TRUST: StabiliTy and Safety ContRoller Synthesis for Black-Box Systems Using a Single Trajectory

Jamie Gardner
School of Computing
Newcastle University
j.gardner3@newcastle.ac.uk

Ben Wooding
School of Computing
Newcastle University
ben.wooding@newcastle.ac.uk

Amy Nejati
School of Computing
Newcastle University
amy.nejati@newcastle.ac.uk

Abolfazl Lavaei
School of Computing
Newcastle University
abolfazl.lavaei@newcastle.ac.uk

ABSTRACT

TRUST is an open-source software tool for the single-trajectory data-driven control synthesis for Stability and safety barrier certificates (BCs). The tool implements sum-of-squares (SOS) optimizations for the systematic production of both linear and polynomial BCs. By ensuring the persistently-excited state of the input data, which is determined by fulfilling a certain rank condition, the system will compute the stability functions and safety barrier certificates and their respective controllers for an infinite time horizon from the finite-length dataset.

The stability and safety properties are calculated for four classes of dynamical systems: (i) continuous-time nonlinear polynomial systems, (ii) continuous-time linear systems, (iii) discrete-time nonlinear polynomial systems, and (iv) discrete-time linear systems. For each of these four scenarios, TRUST can synthesize the stability function and controller or the safety barrier certificate and controller.

TRUST is a Python-based web application, providing an intuitive and reactive GUI via web technologies, and can be accessed at <URL> or installed locally, and supports both manual data entry and data file uploads. The tool uses the SOS optimization toolbox to solve the stability and safety barrier problems, and the symbolic programming library SymPy to handle the symbolic computations. Leveraging the power of the Python backend and Javascript frontend, the tool is designed to be user-friendly and accessible on desktop, tablet, and mobile devices. Physical case studies validate our data-driven approach for each of the four pairs of scenarios.

KEYWORDS

Data-Driven Control Synthesis, Safety Verification, Barrier Certificates

ACM Reference Format:
Jamie Gardner, Ben Wooding, Amy Nejati, and Abolfazl Lavaei. 2025. TRUST: StabiliTy and Safety ContRoller Synthesis for Black-Box Systems Using a Single Trajectory. In ,. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

1.1 Motivation for TRUST

The design of control systems that ensure stability and safety is a fundamental problem in control theory, with applications in safety-critical systems including robotics, aerospace, self-driving cars, and medical devices, to name a few. Traditional model-based approaches to control synthesis rely on accurate mathematical models of the system dynamics, which are often difficult to obtain, or may not accurately reflect the real-world behavior of the system. In contrast, data-driven methods have gained popularity due to their ability to handle unknown mathematical models based on the data collected from the system [1].

A system is considered stable if it either remains at an equilibrium point or returns to it after a disturbance, known as asymptotic stability. In control synthesis, stability is crucial for ensuring that the system behaves as intended, and does not deviate from its desired behavior. This assurance comes from the calculation of a Lyapunov function for a system, which serves as a fundamental guarantee across the system's continuous-state space [2].

In addition to stability, safety barrier certificates formalise the notion of safety in dynamical systems, by ensuring that the system will never enter an unsafe region across an infinite time horizon. Control barrier certificates (CBCs) are particularly useful for real-world systems, where the data is directly measured through input-output observations, known as a “single trajectory” [3].

Willems' Fundamental Lemma provides a necessary condition for the existence of a stability function for continuous-time non-linear polynomial systems [4], thus enabling the design of safety barrier certificates for these systems. The computation of these safety barrier certificates can be performed using Sum of Squares (SOS) optimization techniques, using an existing toolbox [5]. It is worth noting that despite many works on the subject of *continuous-time* nonlinear polynomial systems [6–8], few works deal with the more difficult production of results for single-trajectory *discrete-time* non-linear polynomial systems [9]. Despite this lack of research,

HSCC'25, May 6-9 2025, Irvine, California, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

this work implements a cutting-edge theoretical discovery on data-driven methods for the design of stable and safe controllers for this category of system [10]. However, practical implementation of this research remains scarce, due to the complexity of real-world scenarios where the mathematical model of the system is unknown, or it is expensive in terms of manual effort to first identify and then solve the problem using model-based methods.

As a solution, data-driven techniques offer a promising reduction in cost and complexity, by leveraging

- Clarify the complexity e.g. the problems like "linear quadratic regulation and robust controller design" (rephrase and reuse the references); "model-reference controller design for linear dynamical systems" (same as previous); "stabilization of polynomial systems" (and again); and how these approaches don't account for state constraints (i.e. the level sets - clarify the relevance?);

- Then by clarifying the data requirements e.g. other scenario-based programs that require independent and identically distributed (i.i.d.) data which requires the collection of multiple trajectories (in the order of magnitude of millions)

Recent studies have proposed various data-driven methods for this problem (reference PRoTECT, IMPaCT and FOSSIL), which are much more realistic than traditional model-based approaches. However, these methods often require a large amount of data to provide accurate guarantees (reference), making them impractical for many real-world applications.

- Then highlight this gap that SinTra fills, i.e. synthesizing stability functions and safety barrier certificates using persistently-excited, single-trajectory data.

- Outline the benefits of this tool in terms of the formal guarantees it provides, including the breadth of scenarios for both stability and the extended problem of CBCs.

1.2 Related Literature

1.3 Related Software

- Discuss the existing tools in the area, including PRoTECT, IMPaCT, and FOSSIL, and how they differ from SinTra.

1.4 Contributions

This tool paper contributes the following novel and noteworthy developments:

i. TRUST is a first-of-its-kind tool that uses data-driven approaches to synthesize stability functions and safety barrier certificates from a single-trajectory.

ii. Implemented as a responsive and reactive Flask web application, TRUST provides an intuitive and user-friendly interface for users to interact with the tool.

iii. Real-time client-side updates and error management through asynchronous JavaScript and XML (AJAX) requests provide a seamless user experience across any platform.

iv. The server-side Python application leverages the Flask web framework to handle the data processing and computation of the stability functions and safety barrier certificates, following an MVC architecture and test-driven development for high-quality code.

v. TRUST extends the Sum of Squares (SOS) optimization toolbox powered by MOSEK to solve the stability and safety barrier problems for continuous-time and discrete-time linear and nonlinear polynomial systems using the symbolic programming library SymPy.

vi. The tool allows for both manual data entry and data file uploads for the X0, U0 and X1 datasets, a simple user experience for defining the state space, initial set, and multiple unsafe sets of the system, and, for nonlinear polynomial systems, the monomial terms.

vii. We demonstrate the usefulness of TRUST through eight distinct scenarios, representing the four classes of dynamical system, and their respective stability functions and safety barrier certificates.

1.5 Structure of the Paper

This paper is structured as follows:

(1) Introduction: provides the necessary background information on stability functions and safety barrier certificates, and the data-driven approach to control synthesis.

(2) Overview of the Tool: gives an overview of the TRUST tool, including the notation used, the inputs required, and the outputs produced.

(3) Problem Description: outlines the problem of stability and safety control synthesis for black-box systems using a single trajectory.

(4) Scenarios: presents the four primary scenarios for continuous-time and discrete-time linear and nonlinear polynomial systems, demonstrating the tool's capabilities.

(5) Benchmarks and Case Studies: provides a detailed analysis of the tool's performance through a series of benchmarks and case studies.

(6) Results: presents the results of the benchmarks and case studies, highlighting the tool's effectiveness in solving stability and safety problems.

2 OVERVIEW OF THE TOOL

2.1 Notation

$\mathbb{R}$, $\mathbb{R}_0^+$, and $\mathbb{R}^+$ denote the set of real numbers, non-negative and positive real numbers, respectively. $\mathbb{N}$ and $\mathbb{N}^+$ denote the set of natural numbers including and excluding zero. $\emptyset$ denotes the empty set. The notation $\mathbb{R}^{a \times b}$ in general is used to describe a matrix of size $a \times b$ that contains real values, equivalently $\mathbb{N}^{a \times b}$ for natural numbers. Similarly, a vector of size n is denoted with $\mathbb{R}^n$ or $\mathbb{N}^n$. A^{-1} and $A^\top$ are respectively the inverse and the transpose of matrix A. We use $[a, b]$ for the closed set between a and b.

2.2 Inputs to the tool

2.2.1 *Modes of the Tool.*

TRUST is valuable in solving two main types of problems:

(1) *Stability problems* - synthesizing a control Lyapunov function $\mathcal{V}(x)$ to show stability of the underlying system;

(2) *Safety barrier problems* - synthesizing a control barrier certificate $\mathcal{B}(x)$ to guarantee safe behavior of the underlying system.

In addition to these two modes that assume the presence of control inputs, TRUST can satisfy the same problems without control inputs, which are simpler cases for both modes. Therefore, all parameters relating to the controller or control inputs are optional.

2.2.2 Classes of Systems. TRUST can handle four main classes of systems; discrete-time linear systems (dt-LS), continuous-time linear systems (ct-LS), discrete-time nonlinear polynomial systems (dt-NPS) and continuous-time nonlinear polynomial systems (ct-NPS). It can handle each of these classes of systems for both the generation of stability functions and safety barrier certificates and their respective controllers.

2.2.3 Datasets (`*.csv`, `*.txt`, `*.json`). TRUST takes in as input the necessary datasets that together contain a *persistently excited* data trajectory, these are expected to be in one of the following common data file extensions; `*.csv`, `*.txt`, or `*.json`. The data should be sequential over a time horizon of $T \in \mathbb{N}^+$ steps, and may have multiple state dimensions $n \in \mathbb{N}^+$ and input dimensions $m \in \mathbb{N}$. In the discrete-time cases, the datasets can be represented as:

$$\begin{cases} \mathcal{X}_0 = [x(0), x(1), x(2), \dots, x(T-1)], \\ \mathcal{U}_0 = [u(0), u(1), u(2), \dots, u(T-1)], \\ \mathcal{X}_1 = [x(1), x(2), x(3), \dots, x(T)], \end{cases} \tag{1}$$

where $\mathcal{X}_0$ is generally the predecessor states, $\mathcal{X}_1$ is generally the successor states, and if there are control inputs they are captured by $\mathcal{U}_0$.

Remark 1. *We remark that the tool can handle $m = 0$ when there is no input present, the theoretical results remain largely the same [cite my k-inductive paper] with the key difference being no controller u is designed. In such a situation $\mathcal{U}_0$ is omitted completely.*

In a small abuse of notation, the continuous-time data is again sequential as:

$$\begin{cases} \mathcal{X}_0 = [x(0), x(\tau), x(2\tau), \dots, x((T-1)\tau)], \\ \mathcal{U}_0 = [u(0), u(\tau), u(2\tau), \dots, u((T-1)\tau)], \\ \mathcal{X}_1 = [\dot{x}(0), \dot{x}(\tau), \dot{x}(2\tau), \dots, \dot{x}((T-1)\tau)], \end{cases} \tag{2}$$

where τ is a fixed sampling time for the system. Additionally, $\mathcal{X}_1$ contains the derivatives of the state at each sampling time.

Remark 2. *In practice, these derivatives are likely to be approximated, any error handling relating to this approximation is left to the user.*

TRUST will check that the data satisfies a rank condition that ensures the data is persistantly excited. This rank condition is defined as [11]

$$\text{rank}\begin{bmatrix} \mathcal{U}_0 \\ \mathcal{X}_0 \end{bmatrix} = n + m. \tag{3}$$

2.2.4 Monomials (polynomials in x). For nonlinear polynomial systems, the user should provide a matrix of monomial terms $\mathcal{M}(x)$. As an example, of this when $x \in \mathbb{R}^2$: $\mathcal{M}(x) = \begin{bmatrix} x_1^2 & x_1x_2 & x_2 \end{bmatrix}^\top$ where here the number of monomial terms is $N = 3$ and $N \in \mathbb{N}^+$.

2.2.5 Definitions of initial set X_I , unsafe set X_U, state space X. For safety barrier problems, the user should define the state space X, initial set X_I and unsafe set X_U of the system using hypercubes for the *semi-algebraic sets* [12]. Since they are semi-algebraic, the hypercubes can be described as vectors of polynomials respectively $g_I(x), g_U(x)$ and $g(x)$. TRUST collects the lower bound and the upper bound for each dimension to construct the hypercube.

2.3 Outputs from Tool

The outputs of TRUST will depend on the mode chosen. For stability problems, TRUST will output a Lyapunov function $\mathcal{V}(x)$ based on the data or for safety barrier problems the safety barrier certificate $\mathcal{B}(x)$ and its level sets γ and λ. If either cannot be synthesized then TRUST will return some sort of error message.

In both modes the functions, $\mathcal{V}(x)$ or $\mathcal{B}(x)$, are quadratic of the form $x^\top Px$ in the majority of cases and $\mathcal{M}(x)^\top P\mathcal{M}(x)$ in the continuous-time nonlinear polynomial case, where $P \succ 0$ is a symmetric positive definite matrix. For problems that include data for control inputs $\mathcal{U}_0$, we will also return to the user the controller, $u \in \mathbb{R}^m$, synthesized for $\mathcal{V}(x)$ or $\mathcal{B}(x)$.

3 PROBLEM DESCRIPTION

- A few paragraphs that outline the problem to the reader - Specifically, detail the notations and definitions used (stability function, barrier function) - Concisely describe the tool in terms of applications and underlying algorithms (web-based UI, SOS toolbox, symbolic programming) - Pseudocode of the tool - Touch on the web stack at a high level as the entry point to the system - Focus more on the hierarchy of the problems, i.e. the simultaneous equations to solve for P and H, before the SOS conditions.

4 SCENARIOS

- Introduce the four systems, each with their safety function and stability barrier function alternatives. - Include the controller synthesis in each case - Add a marked-up screenshot of the tool, highlighting each region - three columns, first for data entry, second for state definitions, third for output - footer with the title and description, and the primary Calculate action

We remind the reader that the data considered in this section is of the form in equation (2).

4.1 Continuous Time Nonlinear Polynomial Systems

We consider continuous-time nonlinear polynomial systems (ct-NPS) defined as follows:

Definition 4.1. A ct-NPS is described by the ordinary differential equation

$$\Sigma_c : \dot{x} = A\mathcal{M}(x) + Bu \tag{4}$$

where $A \in \mathbb{R}^{n\times N}$ and $B \in \mathbb{R}^{n\times m}$ are unknown, $\mathcal{M}(x) \in \mathbb{R}^N$ is a vector of monomials in state $x \in X$, and $u \in U$ is a control input, with $X \subset \mathbb{R}^n$, and $U \subset \mathbb{R}^m$ being the state and input sets, respectively.

We now present the following lemma, taken from [8], to obtain a data-based representation of closed-loop ct-NPS (4.1) with controllers $u = K(x)\mathcal{M}(x)$, where $K(x) \in \mathbb{R}^{m\times N}$ is a matrix polynomial, which will be synthesized.

Lemma 4.2 (ct-NPS Q-Lemma [8]). *Let matrix polynomial $Q(x)$ be a $(T \times N)$ matrix such that*

$$\mathbb{I}_n = \mathcal{N}_0 Q(x), \tag{5}$$

with

$$\mathcal{N}_0 = [\mathcal{M}(x(0)), \mathcal{M}(x(\tau)), \mathcal{M}(x(2\tau)), \ldots, \mathcal{M}(x((T-1)\tau))] \tag{6}$$

being an $(N \times T)$ full row-rank matrix. If one sets $u = K(x)\mathcal{M}(x) = \mathcal{U}_0 Q(x)\mathcal{M}(x)$, then the closed-loop system $\dot{x} = A\mathcal{M}(x) + Bu$ has the following data-based representation:

$$\dot{x} = \mathcal{X}_1 Q(x)\mathcal{M}(x), \quad \textit{equivalently,} \quad A + BK(x) = \mathcal{X}_1 Q(x). \tag{7}$$

Using the above lemma, we now have a data-driven representation of our system using $\mathcal{X}_1 Q(x)\mathcal{M}(x)$, which we will use to design controllers for CBCs and stability in the following theorems.

4.1.1 Control Barrier Certificate.

Theorem 4.3 (Data-Driven ct-NPS CBC [8]). *Consider the unknown ct-NPS* (4), i.e. $\dot{x} = A\mathcal{M}(x) + Bu$, *with its data-based representation $\dot{x} = \mathcal{X}_1 Q(x)\mathcal{M}(x)$. Suppose there exists a positive definite symmetric matrix $P \in \mathbb{R}^{n\times n}$ and relationship $Q(x) = H(x)P$ such that the following conditions are satisfied*

$$\mathcal{M}(x)^\top [\mathcal{N}_0 H(x)]^{-1} \mathcal{M}(x) \leq \gamma, \quad \forall x \in X_I, \tag{8}$$

$$\mathcal{M}(x)^\top [\mathcal{N}_0 H(x)]^{-1} \mathcal{M}(x) \geq \lambda, \quad \forall x \in X_U, \tag{9}$$

and $\forall x \in X$,

$$\frac{\partial M}{\partial x}\mathcal{X}_1 H(x) + H(x)^\top \mathcal{X}_1^\top \left(\frac{\partial M}{\partial x}\right)^\top \leq 0, \tag{10}$$

$$\mathcal{N}_0 H(x) = P^{-1}. \tag{11}$$

Then $\mathcal{B}(x) = \mathcal{M}(x)^\top [\mathcal{N}_0 H(x)]^{-1} \mathcal{M}(x)$ is a CBC and $u = \mathcal{U}_0 H(x)[\mathcal{N}_0 H(x)]^{-1}\mathcal{M}(x)$ is its corresponding safety controller for the unknown ct-NPS.

4.1.2 Sum-of-Squares Formulation.

Add a reference for this formulation. We present now how to solve equations (8)-(11) in this first occurrence of the conditions, the CBC conditions for later sections follow similarly but are omitted for brevity.

We now introduce $L_I(x), L_U(x), L(x)$ as vectors of SOS polynomials. We can solve the following using the SumOfSquares toolbox [13] where we also add an additional LMI constraint for the equality (11).

We consider (11) and the following are sum-of-squares

$$-\mathcal{M}(x)^\top [\mathcal{N}_0 H(x)]^{-1} \mathcal{M}(x) - L_I^\top(x) g_I(x) + \gamma, \tag{12}$$

$$\mathcal{M}(x)^\top [\mathcal{N}_0 H(x)]^{-1} \mathcal{M}(x) - L_U^\top(x) g_U(x) + \lambda, \tag{13}$$

$$-\left[\frac{\partial M}{\partial x}\mathcal{X}_1 H(x) + H(x)^\top \mathcal{X}_1^\top \left(\frac{\partial M}{\partial x}\right)^\top\right] - L^\top(x) g(x). \tag{14}$$

A hierarchical approach to solve Theorem 4.3 is used where first $H(x)$ and P are found in equations (10)-(11) before equations (8)-(9) are solved using SOS programming for γ and λ. This is shown in Algorithm 1.

Algorithm 1 Hierarchy for ct-NPS CBCs

```
Require: $\mathcal{X}_0, \mathcal{X}_1, \mathcal{U}_0, X, X_I, X_U, \mathcal{M}(x)$
Ensure: (3) holds
Ensure: semi-algebraic $X, X_I, X_U$
  Solve (10) and (11) for $H(x)$ and $P$ via LMIs
  Substitute and solve (8)-(9) via SOS optimization (12)-(14)
  Return: CBC $\mathcal{B}(x)$, Controller $u$
```

Algorithm 2 Hierarchy for ct-NPS Stability

```
Require: $\mathcal{X}_0, \mathcal{X}_1, \mathcal{U}_0, \mathcal{M}(x)$
Ensure: (3) holds
  Solve (15) for $H(x)$ and $P$ via LMIs
  Return: CBC $\mathcal{V}(x)$, Controller $u$
```

4.1.3 Stability.

Solving only the last CBC conditions (10)-(11) with no restriction over a state space X renders the stability problem.

Theorem 4.4 (ct-NPS Stable Controller Design). *A control Lyapunov function $\mathcal{V}(x) = \mathcal{M}(x)^\top P \mathcal{M}(x)$ guaranteeing stability of the ct-NPS* (4) *can be designed by introducing the new variable*

$$Q(x) = H(x)P,$$

so that any polynomial matrices $H(x) \in \mathbb{R}^{T\times N}$, and $P \in \mathbb{R}^{N\times N}$ satisfying

$$\frac{\partial M}{\partial x}\mathcal{X}_1 H(x) + H(x)^\top \mathcal{X}_1^\top \left(\frac{\partial M}{\partial x}\right)^\top \prec 0, \quad \mathcal{N}_0 H(x) = P^{-1} \tag{15}$$

stabilize (4). *The controller is given by*
$u = \mathcal{U}_0 Q(x)\mathcal{M}(x) = \mathcal{U}_0 H(x)[\mathcal{N}_0 H(x)]^{-1}\mathcal{M}(x)$.

We remark here that the stability conditions are straightforward to solve using semidefinite programming solvers such as SEDUMI or Mosek cite whichever we use, and remove the other. The psuedocode for ct-NPS stability is given in Algorithm 2.

4.2 Continuous Time Linear Systems

We consider continuous-time linear systems (ct-LS) defined as follows:

Definition 4.5. A ct-LS is described by the ordinary differential equation

$$\Sigma_c : \dot{x} = Ax + Bu \tag{16}$$

where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are unknown, $x \in X$ is a system state, and $u \in U$ is a control input, with $X \subset \mathbb{R}^n$, and $U \subset \mathbb{R}^m$ being the state and input sets, respectively.

Remark 3. *We remark that in general ct-NPS can be simplified to the linear system by replacing $\mathcal{M}(x)$ with x and $\mathcal{N}_0$ with $\mathcal{X}_0$.*

We now present the following lemma, adapted here to ct-LS from the work [11], to obtain a data-based representation of closed-loop ct-LS (4.5) with controllers $u = Kx$, where $K \in \mathbb{R}^{m\times n}$ is a matrix, which will be synthesized.

Lemma 4.6 (ct-LS Q-Lemma [11]). *Let matrix Q be a $(T \times n)$ matrix such that*

$$\mathbb{I}_n = \mathcal{X}_0 Q, \tag{17}$$

Algorithm 3 Hierarchy for ct-LS CBCs

```
Require: $\mathcal{X}_0, \mathcal{X}_1, \mathcal{U}_0, X, X_I, X_U$
Ensure: (3) holds
Ensure: semi-algebraic $X, X_I, X_U$
  Solve (21) and (22) for $H(x)$ and $P$ via LMIs
  Substitute and solve (19)-(20) via SOS optimization
  Return: CBC $\mathcal{B}(x)$, Controller $u$
```

where $\mathcal{X}_0$ is an $(n \times T)$ full row-rank matrix. If one sets $u = Kx = \mathcal{U}_0 Qx$, then the closed-loop system $\dot{x} = Ax + Bu$ has the following data-based representation:

$$\dot{x} = \mathcal{X}_1 Qx, \quad \text{equivalently,} \quad A + BK = \mathcal{X}_1 Q. \tag{18}$$

Using the above lemma, we know have a data-driven representation of our system using $\mathcal{X}_1 Qx$, which we will use to design controllers for stability and CBCs in the following theorems.

4.2.1 *Control Barrier Certificates.*

Theorem 4.7 (Data-Driven ct-LS CBC [14]). *Consider the unknown ct-LS* (16), i.e. *$\dot{x} = Ax + Bu$, with its data-based representation $\dot{x} = \mathcal{X}_1 Qx$. Suppose there exists a positive definite symmetric matrix $P \in \mathbb{R}^{n \times n}$ and relationship $Q = HP$ such that the following conditions are satisfied*

$$x^\top [\mathcal{X}_0 H]^{-1} x \leq \gamma, \quad \forall x \in X_I, \tag{19}$$

$$x^\top [\mathcal{X}_0 H]^{-1} x \geq \lambda, \quad \forall x \in X_U, \tag{20}$$

and $\forall x \in X$,

$$\mathcal{X}_1 H + H^\top \mathcal{X}_1^\top \leq 0 \tag{21}$$

$$P^{-1} = \mathcal{X}_0 H. \tag{22}$$

Then $\mathcal{B}(x) = x^\top [\mathcal{X}_0 H]^{-1} x$ is a CBC and $u = \mathcal{U}_0 H [\mathcal{X}_0 H]^{-1} x$ is its corresponding safety controller for the unknown ct-LS.

We can again design a ct-LS CBC using a hierarchical approach as outlined in Algorithm 3.

4.2.2 *Stability.*

We highlight again that solving equations (21)-(22) without restricting X gives the ct-LS stability conditions as follows.

Theorem 4.8 (ct-LS Stable Controller Design [11]). *A control Lyapunov function $\mathcal{V}(x) = x^\top Px$ guaranteeing stability of the ct-LS* (16) *can be designed by introducing the new variable*

$$Q = HP,$$

so that any matrices $H \in \mathbb{R}^{T \times n}$, and $P \in \mathbb{R}^{n \times n}$ satisfying

$$H^\top \mathcal{X}_1^\top + \mathcal{X}_1 H \prec 0, \quad \mathcal{X}_0 H = P^{-1} \tag{23}$$

stabilize (16). *The controller is given by $u = \mathcal{U}_0 Qx = \mathcal{U}_0 H [\mathcal{X}_0 H]^{-1} x$.*

These conditions for stability are straightforward to solve using standard linear matrix inequality (LMI) solvers. The psuedocode for ct-NPS stability is given in Algorithm 4.

Algorithm 4 Hierarchy for ct-LS Stability

```
Require: $\mathcal{X}_0, \mathcal{X}_1, \mathcal{U}_0$
Ensure: (3) holds
  Solve (23) for $H(x)$ and $P$ via LMIs
  Return: CBC $\mathcal{V}(x)$, Controller $u$
```

4.3 Discrete Time Nonlinear Polynomial Systems

We consider discrete-time nonlinear polynomial systems (dt-NPS) defined as follows:

Definition 4.9. A dt-NPS is described by

$$\Sigma_d : x^+ = A\mathcal{M}(x) + Bu \tag{24}$$

where $A \in \mathbb{R}^{n \times N}$ and $B \in \mathbb{R}^{n \times m}$ are unknown, $\mathcal{M}(x) \in \mathbb{R}^N$ is a vector of monomials in state $x \in X$, and $u \in U$ is a control input, with $X \subset \mathbb{R}^n$, and $U \subset \mathbb{R}^m$ being the state and input sets, respectively.

We now present the following lemma, from the work [15], to obtain a data-based representation of closed-loop dt-NPS (4.9) with controllers $u = K(x)x$, where $K(x) \in \mathbb{R}^{m \times n}$ is a matrix, which will be synthesized.

Lemma 4.10 (dt-NPS Q-Lemma [15]). *Let $Q(x)$ be a $(T \times n)$ matrix polynomial such that*

$$\Theta(x) = \mathcal{N}_0 Q(x), \tag{25}$$

where $\Theta(x)$ is a $(N \times n)$ matrix polynomial

$$\mathcal{M}(x) = \Theta(x)x, \tag{26}$$

and where $\mathcal{N}_0$ is an $(N \times T)$ full row-rank matrix:

$$\mathcal{N}_0 = [\mathcal{M}(x(0)), \mathcal{M}(x(1)), \mathcal{M}(x(2)), \ldots, \mathcal{M}(x(T-1))]. \tag{27}$$

If one sets $u = K(x)x = \mathcal{U}_0 Q(x)x$, then the closed-loop system $x^+ = A\mathcal{M}(x) + Bu$ has the following data-based representation:

$$x^+ = \mathcal{X}_1 Q(x)x,$$

equivalently

$$A\Theta(x) + BK(x) = \mathcal{X}_1 Q(x). \tag{28}$$

Using the above lemma, we now have a data-driven representation of our system using $\mathcal{X}_1 Q(x)x$, which we will use to design controllers for CBCs and stability in the following theorems.

4.3.1 *Control Barrier Certificates.*

Theorem 4.11 (Data-Driven dt-NPS CBC [15]). *Consider the unknown dt-NPS* (24), i.e. *$x^+ = A\mathcal{M}(x) + Bu$, with its data-based representation $x^+ = \mathcal{X}_1 Q(x)x$. Suppose there exists a positive definite symmetric matrix $P \in \mathbb{R}^{n \times n}$ and relationship $Q(x) = H(x)P$ such that the following conditions are satisfied*

$$x^\top Px \leq \gamma, \quad \forall x \in X_I, \tag{29}$$

$$x^\top Px \geq \lambda, \quad \forall x \in X_U, \tag{30}$$

and $\forall x \in X$,

$$\begin{bmatrix} P^{-1} & H(x)^\top \mathcal{X}_1^\top \\ \mathcal{X}_1 H(x) & P^{-1} \end{bmatrix} \geq 0, \tag{31}$$

$$\Theta(x) P^{-1} = \mathcal{N}_0 H(x). \tag{32}$$

Algorithm 5 Hierarchy for dt-NPS CBCs

```
Require: $\mathcal{X}_0, \mathcal{X}_1, \mathcal{U}_0, X, X_I, X_U, \mathcal{M}(x)$
Ensure: (3) holds
Ensure: semi-algebraic $X, X_I, X_U$
  Solve (26) for $\Theta(x)$
  Substitute $\Theta(x)$ in (25)
  Solve (31) and (32) for $H(x)$ and $P$ via LMIs
  Substitute $P$ and solve (29)-(30) via SOS optimization
    Return: CBC $\mathcal{B}(x)$, Controller $u$
```

Algorithm 6 Hierarchy for dt-NPS Stability

```
Require: $\mathcal{X}_0, \mathcal{X}_1, \mathcal{U}_0, \mathcal{M}(x)$
Ensure: (3) holds
  Solve (26) for $\Theta(x)$
  Solve (33) for $H(x)$ and $Z$ via LMIs
    Return: CBC $\mathcal{V}(x)$, Controller $u$
```

Then $\mathcal{B}(x) = x^\top Px$ is a k-CBC and $u = \mathcal{U}_0 H(x)Px$ is its corresponding safety controller for the unknown dt-NPS.

We use the following hierarchical method for finding the CBC by first solving (31)-(32) followed by (29)-(30) afterwards. A temporary variable $Z = P^{-1}$ can be considered during the solving. The psuedocode algorithm is described in Algorithm 5.

4.3.2 Stability.

Theorem 4.12 (dt-NPS Stable Controller Design). *A control Lyapunov function $\mathcal{V}(x) = x^\top Px$ guaranteeing stability of the dt-NPS (24) can be designed by introducing the new variable*

$$Q(x)Z = H(x),$$

where $Z = P^{-1}$ so that any matrices $H(x) \in \mathbb{R}^{T\times n}$, $\Theta(x) \in \mathbb{R}^{N\times n}$, $P \in \mathbb{R}^{n\times n}$ and $Z \in \mathbb{R}^{n\times n}$ satisfying

$$\begin{bmatrix} Z & H(x)^\top \mathcal{X}_1^\top \\ \mathcal{X}_1 H(x) & Z \end{bmatrix} \succ 0, \quad \mathcal{N}_0 H(x) = \Theta(x) Z \tag{33}$$

stabilize (24). The controller is given by $u = \mathcal{U}_0 H(x) Z^{-1} x$.

Solving the stability equation (33) is straightforward using standard semidefinite programming solvers such as SEDUMI or Mosek, cite whichever we used. The psuedocode for dt-NPS stability is given in Algorithm 6.

4.4 Discrete Time Linear Systems

We consider discrete-time linear systems (dt-LS) defined as follows:

Definition 4.13. A dt-LS is described by

$$\Sigma_d : x^+ = Ax + Bu \tag{34}$$

where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are unknown, $x \in X$ is a system state, and $u \in U$ is a control input, with $X \subset \mathbb{R}^n$, and $U \subset \mathbb{R}^m$ being the state and input sets, respectively.

Remark 4. *As it was with ct-NPS to ct-LS, we remark that in general the results for dt-NPS can be simplied to linear systems by replacing $\mathcal{M}(x)$ with x, and $\mathcal{N}_0$ with $\mathcal{X}_0$.*

Algorithm 7 Hierarchy for dt-NPS CBCs

```
Require: $\mathcal{X}_0, \mathcal{X}_1, \mathcal{U}_0, X, X_I, X_U$
Ensure: (3) holds
Ensure: semi-algebraic $X, X_I, X_U$
  Solve (39) and (40) for $H(x)$ and $P$ via LMIs
  Substitute and solve (37)-(38) via SOS optimization
    Return: CBC $\mathcal{B}(x)$, Controller $u$
```

We now present the following lemma, adapted here to dt-LS from the work [11], to obtain a data-based representation of closed-loop dt-LS (4.13) with controllers $u = Kx$, where $K \in \mathbb{R}^{m\times n}$ is a matrix, which will be synthesized.

Lemma 4.14 (dt-LS Q-Lemma [11]). *Let matrix Q be a $(T \times n)$ matrix such that*

$$\mathbb{I}_n = \mathcal{X}_0 Q, \tag{35}$$

where $\mathcal{X}_0$ is an $(n \times T)$ full row-rank matrix. If one sets $u = Kx = \mathcal{U}_0 Qx$, then the closed-loop system $x^+ = Ax + Bu$ has the following data-based representation:

$$x^+ = \mathcal{X}_1 Qx, \quad \text{equivalently,} \quad A + BK = \mathcal{X}_1 Q. \tag{36}$$

We use the above lemma, to describe our system in terms of its data-driven representation $\mathcal{X}_1 Qx$ and use that representation in the following theorems.

4.4.1 Control Barrier Certificate.

Theorem 4.15 (Data-Driven dt-LS CBC). *Consider the unknown dt-LS (34), i.e. $x^+ = Ax + Bu$, with its data-based representation $x^+ = \mathcal{X}_1 Qx$. Suppose there exists a positive definite symmetric matrix $P \in \mathbb{R}^{n\times n}$ and relationship $Q = HP$ such that the following conditions are satisfied*

$$x^\top Px \leq \gamma, \quad \forall x \in X_I, \tag{37}$$

$$x^\top Px \geq \lambda, \quad \forall x \in X_U, \tag{38}$$

and $\forall x \in X$,

$$\begin{bmatrix} P^{-1} & H^\top \mathcal{X}_1^\top \\ \mathcal{X}_1 H & P^{-1} \end{bmatrix} \geq 0, \tag{39}$$

$$P^{-1} = \mathcal{X}_0 H. \tag{40}$$

Then $\mathcal{B}(x) = x^\top Px$ is a CBC and $u = \mathcal{U}_0 HPx$ is its corresponding safety controller for the unknown dt-LS.

By substituting $Z = P^{-1}$, the following hierarchical method is used for finding the CBC by solving (39)-(40) first to find H and P^{-1} followed by solving (37)-(38) afterwards. The psuedocode algorithm for this is described in Algorithm 7.

4.4.2 Stability. Removing the restriction over X for conditions (39)-(40) gives the stability conditions.

Theorem 4.16 (dt-LS Stable Controller Design [11]). *A control Lyapunov function $\mathcal{V}(x) = x^\top Px$ guaranteeing stability of the dt-LS (34) can be designed by introducing the new variable*

$$QZ = H,$$

Algorithm 8 Hierarchy for dt-NPS Stability

```
Require: $\mathcal{X}_0, \mathcal{X}_1, \mathcal{U}_0$
Ensure: (3) holds
Solve (41) for $H(x)$ and $Z$ via LMIs
    Return: CBC $\mathcal{V}(x)$, Controller $u$
```

where $Z = P^{-1}$ and $P \in \mathbb{R}^{n \times n}$ so that any matrices $H \in \mathbb{R}^{T \times n}$, and $Z \in \mathbb{R}^{n \times n}$ satisfying

$$\begin{bmatrix} Z & H^\top \mathcal{X}_1^\top \\ \mathcal{X}_1 H & Z \end{bmatrix} \succ 0, \quad \mathcal{X}_0 H = Z \tag{41}$$

stabilize (34)*. The controller is given by $u = \mathcal{U}_0 H[\mathcal{X}_0 H]^{-1} x$.*

The above theorem can be solved in a straightforward manner using any LMI solver. The psuedocode for dt-LS stability is given in Algorithm 8.

5 BENCHMARKS AND CASE STUDIES

We consider the following dynamical systems from the literature as case study examples.

5.1 Continuous-Time Systems

5.1.1 *Nonlinear Systems.*

We consider the following ct-NPS systems.

Jet Engine We consider the nonlinear Moore-Greitzer jet engine model [**?**]:

$$\Sigma_c : \begin{cases} \dot{x}_1(t) = -x_2(t) - 1.5x_1^2(t) - 0.5x_1^3(t), \\ \dot{x}_2(t) = x_1(t), \end{cases}$$

where $x_1 = \Phi - 1, x_2 = \Psi - \Lambda - 2$, with Φ, Ψ, Λ being, respectively, the mass flow, pressure rise, and a constant. We consider $X = [0.1, 1]^2$, $X_I = [0.1, 0.5]^2$, and $X_U = [0.7, 1]^2$. add details above initial state and trajectory length.

5.1.2 *Linear Systems.*

We consider the following ct-LS systems.

FOSSIL benchmark - High Order 8. We consider the following 8-dimensional benchmark [16]

$$\Sigma_c : \begin{cases} \dot{x}_1(t) = & x_2(t), \\ \dot{x}_2(t) = & x_3(t), \\ \dot{x}_3(t) = & x_4(t), \\ \dot{x}_4(t) = & x_5(t), \\ \dot{x}_5(t) = & x_6(t), \\ \dot{x}_6(t) = & x_7(t), \\ \dot{x}_7(t) = & x_8(t), \\ \dot{x}_8(t) = & -20x_8(t) - 170x_7(t) - 800x_6(t) - 2273x_5(t) \\ & -3980x_4(t) - 4180x_3(t) - 2400x_2(t) - 576x_1(t), \end{cases}$$

with the state space $X = [-2.2, 2.2]^8$, initial region $X_I = [0.9, 1.1]^8$, and unsafe region $X_U = [-2.2, -1.8]^8$. add details above initial state and trajectory length.

5.2 Discrete-Time Systems

5.2.1 *Nonlinear Systems.*

We consider the following dt-NPS systems.

Lorenz Attractor We consider the Lorenz attractor a well studied dynamical system with chaotic behavior. The map has dynamics

$$\Sigma : \begin{cases} x_1(k+1) & = x_1(k) + \tau\sigma(x_1(k) + x_2(k)), \\ x_2(k+1) & = x_2(k) + \\ & \quad \tau(\rho x_1(k) - x_2(k) - x_1(k)x_3(k) + u(k)), \\ x_3(k+1) & = x_3(k) + \tau(x_1(k)x_2(k) - \beta x_3(k)), \end{cases}$$

where $\rho = 28, \sigma = 10, \beta = \frac{8}{3}, tau = 10^{-3}$. We consider the state space $X = [-5, 5]^3$, initial set $X_I = [-1, 1]^3$ and two unsafe sets $X_U = [-5, -2.5]^3 \cup [2.5, 5]^3$. We consider monomials from the set $\mathcal{M}(x) =$

$$[x_1(k)\ x_2(k)\ x_3(k)\ x_1(k)x_2(k)\ x_2x_3(k)\ x_1(k)x_3(k)]^\top,$$

and collect data $\mathcal{X}_0$ and $\mathcal{X}_1$ with time horizon $T = 50$, equivalently 0.05s, from initial point $(0.5, 0.5, 0.5)$ and choose random inputs along the evolution of the trajectory from the set $U = [-100, 100]$.
Second Nonlinear

5.2.2 *Linear Systems.*

We consider the following dt-LS systems.

RLC Circuit.

$$\Sigma : \begin{cases} i(k+1) = i(k) + \tau(-\frac{R}{L}i(k) - \frac{1}{L}v(k)) + u_1(k), \\ v(k+1) = v(k) + \tau(\frac{1}{C}i(k)) + u_2(k), \end{cases}$$

We consider the safety of an RLC Circuit, based on [17], with the dynamics given above, where $i(k)$ denotes the current at time k, $v(k)$ is the voltage, $\tau = 0.5$s is the sampling time, $R = 2\omega$ is the series resistance, $L = 9$H is the series inductance, $C = 0.5$F is the capacitance of the circuit. We use hypercubes as the state space of the system $X = [-2, 2] \times [-4, 4]$, initial set $X_I = [0, 0.5] \times [0, 1]$, and unsafe set $X_U = [1, 2] \times [-4, 4]$. We construct $\mathcal{X}_0, \mathcal{U}_0$ and $\mathcal{X}_1$ by taking a 15s time horizon trajectory, with initial state $(0, 0)$ and random inputs along the evolution of the trajectory from the set $U = [-1, 1] \times [-1, 1]$.

DC Motor.

$$\Sigma : \begin{cases} x_1(k+1) = x_1(k) - \tau(\frac{R}{L}x_1(k) + \frac{k_{dc}}{L}x_2(k)) + u_1(k), \\ x_2(k+1) = x_2(k) + \tau(\frac{k_{dc}}{J}x_1(k) - \frac{b}{J}x_2(k)) + u_2(k), \end{cases}$$

We also consider the safe operation of a DC Motor, based on [18], with the dynamics given above where $x_1, x_2, R = 1, L = 0.01, J = 0.01$ are the armature current, the rotational speed of the shaft, the electrical resistance, the electrical inductance, and the moment of inertia of the rotor, respectively. In addition, $\tau = 0.01$, $b = 1$, and $k_{dc} = 0.01$, represent, respectively, the sampling time, the motor torque, and the back electromotive force. We use hypercubes as the state space of the system $X = [0.1, 0.5] \times [0.1, 1]$, initial set $X_I = [0.1, 0.4] \times [0.1, 0.55]$, and unsafe set $X_U = [0.45, 0.5] \times [0.6, 1]$. We first construct the matrices $\mathcal{X}_0, \mathcal{U}_0$ and $\mathcal{X}_1$ by taking a trajectory with time horizon 15s. We construct $\mathcal{X}_0, \mathcal{U}_0$ and $\mathcal{X}_1$ by taking a 15s time horizon trajectory, with initial state $(0, 0)$ and random inputs along the evolution of the trajectory from the set $U = [-0.1, 0.1] \times [-0.1, 0.1]$.

Room Temperature System.

$$\Sigma_d : \begin{cases} x_1(k+1) = (1 - \tau(\alpha + \alpha_{e1}))x_1(k) + \tau\alpha x_2(k) + \tau\alpha_{e1}T_e, \\ x_2(k+1) = (1 - \tau(\alpha + \alpha_{e2}))x_2(k) + \tau\alpha x_1(k) + \tau\alpha_{e2}T_e, \end{cases}$$

We consider the two room system from the FOSSIL benchmarks [16], with dynamics above, where the discretization parameter $\tau = 5$, heat exchange constants $\alpha = 5{\times}10^{-2}$, $\alpha_{e1} = 5{\times}10^{-3}$, $\alpha_{e2} = 8{\times}10^{-3}$, and external temperature $T_e = 15$. We consider regions of interest $X = [18, 23]^2$, $X_I = [18, 19.75]^2$, and $X_U = [22, 23]$. add details above initial state and trajectory length. This is a verification example as no control inputs are present and no controller is designed.

Two Tank System.

$$\Sigma_d^{\varsigma}: \begin{cases} h_1(k+1) = (1 - \tau\frac{\alpha_1}{A_1})h_1(k) + \tau\frac{q_1(k)}{A_1}, \\ h_2(k+1) = \tau\frac{\alpha_1}{A_2}h_1(k) + (1 - \tau\frac{\alpha_2}{A_2})h_2(k) + \tau\frac{q_o(k)}{A_2}, \end{cases}$$

Consider a two-tank system [**?**], characterized by the above difference equations where h_1, h_2 are heights of the fluid in two tanks. Additionally, α_i and A_i are the valve coefficient and area of tank i, and q_1 and q_o are the inflow and outflow rate of tank 1 and 2, respectively. Furthermore, $\tau = 0.1$, $\frac{\alpha_1}{A_1} = 1, \frac{q_1}{A_1} = 4.5$, $\frac{\alpha_1}{A_2} = 1$, $\frac{\alpha_2}{A_2} = 1$ and $\frac{q_o}{A_2} = -3$. Regions of interest are $X = [1, 10] \times [1, 10]$, $X_I = [1.75, 2.25] \times [1.75, 2.25]$, $X_U = [9, 10] \times [9, 10]$. add details above initial state and trajectory length. This is a verification example as no control inputs are present and no controller is designed.

6 RESULTS

- Create a full page table for the four scenarios, each with the safety function, stability barrier function and controller. - Compare to other tools both in terms of performance and data requirements - MATLAB w/ these algorithms? Presumably can cite Behrad's work for dt-NPS? - Can we cite PRoTECT but state that we're not comparing since we're not dealing with stochastic systems? (Yes, but we'll do this in the intro's related materials subsection)

REFERENCES

[1] A. Nejati and M. Zamani, "Data-driven synthesis of safety controllers via multiple control barrier certificates," *IEEE Control Systems Letters*, 2023.

[2] V. Breschi, C. De Persis, S. Formentin, and P. Tesi, "Direct data-driven model-reference control with lyapunov stability guarantees," in *2021 60th IEEE conference on decision and control (CDC)*. IEEE, 2021, pp. 1456–1461.

[3] T. Ren, W. Lin, and Z. Ding, "Formal synthesis of safety controllers via k-inductive control barrier certificates," *IEEE Transactions on Reliability*, 2024.

[4] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.

[5] S. Prajna, A. Papachristodoulou, and P. A. Parrilo, "Introducing sostools: A general purpose sum of squares programming solver," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 1. IEEE, 2002, pp. 741–746.

[6] A. Bisoffi, C. De Persis, and P. Tesi, "Data-based guarantees of set invariance properties," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 3953–3958, 2020.

[7] ——, "Controller design for robust invariance from noisy data," *IEEE Transactions on Automatic Control*, vol. 68, no. 1, pp. 636–643, 2022.

[8] A. Nejati, B. Zhong, M. Caccamo, and M. Zamani, "Data-driven controller synthesis of unknown nonlinear polynomial systems via control barrier certificates," in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 763–776.

[9] T. Martin, T. B. Schön, and F. Allgöwer, "Guarantees for data-driven control of nonlinear systems using semidefinite programming: A survey," *Annual Reviews in Control*, p. 100911, 2023.

[10] B. Samari, O. Akbarzadeh, M. Zaker, and A. Lavaei, "From a single trajectory to safety controller synthesis of discrete-time nonlinear polynomial systems," *arXiv preprint arXiv:2409.10026*, 2024.

[11] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 909–924, 2019.

[12] P. A. Parrilo, "Semidefinite programming relaxations for semialgebraic problems," *Mathematical programming*, vol. 96, pp. 293–320, 2003.

[13] C. Yuan, "SumOfSquares.py." [Online]. Available: https://github.com/yuanchenyang/SumOfSquares.py

[14] H. Wang, K. Margellos, A. Papachristodoulou, and C. De Persis, "Convex co-design of control barrier functions and feedback controllers for linear systems," 2024.

[15] B. Samari, O. Akbarzadeh, M. Zaker, and A. Lavaei, "From a single trajectory to safety controller synthesis of discrete-time nonlinear polynomial systems," 2024. [Online]. Available: https://arxiv.org/abs/2409.10026

[16] A. Abate, D. Ahmed, A. Edwards, M. Giacobbe, and A. Peruffo, "FOSSIL: a software tool for the formal synthesis of lyapunov functions and barrier certificates using neural networks," in *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, 2021, pp. 1–11.

[17] M. Anand, V. Murali, A. Trivedi, and M. Zamani, "K-inductive barrier certificates for stochastic systems," in *Proceedings of the 25th ACM International Conference on Hybrid Systems: Computation and Control*, 2022, pp. 1–11.

[18] P. A. Adewuyi, "Dc motor speed control: A case between pid controller and fuzzy logic controller," *international journal of multidisciplinary sciences and engineering*, vol. 4, no. 4, pp. 36–40, 2013.