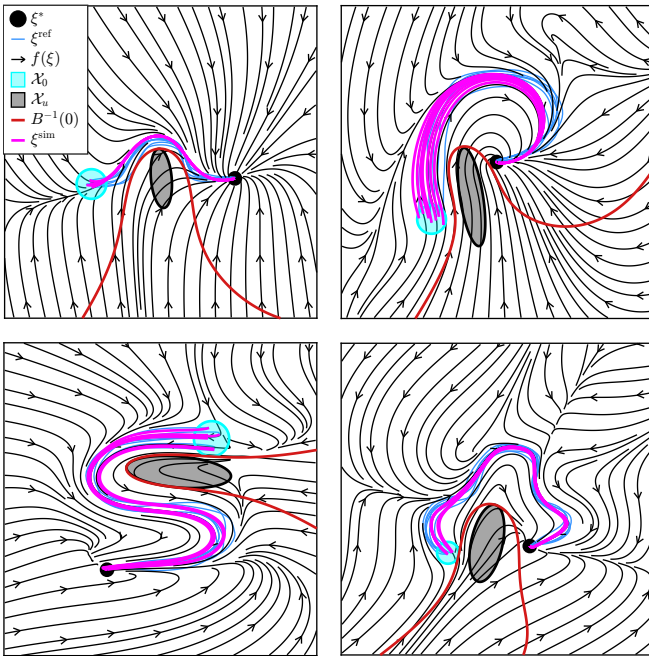


Fig. 5. Performance of ABC-DS on different data from the LASA dataset in the presence of elliptical unsafe sets. The obstacles are safely avoided with non-conservative barrier certificates and the trajectories simulated with the computed DS (black stream lines) are still close to the reference trajectories.

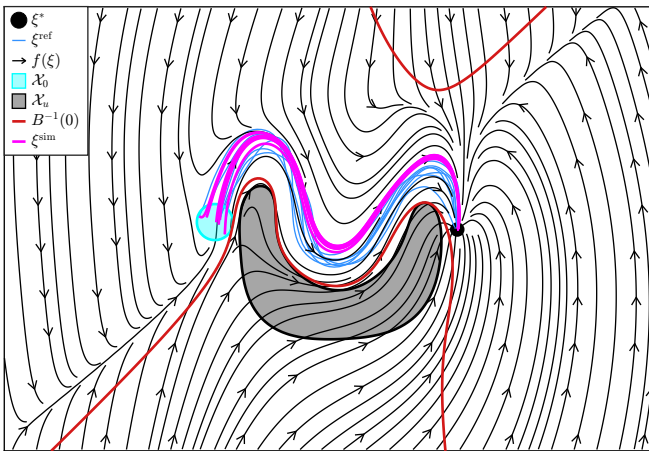


Fig. 6. ABC-DS works with non-star shaped obstacles (black). Here we exemplify the performance in a scenario where the reference trajectories penetrate the convex hull of such obstacle. The solution is obtained by setting $\deg(f) = 5$, $\deg(V) = 2$, and $\deg(B) = 4$.

and thereby guide it along the desired path. Utilizing the joints' position sensors and forward kinematics, we record 14 similar trajectories. For each, the robot is initially positioned randomly within a small circular area. The trajectories are subsampled to 100 points each. The obstacles' corner points in the plane are measured by hand. We obtain a basic semialgebraic set description for the unsafe set by first enlarging the polygons—to roughly account for the gripper dimensions—and then fitting a polynomial. First, we initialize certain variables in the optimization problem presented in Algorithm 1 using the solution derived from the convex optimization problem where we enforce $\deg(f) = 1$ and $V(\xi) = \xi^T \xi$. Then, using $\deg(f) = 5$, $\deg(V) = 2$,

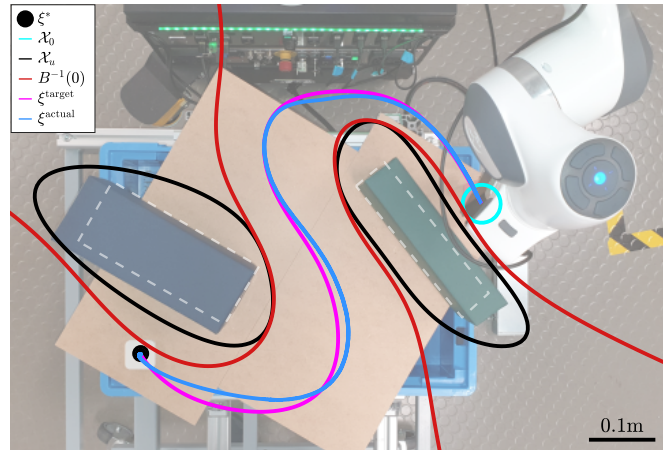


Fig. 7. The robot is shown at a position in the initial set (cyan), about to move along in an S-shaped motion towards the globally asymptotically stable equilibrium (black point). The dark green and dark blue boxes are the obstacles (outlines on the plane are highlighted in white). Overlaid is the derived unsafe set (black). The computed barrier certificate's 0-level set is plotted in red. The pink trajectory is obtained by integrating the DS from the initial position; and blue shows the robot's actual path when controlled by the DS, which closely follows the former.

$\deg(B) = 4$, and \mathcal{X}_0 as a hypersphere enclosing all reference trajectories' starting points, our ABC-DS algorithm is able to find a good feasible solution, as plotted in Fig. 2. Similar to scenarios with a single obstacle, the certified safe set \mathcal{X}_s encompasses a non-conservative portion of the state space.

Control with DS: To control the robot with the computed DS, we deploy Cartesian impedance control (for X and Y axis) with forward integration. Rudimentary tuning of a few parameters such as the robot impedance is performed. From Figs. 2 and 7 it can be seen that our DS does indeed keep the robot close to the desired trajectory and also respects the obstacles, thereby demonstrating the effectiveness of ABC-DS in practice.

Recovery after perturbations: The final experiment combines compliant behavior, handling of perturbations, obstacle avoidance, and maintaining similarity to a simulated target trajectory. Here, we deploy passive interaction control [4]. We demonstrate that the behavior during perturbations is as desired. Specifically, when pushed to anywhere in the safe region of the state space, i.e. where $B(\xi) < 0$, the robot correctly follows the underlying DS which (a) leads it to the globally stable equilibrium, and (b) guides it along paths that never cross the 0-level set of the barrier certificate.

VI. CONCLUSION

Our paper introduces ABC-DS, a novel approach that generates stable and safe DSs by solving a SOS optimization problem that simultaneously maximizes the encoding quality of user demonstrations in the DSs while also generating barrier certificates that guarantee obstacle avoidance in certified safe regions of the DSs. By using basic semialgebraic descriptions for obstacles, ABC-DS is able to deal with non star-shaped obstacles and obstacles with holes. In future work, we will investigate applying compositionality results, e.g. [34], [35], to our framework to further increase the scalability of our results.

REFERENCES

- [1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, *Robot Programming by Demonstration*, pp. 1371–1394. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.
- [2] A. Billard, S. Mirrazavi, and N. Figueroa, *Learning for Adaptive and Reactive Robot Control: A Dynamical Systems Approach*. MIT Press, 2022.
- [3] S. M. Khansari-Zadeh and A. Billard, “Learning stable nonlinear dynamical systems with Gaussian mixture models,” *IEEE Transactions on Robotics*, vol. 27, pp. 943–957, Oct. 2011.
- [4] K. Kronander and A. Billard, “Passive interaction control with dynamical systems,” *IEEE Robotics and Automation Letters*, vol. 1, pp. 106–113, Jan. 2016.
- [5] S. Gupta, A. Nayak, and A. Billard, “Learning high dimensional demonstrations using Laplacian eigenmaps,” *arXiv preprint arXiv:2207.08714*, 2022.
- [6] N. Figueroa and A. Billard, “Locally active globally stable dynamical systems: Theory, learning, and experiments,” *The International Journal of Robotics Research*, vol. 41, no. 3, pp. 312–347, 2022.
- [7] S. M. Khansari-Zadeh and A. Billard, “A dynamical system approach to real-time obstacle avoidance,” *Autonomous Robots*, vol. 32, pp. 433–454, 2012.
- [8] L. Huber, A. Billard, and J.-J. Slotine, “Avoidance of convex and concave obstacles with convergence ensured through contraction,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1462–1469, 2019.
- [9] L. Huber, J.-J. Slotine, and A. Billard, “Avoiding dense and dynamic obstacles in enclosed spaces: Application to moving in crowds,” *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 3113–3132, 2022.
- [10] L. Huber, J.-J. Slotine, and A. Billard, “Avoidance of concave obstacles through rotation of nonlinear dynamics,” *IEEE Transactions on Robotics*, 2023. Early Access.
- [11] S. Prajna and A. Jadbabaie, “Safety verification of hybrid systems using barrier certificates,” in *International Workshop on Hybrid Systems: Computation and Control*, pp. 477–492, Springer, 2004.
- [12] N. Figueroa and A. Billard, “A physically-consistent Bayesian non-parametric mixture model for dynamical system learning,” in *Conference on Robot Learning*, pp. 927–946, PMLR, 2018.
- [13] F. Dabbene, D. Henrion, and C. M. Lagoa, “Simple approximations of semialgebraic sets and their applications to control,” *Automatica*, vol. 78, pp. 110–118, 2017.
- [14] S. Prajna, A. Jadbabaie, and G. J. Pappas, “A framework for worst-case and stochastic safety verification using barrier certificates,” *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1415–1428, 2007.
- [15] Z. She, D. Song, and M. Li, “Safety verification of hybrid systems using certified multiple Lyapunov-like functions,” in *Computer Algebra in Scientific Computing: 17th International Workshop, CASC 2015, Aachen, Germany, September 14–18, 2015, Proceedings 17*, pp. 440–456, Springer, 2015.
- [16] A. A. Ahmadi and B. E. Khadir, “Learning dynamical systems with side information,” *SIAM Review*, vol. 65, no. 1, pp. 183–223, 2023.
- [17] A. Abyaneh and H.-C. Lin, “Learning Lyapunov-stable polynomial dynamical systems through imitation,” in *Proceedings of the 2023 Conference on Robot Learning (CoRL 2023)*, 2023.
- [18] P. A. Parrilo, “Semidefinite programming relaxations for semialgebraic problems,” *Mathematical Programming*, vol. 96, pp. 293–320, May 2003.
- [19] A. Papachristodoulou and S. Prajna, “A tutorial on sum of squares techniques for systems analysis,” in *Proceedings of the 2005, American Control Conference, 2005.*, pp. 2686–2700, IEEE, 2005.
- [20] M. Kočvara and M. Stingl, *PENBMI User’s Guide. Version 2.1*. PENOPT GbR, 2006.
- [21] M. Fukuda and M. Kojima, “Branch-and-cut algorithms for the bilinear matrix inequality eigenvalue problem,” *Computational Optimization and Applications*, vol. 19, pp. 79–105, 2001.
- [22] T. Cunis and B. Legat, “Sequential sum-of-squares programming for analysis of nonlinear systems,” in *Proceedings of the 2023 American Control Conference (ACC)*, pp. 756–772, IEEE, 2023.
- [23] J. Löfberg, “Pre- and post-processing sum-of-squares programs in practice,” *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 1007–1011, 2009.
- [24] A. Papachristodoulou, J. Anderson, G. Valmorbida, S. Prajna, P. Seiler, P. A. Parrilo, M. M. Peet, and D. Jagt, *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*. <http://arxiv.org/abs/1310.4716>, 2021. Available from <https://github.com/oxfordcontrol/SOSTOOLS>.
- [25] T. Weisser, B. Legat, C. Coey, L. Kapelevich, and J. P. Vielma, “Polynomial and moment optimization in Julia and JuMP,” in *JuliaCon*, 2019.
- [26] A. Majumdar, G. Hall, and A. A. Ahmadi, “Recent scalability improvements for semidefinite programming with applications in machine learning, control, and robotics,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 331–360, 2020.
- [27] P. Roux, Y.-L. Voronin, and S. Sankaranarayanan, “Validating numerical semidefinite programming solvers for polynomial invariants,” *Formal Methods in System Design*, vol. 53, pp. 286–312, 2018.
- [28] L. Dai, T. Gan, B. Xia, and N. Zhan, “Barrier certificates revisited,” *Journal of Symbolic Computation*, vol. 80, pp. 62–86, 2017.
- [29] S. Basagiannis, L. Battista, A. Becchi, A. Cimatti, G. Giantamidis, S. Mover, A. Tacchella, S. Tonetta, and V. Tsachouridis, “SMT-based stability verification of an industrial switched PI control systems,” in *2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pp. 243–250, IEEE, 2023.
- [30] H. Barbosa, C. Barrett, M. Brain, G. Kremer, H. Lachnitt, M. Mann, A. Mohamed, M. Mohamed, A. Niemetz, A. Nötzli, et al., “cvc5: A versatile and industrial-strength SMT solver,” in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pp. 415–442, Springer, 2022.
- [31] S. Gao, S. Kong, and E. M. Clarke, “dReal: An SMT solver for nonlinear theories over the reals,” in *Automated Deduction—CADE-24: 24th International Conference on Automated Deduction, Lake Placid, NY, USA, June 9–14, 2013. Proceedings 24*, pp. 208–214, Springer, 2013.
- [32] M. ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 10.0.*, 2023.
- [33] S. Haddadin, S. Parusel, L. Johannsmeier, S. Golz, S. Gabl, F. Walch, M. Sabaghian, C. Jaehne, L. Hausperger, and S. Haddadin, “The Franka Emika robot: A reference platform for robotics research and education,” *IEEE Robotics & Automation Magazine*, 2022.
- [34] P. Jagtap, A. Swikir, and M. Zamani, “Compositional construction of control barrier functions for interconnected control systems,” *HSCC ’20*, (New York, NY, USA), Association for Computing Machinery, 2020.
- [35] A. Swikir, *Compositional Synthesis of Symbolic Models for (In)Finite Networks of Cyber-Physical Systems*. PhD thesis, Technische Universität München, 2020.