Safe Autonomous Takeover for Unknown Descriptor Systems

Ben Wooding[1], Mohamad H. Kazma[2], Hongchao Zhang[1],
Ahmad F. Taha[2], Meiyi Ma[1], and Taylor T. Johnson[1]

***Abstract*— Discrete-time descriptor systems are systems with dynamics described by difference equations that are subject to algebraic constraints. These constraints make standard state-space control design approaches significantly more challenging. Recent works have designed control barrier certificates (CBCs) to ensure safety of *unknown* discrete-time systems via data-driven parameterization methods. However, the singular nature of a descriptor system ensures that collected data trajectories will not satisfy the *full-row rank condition* required for persistence of excitation. Essentially, the data will not be rich enough to replace knowing the full descriptor system. In this work, we derive an algorithmic approach to design a safe control policy *online* for unknown discrete-time *nonlinear* descriptor systems, certified by a CBC. We prove that data collected from the subspace described by the algebraic constraints is rich enough to guarantee persistence of excitation. We then demonstrate *autonomous takeover*, a user (who selects safe input choices until autonomous takeover is ready) passes system control fully to a safe autonomous control strategy, guaranteeing safe behavior for all future time steps. We consider two unstable descriptor systems; an ecological case study and a (*trigonometric*) constrained inverted pendulum.**

I. Introduction and Paper Contributions

A discrete-time descriptor system is a mathematical model for dynamic systems where the state evolution is defined by a mixture of difference equations and algebraic constraints. Both discrete-time and continuous-time descriptor systems [1]–[4] describe many constrained real-world applications, and have significant relevance to system-theoretic analysis. Verifying safety properties (preventing unsafe behavior) for such systems is therefore of significant worth. While control barrier certificates (CBCs) [5] are an established method to guarantee system safety for difference equations [6] and ordinary differential equations [7], little research has been conducted to guarantee safety for descriptor systems.

Few works combine formal approaches with descriptor system analysis, with the power system domain primarily considered. The work [8] considers explicit differential-algebraic barrier certificate verification but no control design. The work [9] synthesizes distributed barrier-based control policies on polynomial approximations of microgrid dynamics. The works [10], [11] consider reachability analysis to formally verify descriptor power systems. Only the works [12] and [13] design a control policy to guarantee safety while fully considering the algebraic constraints. In this setting, our work is unique as it is the first work to design CBCs for *discrete-time* descriptor systems; we also derive all results *from data* assuming *unknown* system coefficients.

Data-driven discrete-time CBC approaches have been explored in [14]–[18], leveraging data parameterization for controller synthesis for non-algebraic systems. The work [19] uses the scenario approach to design barrier certificates purely from data, but does not consider controller design. We consider that the descriptor systems under analysis have unknown matrix coefficients.

Leveraging results from the data-parameterization literature [20]–[22], we demonstrate how a single input-state trajectory can provide safety guarantees for an unknown *nonlinear* descriptor system. The considered conditions build upon the foundation of Willems' fundamental lemma [23], but are more relaxed, with non-consecutive data trajectories being viable. Such results for descriptor systems have been considered by [21] and [24]. However, these works consider only discrete-time *linear* descriptor systems, while we demonstrate success for *nonlinear* descriptor systems.

Contributions. This work studies data-driven control of nonlinear descriptor systems with formal guarantees. Our paper contributes the following:

- We provide sufficient conditions to design a data-driven control policy and CBC, guaranteeing safety for *unknown discrete-time nonlinear* descriptor systems;
- We show theoretically how the persistency of excitation (PE) rank condition can be satisfied by projecting the dynamic equation onto the algebraic subspace during data collection, certifying the data is rich enough to use in place of knowing the true system;
- We demonstrate a novel *online* algorithmic framework, enabling a control policy to conduct *autonomous takeover* from a user once safety is assured. We demonstrate the approach on two unstable nonlinear index-1 descriptor systems, including a *trigonometric* nonlinear term (without approximation).

Notation. We use $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{>0}$, $\mathbb{C}$, $\mathbb{N}$ and $\mathbb{N}_{>0}$ to denote real numbers, non-negative real numbers, positive real numbers, complex numbers, and non-negative and positive integers, respectively. $\mathbb{I}_n$ is the $\mathbb{R}^{n\times n}$ identity matrix. Respectively, $\mathrm{rank}(A)$ and $\det(A)$ denote the row rank and determinant of matrix A. $A^\dagger$ indicates the Moore-Penrose pseudoinverse of A. We denote the empty set by $\emptyset$.

Organization. Section II presents necessary background. Section III outlines projecting the dynamics of the descriptor

[1] Institute for Convergent Software Integrated Systems, [2] Department of Civil and Environmental Engineering, Vanderbilt University, Nashville, TN 37235 USA. Corresponding author: ben.wooding@vanderbilt.edu

The material presented in this paper is based upon work supported by the National Science Foundation (NSF) through grant numbers 2220401, 2443803, 2152450 and 2550934, and the Defense Advanced Research Projects Agency (DARPA) under contract number FA8750-23-C-0518. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of DARPA or NSF.

system onto its algebraic subspace. Section IV derives the data-driven approach to guarantee safety. Section V shows our algorithm. Section VI demonstrates the approach on two examples. Finally, Section VII concludes.

II. Key Concepts and Problem Statement

The descriptor systems under study are nonlinear with unknown matrix coefficients and evolve in discrete-time. Using data, we will design a control policy that guarantees the safety of such a system.

Definition 1 (Nonlinear Descriptor System)**.** *A discrete-time nonlinear descriptor system in this work is described by*

$$E\mathbf{x}(k+1) = A\mathfrak{D}(\mathbf{x}(k)) + B\mathbf{u}(k), \tag{1}$$

with consistent initial condition $\mathbf{x}(0)$*, where* $\mathbf{x}(k) \in \mathbb{R}^n$ *are system states,* $\mathbf{u}(k) \in \mathbb{R}^m$ *are control inputs, and* $\mathfrak{D}(\mathbf{x}(k)) \in \mathbb{R}^N$ *is a dictionary of* known *nonlinear terms. The matrices* $A \in \mathbb{R}^{n\times N}, E \in \mathbb{R}^{n\times n}$*, and* $B \in \mathbb{R}^{n\times m}$ *are all assumed to be* unknown*.*

When clear from context, we ease notation by dropping the variable k, *i.e.* we use $\mathbf{x}$ for $\mathbf{x}(k)$, and $\mathbf{x}^+$ for $\mathbf{x}(k+1)$.

Assumption 1. *We assume* $\mathfrak{D}(\mathbf{x}) = \Theta(\mathbf{x})\mathbf{x}$ *is known, where* $\Theta(\mathbf{x})$ *is of appropriate dimension.*

While we assume that the coefficient matrices are *unknown*, reflecting practical scenarios, the dictionary $\mathfrak{D}$ is assumed to be available. The dictionary should be *sufficiently comprehensive* to capture the true nonlinearities in the actual system even if irrelevant terms are included. This decomposition is widely used [25], *e.g.,* the dictionary $\mathfrak{D}$ may be derived from first principles, despite the specific coefficients remaining unknown. We make a semantic choice to describe this system as *unknown*. If matrix E in (1) is nonsingular, then the system can be rewritten as an explicit difference equation,

$$\mathbf{x}^+ = E^{-1}A\mathfrak{D}(\mathbf{x}) + E^{-1}B\mathbf{u}.$$

This paper therefore focuses on the descriptor system (1) when E is singular, $\mathrm{rank}(E) < n$, and the system is *causal*.

A. Generating Persistently Excited Data

Data parameterization and persistency of excitation (PE) has been used to *exactly* represent the behavior of an unknown system for controller design using data. We represent the state-input data collected from the unknown descriptor over the time horizon $[0, T]$, where $T \in \mathbb{N}_{>0}$, as:

$$\mathbb{X}_0 = [\ \mathbf{x}(0),\ \mathbf{x}(1),\ \ldots,\ \mathbf{x}(T-1)] \in \mathbb{R}^{n\times T}, \tag{2a}$$
$$\mathbb{X}_1 = [\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(T)] \in \mathbb{R}^{n\times T}, \tag{2b}$$
$$\mathbb{U}_0 = [\mathbf{u}(0), \mathbf{u}(1), \ldots, \mathbf{u}(T-1)] \in \mathbb{R}^{m\times T}. \tag{2c}$$

We also include the data representation of the extensive dictionary of nonlinear functions:

$$\mathbb{D}_0 = [\mathfrak{D}(\mathbf{x}(0)), \ldots, \mathfrak{D}(\mathbf{x}(T-1))], \in \mathbb{R}^{N\times T}. \tag{2d}$$

We name (2) collectively as a single input-state trajectory. To ensure PE, the collected data must satisfy full-row rank conditions of the data matrices (2), this fact is related to the famous *fundamental lemma* [23]. For model identification [20], it is necessary that, $\mathbb{D}_0$ and $\mathbb{U}_0$ are full-row rank,

$$\mathrm{rank}\begin{bmatrix}\mathbb{U}_0\\ \mathbb{D}_0\end{bmatrix} = N + m.$$

For the simpler case of learning a control policy [26], it is sufficient that only $\mathbb{D}_0$ is full-row rank,

$$\mathrm{rank}\begin{bmatrix}\mathbb{D}_0\end{bmatrix} = N,$$

to guarantee PE. For the derivations of these conditions, we refer to the original works [20], [26]. Crucially, applying this approach directly to (1) fails. With E singular, the matrices cannot satisfy the full-row rank conditions required to parameterized the descriptor system.

B. Control Barrier Certificates

In this study we design a control policy such that all trajectories evolving from an initial region remain safe. The following definition defines safety.

Definition 2 (Safety)**.** *The descriptor system* (1) *is safe if all state trajectories evolving from the initial region* $\mathbf{x}(0) \in \mathcal{X}_\mathcal{I}$ *never reach the unsafe region* $\mathcal{X}_\mathcal{U}$*,* i.e.*,* $\mathbf{x}(k) \notin \mathcal{X}_\mathcal{U},\ \forall k \in \mathbb{N}$*.*

A well-established safety guarantee is the synthesis of a control barrier certificate (CBC) [5].

Definition 3 (CBC)**.** *Let* $\mathbf{x}^+ = f(\mathbf{x}, \mathbf{u})$ *be a general discrete-time system and* $\mathcal{X}, \mathcal{X}_\mathcal{I}, \mathcal{X}_\mathcal{U}$ *the given state space, initial region, and unsafe region, respectively, where* $\mathcal{X}_\mathcal{I}, \mathcal{X}_\mathcal{U} \subset \mathcal{X}$ *and* $\mathcal{X}_\mathcal{I} \cap \mathcal{X}_\mathcal{U} = \emptyset$*. Suppose there exists a control policy* $\mathbf{u}$ *with control barrier certificate* $\mathfrak{B} : \mathcal{X} \to \mathbb{R}$ *that satisfies the following conditions:*

$$\mathfrak{B}(\mathbf{x}) \le \beta_1, \qquad \forall \mathbf{x} \in \mathcal{X}_\mathcal{I} \tag{3a}$$
$$\mathfrak{B}(f(\mathbf{x}, \mathbf{u})) \le \mathfrak{B}(\mathbf{x}), \qquad \forall \mathbf{x} \in \mathcal{X}_\mathcal{I} \tag{3b}$$
$$\mathfrak{B}(\mathbf{x}) \ge \beta_2, \qquad \forall \mathbf{x} \in \mathcal{X}_\mathcal{U} \tag{3c}$$

where $\beta_1, \beta_2 \in \mathbb{R}$ *are the sublevel sets of* $\mathfrak{B}(\mathbf{x})$*, and* $\beta_2 > \beta_1$*, then the safety of the discrete-time system is guaranteed.*

These safety conditions are analogous to Lyapunov functions. Since, the unsafe region $\mathcal{X}_\mathcal{U}$ maps to larger values than the values of the initial region $\mathcal{X}_\mathcal{I}$, then, a control policy $\mathbf{u}$ ensuring the barrier value of any state transitions (initialized in $\mathcal{X}_\mathcal{I}$) are non-increasing assures safety.

C. Problem Statement

We now describe the problem we address in this paper.

Problem Statement. Consider the singular system (1) with unknown matrices E, A and B, a known dictionary $\mathfrak{D}(\mathbf{x}(k))$, and initial and unsafe sets $\mathcal{X}_\mathcal{I}, \mathcal{X}_\mathcal{U} \subset \mathcal{X}$ where $\mathcal{X}_\mathcal{I} \cap \mathcal{X}_\mathcal{U} = \emptyset$. Design a safe control policy $\mathbf{u}(k)$ certified by a CBC $\mathfrak{B}(\mathbf{x}(k))$, using a single input-state trajectory (2) collected from the user-controlled unknown system, and enabling the online transfer of system control from user to the safe autonomous policy.

In previous work, the data-driven control policy design for ODE systems has been hierarchical (*e.g.* [18]): (i) a persistently excited data trajectory is collected, (ii) a CBC and

safe control policy are designed offline, (iii) the controller is deployed to the system. We present here the novelty of collecting the data and designing the control policy online, our algorithmic approach can alert a user when *autonomous takeover* is possible.

III. Projection to Algebraic Subspace

We motivate this section using linear systems.

A. Discrete-Time Linear Descriptor Systems

Definition 4 (Linear Descriptor System)**.** *A discrete-time linear descriptor system is described by*

$$E\mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k), \tag{4}$$

with consistent initial condition $\mathbf{x}(0)$*, where* $\mathbf{x}(k) \in \mathbb{R}^n$ *are system states and* $\mathbf{u}(k) \in \mathbb{R}^m$ *are control inputs. The matrices* $A, E \in \mathbb{R}^{n\times n}$*, and* $B \in \mathbb{R}^{n\times m}$*.*

Assumption 2 (Regularity)**.** *We make a standard assumption throughout this work that* $\det(\lambda E - A) \neq 0$ *for some* $\lambda \in \mathbb{C}$*,* i.e. *system* (4) *is regular.*

Regularity is standard in practice, *e.g.* for power systems [27]. Ensuring the well-posedness and uniqueness of descriptor systems under analysis. Since (4) is regular, the following lemma describes a transformation to a semi-explicit representation, enabling the separation of difference equations and algebraic constraints. Computing this form is described further in [2, Chap. 8].

Lemma 1 (Canonical Form)**.** *Suppose* (4) *is regular. There exist nonsingular matrices* $S \in \mathbb{R}^{n\times n}$ *and* $Q \in \mathbb{R}^{n\times n}$ *such that:*

$$SEQ = \begin{bmatrix} \mathbb{I}_{n_1} & 0 \\ 0 & N \end{bmatrix}, \quad SAQ = \begin{bmatrix} A_1 & 0 \\ 0 & \mathbb{I}_{n_2} \end{bmatrix}, \quad SB = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix},$$

where $A_1 \in \mathbb{R}^{n_1\times n_1}$ *and* $N \in \mathbb{R}^{n_2\times n_2}$ *is nilpotent, with nilpotency index* $\mu \in \mathbb{N}$ *and* $n_1 + n_2 = n$*. With a change of variables,*

$$\mathbf{x} = Q \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}.$$

The canonical form is then,

$$\begin{bmatrix} \mathbb{I}_{n_1} & 0 \\ 0 & N \end{bmatrix} \begin{bmatrix} \mathbf{x}_1^+ \\ \mathbf{x}_2^+ \end{bmatrix} = \begin{bmatrix} A_1 & 0 \\ 0 & \mathbb{I}_{n_2} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \mathbf{u}.$$

We therefore acquire two equations, a forward recurrent equation uniquely determined by $\mathbf{x}(0)$ and $\mathbf{u}(k)$,

$$\mathbf{x}_1(k) = A_1^k \mathbf{x}_1(0) + \sum_{i=0}^{k-1} A_1^{k-i-1} B_1 \mathbf{u}(i),$$

and a backward recurrent equation, described using the terminal time instance L, and uniquely determined by the terminal state $\mathbf{x}_2(L)$ and $\mathbf{u}(k)$,

$$\mathbf{x}_2(k) = N^{L-k}\mathbf{x}_2(L) + \sum_{i=0}^{L-k-1} N^i B_2 \mathbf{u}(k+i). \tag{5}$$

Therefore the state at any time point in a descriptor system is related to the initial state $\mathbf{x}(0)$ and former inputs, as well as the terminal state $\mathbf{x}(L)$ and future inputs up to time step L. When the descriptor system can be described purely by its initial state $\mathbf{x}(0)$ and former inputs $\mathbf{u}(i)$ for $i = 0, 1, \dots k-1$, the system is *causal*. From (5), it is clear when $N \neq 0$, then the system is *non-causal*, and the descriptor system may be described by future unseen inputs $\mathbf{u}(k+i)$.

Definition 5 (Nilpotency Index [2])**.** *The smallest* $\mu \in \mathbb{N}$*, such that* $N^\mu = 0$*.*

If the nilpotency index is known, then (5) simplifies to the substate

$$\mathbf{x}_2(k) = \sum_{i=0}^{\mu-1} N^i B_2 \mathbf{u}(k+i),$$

however, future inputs would still be required, and the formulation may be non-causal. System (4) is causal when the *nilpotency index* $\mu \in \{0, 1\}$.

In our problem setting, the nonlinear system (1) has constant matrices E, A, B (only $\mathfrak{D}(\mathbf{x})$ is nonlinear), so the pencil $(\lambda E - A)$ is still well-defined and an equivalent canonical form exists for (1) (admitting a semi-explicit representation). We therefore directly continue with the regularity assumption and nilpotency index conceptualization in our nonlinear setting. As we consider only causal systems, and with $\mu = 0$ a trivial case (non-singular), we limit our study to nonlinear descriptor systems with nilpotency index $\mu = 1$.

B. Projection of Nonlinear System

To represent (1) using parameterized data, we need to satisfy the before mentioned full-row rank conditions. The chief culprit of this failure is the algebraic constraints caused by matrix E being singular. By Lemma 1, consider a *causal* semi-explicit representation of (1):

$$\overbrace{\begin{bmatrix} \mathbb{I}_{n_d} & 0 \\ 0 & 0 \end{bmatrix}}^{E} \overbrace{\begin{bmatrix} \mathbf{x}_d^+ \\ \mathbf{x}_a^+ \end{bmatrix}}^{\mathbf{x}^+} = \overbrace{\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} \mathfrak{D}_d(\mathbf{x}_d) \\ \mathfrak{D}_a(\mathbf{x}_a) \end{bmatrix}}^{\mathfrak{D}(\mathbf{x})} + \overbrace{\begin{bmatrix} B_1 \\ B_2 \end{bmatrix}}^{B} \mathbf{u},$$

where $\mathbf{x}_d(k) \in \mathbb{R}^{n_d}$ and $\mathbf{x}_a(k) \in \mathbb{R}^{n_a}$ are the separated dynamic states and algebraic states of the system $\mathbf{x} = [\mathbf{x}_d^\top \ \mathbf{x}_a^\top]^\top$, and $n_d + n_a = n$. Similarly, we extend Assumption 1 to partition the dictionary $\mathfrak{D}(\mathbf{x})$ into $\mathfrak{D}_d(\mathbf{x}_d) \in \mathbb{R}^{N_d}$ and $\mathfrak{D}_a(\mathbf{x}_a) \in \mathbb{R}^{N_a}$, where $\mathfrak{D}_i(\mathbf{x}_i) = \Theta_i(\mathbf{x}_i)\mathbf{x}_i$ for $i \in \{d, a\}$, and $N_d + N_a = N$. The introduced sub-matrices are considered to be of appropriate size, and $\mathbf{u}$ is unchanged. We simplify to get the following:

$$\mathbf{x}_d^+ = A_{11}\mathfrak{D}_d(\mathbf{x}_d) + A_{12}\mathfrak{D}_a(\mathbf{x}_a) + B_1\mathbf{u}, \tag{6a}$$

$$0 = A_{21}\mathfrak{D}_d(\mathbf{x}_d) + A_{22}\mathfrak{D}_a(\mathbf{x}_a) + B_2\mathbf{u}, \tag{6b}$$

To satisfy the rank condition, we intend to project the dynamics of the system (6a) onto the algebraic subspace $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n : A_{21}\mathfrak{D}_d(\mathbf{x}_d) + A_{22}\mathfrak{D}_a(\mathbf{x}_a) + B_2\mathbf{u} = 0\}$, described by (6b). This projection lets us consider a reduced-order difference equation that satisfies the rank condition while not abstracting away any crucial impulse behaviors that can destroy the system in practice [2].

We present the following theorem, providing the reduced-order difference equation that is a projection of (6a) onto the constraints subspace (6b). We call this new system description the *projected system* or *projected model*.

Theorem 1 (Projected System)**.** *For nonsingular matrix A_{22}, the dynamics $\mathbf{x}_d$ in the causal full descriptor system* (6)*, when the trajectories lie on the algebraic subspace* (6b)*, are exactly equivalent to the difference equation*

$$\mathbf{z}^+ = \tilde{A}\mathfrak{D}_d(\mathbf{z}) + \tilde{B}\mathbf{u}, \tag{7}$$

where $\mathbf{z} = \mathbf{x}_d$, $\tilde{A} = A_{11} - A_{12}A_{22}^{-1}A_{21}$, and $\tilde{B} = B_1 - A_{12}A_{22}^{-1}B_2$.

Proof. With A_{22} nonsingular, we can rewrite (6b) as the following:

$$\mathfrak{D}_a(\mathbf{x}_a) = -A_{22}^{-1}[A_{21}\mathfrak{D}_d(\mathbf{x}_d) + B_2\mathbf{u}].$$

We project the dynamics into the subspace by canceling $\mathfrak{D}_a(\mathbf{x}_a)$ in (6a):

$$\mathbf{x}_d^+ = [A_{11} - A_{12}A_{22}^{-1}A_{21}]\mathfrak{D}_d(\mathbf{x}_d) + [B_1 - A_{12}A_{22}^{-1}B_2]\mathbf{u}.$$

Substituting $\mathbf{z} = \mathbf{x}_d$, $\tilde{A} = A_{11} - A_{12}A_{22}^{-1}A_{21}$, and $\tilde{B} = B_1 - A_{12}A_{22}^{-1}B_2$ completes the proof. □

The matrices $\tilde{A}$ and $\tilde{B}$ abstract away the unknown matrices of (6), and are themselves unknown. Using the parameterization approach of this work we never need to compute $\tilde{A}$ and $\tilde{B}$ explicitly, and they are eliminated when constructing the conditions that require only the single input-state trajectory.

Remark 1. *As $\mathcal{S} \subset \mathcal{X}$, the initial and unsafe regions on the subspace can be redefined as $\mathcal{X}_\mathcal{I} := \mathcal{S} \cap \mathcal{X}_\mathcal{I}$ and $\mathcal{X}_\mathcal{U} := \mathcal{S} \cap \mathcal{X}_\mathcal{U}$, respectively.*

IV. Data-Driven Control Barrier Certificates

The previous section demonstrated how we can consider the projected system (7) in place of the descriptor system (6) by projecting the dynamics onto the algebraic constraints. To design a control policy and synthesize a CBC, we can collect a single input-state trajectory relating to the projected system (7) instead of the descriptor system (6), as the latter does not satisfy the full-row rank condition required for PE. To collect the correct trajectory, we therefore require the following assumptions:

Assumption 3 (Subspace Data Collection)**.** *The single input-state trajectory satisfies the following:*

1) *the initial condition* $\mathbf{z}(0) \in \mathcal{S}$ *(*causal solvability*);*
2) *the trajectory remains in the subspace $\mathcal{S}$;*
3) *the system is* impulse-free*, (system is causal and independent of future inputs).*

These assumptions are not unduly restrictive. The constraints subspace (6b) relates to real phenomena restricting the physical system, *e.g.* Kirchhoff's laws or robot joint overextension, of which the user can have some prior understanding. Assumption 1 is justified by deriving the dictionary from first principles, where spurious additional terms are tolerated. The user has total freedom in how to collect the data trajectory, choosing a consistent initial condition and any input that keeps the trajectory in $\mathcal{S}$. The user can also enforce safety during data collection by selecting known safe input choices. In practice, the "user" need not be human: any pre-existing (uncertified) controller, *e.g.* a legacy stabilizing controller, can generate the initial safe inputs. This is pertinent for fast unstable systems, where manual safe operation is difficult.

We (re)define our collected single trajectory for data collected on the subspace (6b) as follows ($\mathbb{U}_0$ unchanged):

$$\mathbb{Z}_0 = [\mathbf{z}(0), \dots, \mathbf{z}(T-1)] \in \mathbb{R}^{n_d \times T}, \tag{8a}$$
$$\mathbb{D}_0 = [\mathfrak{D}_d(\mathbf{z}(0)), \dots, \mathfrak{D}_d(\mathbf{z}(T-1))] \in \mathbb{R}^{N_d \times T}, \tag{8b}$$
$$\mathbb{Z}_1 = [\mathbf{z}(1), \dots, \mathbf{z}(T)] \in \mathbb{R}^{n_d \times T}, \tag{8c}$$
$$\mathbb{U}_0 = [\mathbf{u}(0), \dots, \mathbf{u}(T-1)] \in \mathbb{R}^{m \times T}. \tag{8d}$$

From (7) it is trivial that $\mathbb{Z}_1 = \tilde{A}\mathbb{D}_0 + \tilde{B}\mathbb{U}_0$.

We will derive the conditions for a data-driven control policy $\mathbf{u} = K(\mathbf{z})\mathbf{z}$, that guarantees safety via a CBC. The condition $\operatorname{rank}(\mathbb{D}_0) = N_d$ is sufficient for the data to satisfy PE and the row rank is simple to calculate. We consider the data parameterization approach of [20] which is more relaxed than [23]. Notably, the collected data could be non-consecutive, but $T \geq N_d$ is a necessary condition for PE.

A. Data-Driven Projected Model

Using the single input-state trajectory (8), we derive a closed-form data-driven model of (7) with control policy $\mathbf{u} = K(\mathbf{z})\mathbf{z} = \mathbb{U}_0 Q(\mathbf{z})\mathbf{z}$.

Theorem 2 (Data-Driven Model)**.** *Let $Q(\mathbf{z}) \in \mathbb{R}^{T \times n_d}$ be a matrix-valued function such that*

$$\Theta_d(\mathbf{z}) = \mathbb{D}_0 Q(\mathbf{z}), \tag{9}$$

where $\Theta_d(\mathbf{z}) \in \mathbb{R}^{N_d \times n_d}$, $\mathfrak{D}_d(\mathbf{z}) = \Theta_d(\mathbf{z})\mathbf{z}$ and $\mathbb{D}_0$ is full row-rank. If one synthesizes $\mathbf{u} = K(\mathbf{z})\mathbf{z} = \mathbb{U}_0 Q(\mathbf{z})\mathbf{z}$, then the closed-loop system $\mathbf{z}^+ = \tilde{A}\mathfrak{D}_d(\mathbf{z}) + \tilde{B}\mathbf{u}$ has the following data-based representation:

$$\mathbf{z}^+ = \mathbb{Z}_1 Q(\mathbf{z})\mathbf{z}, \tag{10}$$

equivalently, $\tilde{A}\Theta_d(\mathbf{z}) + \tilde{B}K(\mathbf{z}) = \mathbb{Z}_1 Q(\mathbf{z})$.

Proof. $\mathbf{z}^+ = \tilde{A}\mathfrak{D}_d(\mathbf{z}) + \tilde{B}\mathbf{u}$ is a projected model exactly representing the descriptor (6) (using Theorem 1). Since $\mathbf{u} = K(\mathbf{z})\mathbf{z} = \mathbb{U}_0 Q(\mathbf{z})\mathbf{z}$, the closed-loop system (7) can be written as

$$\mathbf{z}^+ = (\tilde{A}\Theta_d(\mathbf{z}) + \tilde{B}K(\mathbf{z}))\mathbf{z} = \begin{bmatrix}\tilde{B} & \tilde{A}\end{bmatrix}\begin{bmatrix}K(\mathbf{z}) \\ \Theta_d(\mathbf{z})\end{bmatrix}\mathbf{z}.$$

Leveraging (9), $\mathbf{z}^+ = \begin{bmatrix}\tilde{B} & \tilde{A}\end{bmatrix}\begin{bmatrix}\mathbb{U}_0 \\ \mathbb{D}_0\end{bmatrix} Q(\mathbf{z})\mathbf{z}$. Therefore, as $\mathbb{Z}_1 = \tilde{A}\mathbb{D}_0 + \tilde{B}\mathbb{U}_0$, then, $\mathbf{z}^+ = \mathbb{Z}_1 Q(\mathbf{z})\mathbf{z}$, equivalently, $\tilde{A}\Theta_d(\mathbf{z}) + \tilde{B}K(\mathbf{z}) = \mathbb{Z}_1 Q(\mathbf{z})$, is the data-based representation of the closed loop (7), completing the proof. □

Inaccuracy in the choice of $\mathfrak{D}(\mathbf{x})$ may arise in two ways: *(i)* $\mathfrak{D}(\mathbf{x})$ includes all true basis functions but also spurious ones (our setting), or *(ii)* $\mathfrak{D}(\mathbf{x})$ omits genuine terms. In the first, $\mathfrak{D}(\mathbf{x})$ remains sufficiently rich, although possibly overparameterized, which *only* increases computational cost (more data and decision variables), but the framework remains valid under the inaccuracy. In the second, $\mathfrak{D}(\mathbf{x})$ lacks true dynamic terms, Theorem 2 implies that the data-based model no longer represents the true closed-loop system, making the CBC invalid for the real dynamics. Related

studies [26] have considered this situation of neglected nonlinearities. Limited guarantees can be maintained, though controller performance typically degrades and simulations must be confined near the origin.

B. Safety Guarantees via Control Barrier Certificates

One can guarantee the control policy $\mathbf{u} = K(\mathbf{z})\mathbf{z}$ ensures the true descriptor (6) is always safe by certifying a CBC if the conditions of the following theorem are satisfiable. The PE condition is essential to guarantee the Moore-Penrose pseudoinverse of $\mathbb{D}_0$ exists.

Theorem 3 (Data-Driven CBC for Projected System)**.** *Consider the causal descriptor* (6) *with unknown matrices* $A_{11}, A_{12}, A_{21}, A_{22}$*,* B_1*, and* B_2*, and its data-driven representation* (10)*. Let* $\mathcal{X}_\mathcal{I}, \mathcal{X}_\mathcal{U} \subset \mathcal{X}$ *represent the initial and unsafe regions of the descriptor system, respectively, where* $\mathcal{X}_\mathcal{I} \cap \mathcal{X}_\mathcal{U} = \emptyset$*. Suppose there exist constants* $\beta_1, \beta_2 \in \mathbb{R}_{>0}$*, where* $\beta_2 > \beta_1$*, positive-definite matrix* $P \in \mathbb{R}^{n_d \times n_d}$*, and* $M(\mathbf{z}) = \mathbb{Z}_1 \mathbb{D}_0^\dagger \Theta_d(\mathbf{z})$ *satisfying the following constraints:*

$$\begin{bmatrix} \beta_1 & \mathbf{z}^\top \\ \mathbf{z} & P^{-1} \end{bmatrix} \succeq 0, \qquad \forall \mathbf{z} \in \mathcal{X}_\mathcal{I} \tag{11a}$$

$$\begin{bmatrix} P^{-1} & M(\mathbf{z})P^{-1} \\ P^{-1}M(\mathbf{z})^\top & P^{-1} \end{bmatrix} \succeq 0, \qquad \forall \mathbf{z} \in \mathcal{X}_\mathcal{I} \tag{11b}$$

$$\mathbf{z}^\top P \mathbf{z} \geq \beta_2, \qquad \forall \mathbf{z} \in \mathcal{X}_\mathcal{U} \tag{11c}$$

Then $\mathfrak{B}(\mathbf{z}) = \mathbf{z}^\top P \mathbf{z}$ *is a CBC and* $\mathbf{u} = \mathbb{U}_0 \mathbb{D}_0^\dagger \mathfrak{D}_d(\mathbf{z})$ *is its corresponding safety controller for the collected PE trajectory of the descriptor system* (6)*.*

Proof. Consider Definition 3 with quadratic barrier certificate $\mathfrak{B}(\mathbf{x}) = \mathbf{x}^\top P \mathbf{x}$. Replacing the left-hand side of (3a) and (3c) with the above gives:

$$\begin{aligned} \mathbf{x}^\top P \mathbf{x} &\leq \beta_1, \qquad \forall \mathbf{x} \in \mathcal{X}_\mathcal{I}, \\ \mathbf{x}^\top P \mathbf{x} &\geq \beta_2, \qquad \forall \mathbf{x} \in \mathcal{X}_\mathcal{U}. \end{aligned} \tag{12}$$

From Theorem 1 and Theorem 2, we can directly replace the semi-explicit descriptor system (6) by the data-driven reduced-order model (10). Therefore substitute $\mathbf{x}$ with $\mathbf{z}$ from (10), where $\mathbf{z} = \mathbf{x}_d$. Taking the Schur complement of (12) gives (11a).

Condition (11b) maps to (3b), enforcing the barrier function $\mathfrak{B}(\cdot)$ to be non-increasing. It is equivalent to the Lyapunov stability condition for (10):

$$\begin{aligned} (\mathbf{z}^+)^\top P \mathbf{z}^+ &\leq \mathbf{z}^\top P \mathbf{z}, \\ P^{-1} - P^{-1}Q(\mathbf{z})^\top \mathbb{Z}_1^\top P \mathbb{Z}_1 Q(\mathbf{z}) P^{-1} &\succeq 0, \\ \begin{bmatrix} P^{-1} & \mathbb{Z}_1 Q(\mathbf{z}) P^{-1} \\ P^{-1}(\mathbb{Z}_1 Q(\mathbf{z}))^\top & P^{-1} \end{bmatrix} &\succeq 0, \end{aligned}$$

as $Q(\mathbf{z}) = \mathbb{D}_0^\dagger \Theta_d(\mathbf{z})$, substituting in $M(\mathbf{z})$ completes the proof. □

The results can also be generalized beyond quadratic CBCs to the conditions in Definition 3 replacing $\mathbf{x}$ with $\mathbf{z}$ and $f(\mathbf{z}, \mathbf{u})$ with the data-driven projected system description (10). For fixed convex sets describing $\mathbf{z}$, (11a) and (11b) can be solved as LMIs directly as the conditions become linear in P^{-1}, and the sublevel set β_1 is a decision variable to increase the likelihood of valid CBCs. This formulation enables us to move beyond polynomial-type function restrictions, as demonstrated in our second case study. Although the controller design appears decoupled from the CBC, the user has complete freedom in designing the control policy as the user's selection of inputs for $\mathbb{U}_0$ directly affects $\mathbf{u}(k)$. We elaborate on the non-restrictive nature of these conditions in Section V. In the next subsection, we present an alternate control design that is coupled and applied to stability but applicable for safety.

C. Stability Guarantees via Control Lyapunov Functions

Implicitly we are also studying the design of a control policy for system stability as the CBC condition (3b) is equivalent to the Lyapunov stability condition of a discrete-time control Lyapunov function (CLF).

Corollary 1 (Data-Driven CLF)**.** *Consider the unknown descriptor in* (6)*, with its reduced-order projected system's data-driven representation* (10)*. Suppose there exists matrix* $H(\mathbf{z}) \in \mathbb{R}^{T \times n_d}$ *and positive-definite matrix* $P \in \mathbb{R}^{n_d \times n_d}$*, such that* $Q(\mathbf{z}) = H(\mathbf{z})P$*, and constraints,*

$$\mathbb{D}_0 H(\mathbf{z}) = \Theta_d(\mathbf{z}) P^{-1} \tag{13a}$$

$$\begin{bmatrix} P^{-1} & \mathbb{Z}_1 H(\mathbf{z}) \\ H(\mathbf{z})^\top \mathbb{Z}_1^\top & P^{-1} \end{bmatrix} \succeq 0, \tag{13b}$$

are satisfied. Then $\mathfrak{V}(\mathbf{z}) = \mathbf{z}^\top P \mathbf{z}$ *is a control Lyapunov function (CLF) and* $\mathbf{u} = \mathbb{U}_0 H(\mathbf{z}) P \mathbf{z}$ *is its corresponding stabilizing controller for the unknown descriptor* (6)*.*

V. Online Algorithmic Approach

We will now use these theoretical guarantees algorithmically. We present a novel controller design strategy that signals online when it is safe to transition over from user inputs to autonomous inputs. The user begins fully controlling the system, then, when the *autonomous takeover* signal is given, the user transfers control to the autonomous control policy that guarantees safety from that instance onward. Our algorithm demonstrates this for the control policy in Theorem 3, but it is also possible to substitute (13) for (11b) for the alternative control policy, we mark the differences for the latter using algorithm comments in blue. The online approach is provided in Algorithm 1.

We assume we have access to collect data from the descriptor system (6), and a maximal time horizon $T_{\max}$ is decided in advance that manually terminates the loop. We also assume the initial region $\mathcal{X}_\mathcal{I}$ and unsafe region $\mathcal{X}_\mathcal{U}$ are known, along with the extensive dictionary $\mathfrak{D}_d(\cdot)$. As a useful feature, if no safe controller has been found after several time steps, one can *redefine* $\mathcal{X}_\mathcal{I}$ and $\mathcal{X}_\mathcal{U}$ online at any time step. However, once a safe controller is found these regions can no longer be amended without redesigning the control policy.

The algorithm is run as an iterative loop (Step 3), initialized with the initial state and input at $T = 0$ (Step $1 - 2$). In Step 4 the user must choose a safe control input, since the user is manually controlling the system until safe autonomous takeover is possible. In Step 5 we

Algorithm 1 Autonomous takeover by *safety controller* via CBC using Theorem 3 (Corollary 1)

Require: $\mathcal{X}_\mathcal{I}, \mathcal{X}_\mathcal{U}$, $\mathfrak{D}_d(\cdot)$, *causal* system (6), horizon $T_{\max}$

1: Select $\mathbf{z}(0)$ in $\mathcal{S}$ and impulse-free $\mathbf{u}(0)$
2: Record $\mathbb{U}_0$, $\mathbb{Z}_0$, $\mathbb{D}_0$, $\mathbb{Z}_1$ for $T = 0$
3: **for** $T = 1, \ldots, T_{\max}$ **do**
4: Select safe control input $\mathbf{u}(T)$
5: Extend $\mathbb{U}_0$, $\mathbb{Z}_0$, $\mathbb{D}_0$, $\mathbb{Z}_1$
6: **if** $\text{rank}(\mathbb{D}_0) < N_d$ **then continue**
7: **end if** ▷ Replace
8: Solve (11a) and (11b) for P and β_1 ▷ (13)→(11b)
9: **if** infeasible **then continue**
10: **end if**
11: Optimize for $\beta_2 = \min_{\mathbf{z} \in \mathcal{X}_\mathcal{U}} \mathfrak{B}(\mathbf{z})$
12: **if** $\beta_2 \leq \beta_1$ **then continue**
13: **end if**
14: *Autonomous Takeover!* $\mathbf{u}$ *is safe for all* $k > T$.
15: **return** safe controller $\mathbf{u} = \mathbb{U}_0 \mathbb{D}_0^\dagger \mathfrak{D}_d(\mathbf{z})$ certified by CBC $\mathfrak{B}(\mathbf{z}) = \mathbf{z}^\top P \mathbf{z}$ ▷ Controller $\mathbf{u} = \mathbb{U}_0 H(\mathbf{z}) P \mathbf{z}$
16: **end for**
17: **return** infeasible (no CBC safety certificate found)

collect the data matrices, appending new collected data to the matrices acquired at $T-1$. In Step $6-7$ we check if the data is persistently excited, if not we *continue* (skip the remaining steps and move onto next loop iteration with $T = T+1$). Until at least $T = N_d$ the full-row rank condition will not be satisfied, and so the user will need to be controlling the system manually. In Step $8-10$ we solve for P and β_1, (*e.g.* using semidefinite programming), if the solution is infeasible we *continue*. For the alternate controller design, we can solve for P, $H(\mathbf{z})$ and β_1 by replacing condition (11b) with (13). In Step $11-13$ we optimize for the level set β_2, and if $\beta_2 \leq \beta_1$ we *continue*. Note Steps $8-13$ require separate steps even when solvable as LMIs due to the difficulty of solving for P and P^{-1} simultaneously. By solving β_1 with P^{-1} there is a higher likelihood of finding a valid CBC compared with previous hierarchical methods [14]. Reaching Step 14 guarantees the design of a safety controller for both approaches, where the controllers have slightly different formulations. Reaching this stage allows the user to fully transfer control to the autonomous policy with safety guaranteed for all future time steps. The safe controller and CBC are returned in Step 15. The algorithm terminates if no safe controller was designed within $T_{\max}$ iterations in Step 17. $T_{\max}$ can be selected when the number of decision variables becomes too large, and the algorithm could be reinitialized at any point since the user maintains safe system control. Similarly, a sliding window, could be considered to reduce the computational complexity of finding a safe control policy as long as the PE condition is satisfied.

Each iteration of Algorithm 1 requires a rank test on $\mathbb{D}_0 \in \mathbb{R}^{N_d \times T}$ (Step 6), costing $\mathcal{O}(N_d^2 T)$ operations, followed (once PE holds) by an SDP whose cost scales with the *projected* dimension n_d rather than the full state dimension n. The data horizon T enters through the pseudoinverse

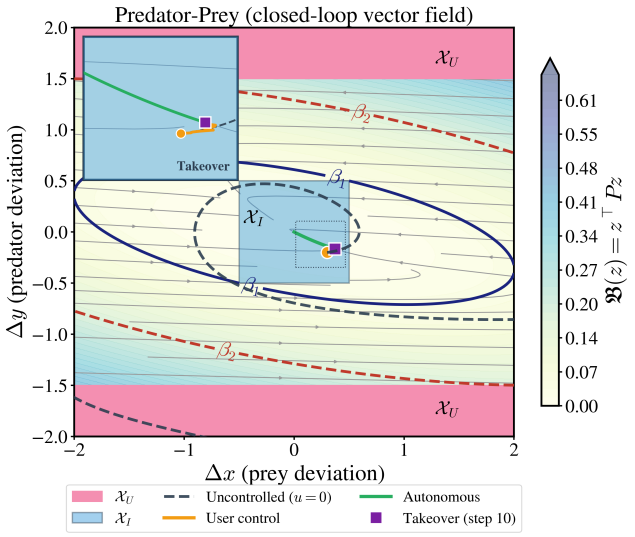

Fig. 1: Demonstration of safe autonomous takeover, the vector field shows the closed-loop system with the autonomous controller applied. Takeover occurs at $k = 10$ (yellow changes to green), with the uncontrolled trajectory (gray-dashed) causing ecological crisis by reaching the unsafe region $\mathcal{X}_\mathcal{U}$ (bottom left corner). A zoom of the dotted area of the figure is displayed in the top left corner.

$\mathbb{D}_0^\dagger$ ($\mathcal{O}(N_d^2 T)$) and, for the alternate design of Corollary 1, through the decision variable $H(\mathbf{z}) \in \mathbb{R}^{T \times n_d}$, motivating the sliding window above for larger systems.

In the same way that the initial region and unsafe region can be updated online when searching for a safe control policy and CBC. It is also possible to expand the extensive dictionary to consider additional nonlinear terms online. By storing $\mathbb{Z}_0$ one can add additional nonlinearities and use $\mathbb{Z}_0$ to build a broader (more extensive dictionary) $\mathbb{D}_0$. Finally, further flexibility in the controller design comes from the decomposition of $\mathfrak{D}_d(\mathbf{z})$, since $\mathfrak{D}_d(\mathbf{z}) = \Theta_d(\mathbf{z})\mathbf{z}$ does not have a unique $\Theta_d(\mathbf{z})$. Overall, the control approach is highly flexible and the conditions to satisfy are relatively lightweight.

VI. Simulation Results

We present two unstable nonlinear case studies, the latter with a trigonometric nonlinear term. We do not consider spurious nonlinearities although they are supported by the theory, *e.g.* see [14]. Both experiments complete online synthesis in under 0.5 s on an Intel Core Ultra 9 285K processor with 64GB RAM.

A. Predator-Prey Ecological Model

We consider the following variation on the famous Lotka-Volterra predator-prey ecological system:

$$x(k+1) = x(k) - h(br_1 y(k) - bx(k)y(k) - u(k)), \quad (14a)$$
$$y(k+1) = y(k) + h(dr_2 x(k) + dx(k)y(k) + q(k)) \quad (14b)$$
$$0 = ex(k) + fy(k) + q(k), \quad (14c)$$

where $x(k)$ and $y(k)$ are the relative prey and predator populations, respectively, and $q(k)$ is a harvesting quota,

introducing an algebraic constraint. This quota introduces greedy predators, when the predators increase relative to the prey then the predators become competitive and cull even more prey. This amplifies each cycle leading to a potential ecological crisis if the predator population deviates from the prey more than ± 1.5 units. The constant terms are; prey birth rate $a = 1.0$, predator birth rate $b = 0.5$, predator death rate $c = 0.8$, predation benefit $d = 0.3$, harvesting sensitivity to prey $e = 0.2$ and harvesting sensitivity to predators $f = -0.3$, $h = 0.05$ is the discretization parameter, and $r_1 = \frac{c}{d}$ and $r_2 = \frac{a}{b}$. The input $u(k)$ is the conservationists intervention on the prey population.

Converting this system into the descriptor form gives $E = \mathrm{diag}(1, 1, 0)$ which is singular. We remind the reader that we consider all coefficients to be unknown in our experiments but assume we know the dictionary $\mathfrak{D}_d(\mathbf{x}_d) = \Theta_d(\mathbf{x}_d)\mathbf{x}_d = [x,\ y,\ xy]^\top$ where $\mathbf{x}_d = [x,\ y]^\top$ and $\mathbf{x}_a = q$. We consider an initial region $\mathcal{X}_\mathcal{I} = [-0.5, 0.5]^2$ and unsafe regions $\mathcal{X}_\mathcal{U} = \{|y| \geq 1.5\}$ (describing ecological crisis).

We collect data according to Assumption 3, where we consider $\mathbf{z}^+ = \tilde{A}\mathfrak{D}_d(\mathbf{z}) + \tilde{B}\mathbf{u}$ to be the system (14a) projected onto its subspace $\mathcal{S}$ (14c), where $\mathbf{z} = \mathbf{x}_d$. Figure 1 demonstrates the results of running the case study. The system is initialized at $\mathbf{z}(0) = [0.3, -0.2]^\top$. The user then selects the following safe inputs (yellow line):

$$\mathbb{U}_0 = [0.286, -0.184, 0.163, 0.474, -0.608, \\ -0.562, 0.383, -0.204, -0.869, -0.253],$$

before the system declares it can complete *autonomous takeover* when $k = 10$ at $\mathbf{z}(10) = [0.372, -0.165]^\top$ marked by the purple box. Based on the user input, the full-row rank condition was satisfied at $k = 10$. The predator-prey system is polynomial so we leverage the well-established S-procedure in step 8 of the algorithm and solve for a feasible P and β_1 via SDP. β_2 is computed simply in step 11 using convex optimization over the semi-algebraic region $\mathcal{X}_\mathcal{U}$. The subsequent autonomous controller (green line) is given by,

$$u(k) = -0.608x(k) - 15.508y(k) + 41.570x(k)y(k)$$

and certified by the CBC,

$$\mathfrak{B}(\mathbf{z}(k)) = 0.014x(k)^2 + 0.0406x(k)y(k) + 0.1119y(k)^2,$$

with level sets $\beta_1 = 0.0416$ and $\beta_2 = 0.1858$. Without human intervention, and the safe autonomous takeover, the system would fall into ecological crisis, shown by the gray-dashed trajectory reaching the unsafe region (bottom left corner).

B. Inverted Pendulum

As our second example, consider an inverted pendulum mounted on a fixed pivot, coupled to a rotary viscous damper whose reaction torque τ_r enters as an algebraic variable:

$$\theta(k+1) = \theta(k) + h\omega(k), \tag{15a}$$
$$J\omega(k+1) = (J - hb)\omega(k) + hmgl\sin(\theta(k)) + h\tau_r(k) + hu(k), \tag{15b}$$
$$0 = c\omega(k) + \tau_r(k), \tag{15c}$$

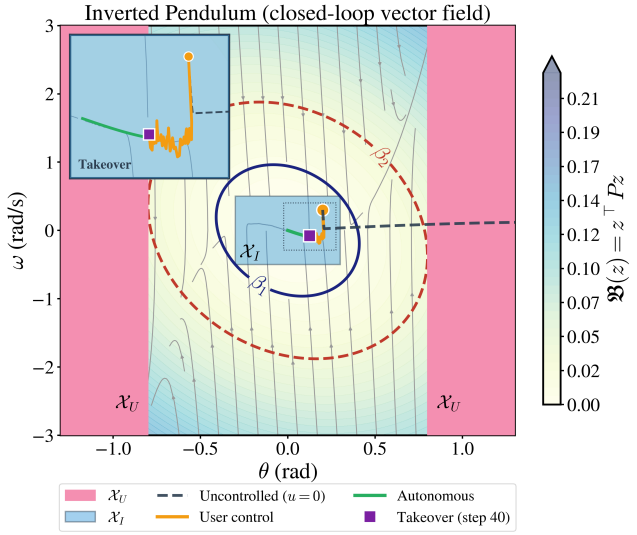

Fig. 2: Demonstration of safe autonomous takeover for the inverted pendulum. The vector field shows the closed-loop system with autonomous controller applied. Takeover occurs at $k = 40$ (yellow changes to green), with the uncontrolled trajectory (gray-dashed) causing the pendulum to fall by reaching the unsafe region $\mathcal{X}_\mathcal{U}$. A zoom of the dotted area of the figure is displayed in the top left corner.

where $\theta(k)$ is the angle from the upright equilibrium and $\omega(k)$ is the angular velocity. The algebraic constraint (15c) models a viscous rotary damper in equilibrium with $\tau_r(k) = -c\omega(k)$. The constant terms are; pendulum mass $m = 1.0$kg, gravitational acceleration $g = 9.81$m/s^2, pendulum length $l = 0.5$m, rotational inertia $J = 0.25$kgm^2, pivot friction $b = 0.5$Nms/rad, and discretization parameter $h = 0.02$. The input $u(k)$ is a control torque applied at the pivot. Without active control, the gravitational torque drives the pendulum to the unsafe region (gray-dashed line in Figure 2). We consider an initial region $\mathcal{X}_\mathcal{I} = [-0.3, 0.3] \times [-0.5, 0.5]$ and unsafe region $\mathcal{X}_\mathcal{U} = \{|\theta| \geq 0.8\}$ (pendulum too far from vertical). Similar to the previous case, the descriptor form here gives $E = \mathrm{diag}(1, J, 0)$ which is singular. We consider the dictionary $\mathfrak{D}_d(\mathbf{x}_d) = [\theta,\ \omega,\ \sin(\theta)]^\top$ where $\mathbf{x}_d = [\theta,\ \omega]^\top$ and $\mathbf{x}_a = \tau_r$.

The inverted pendulum dictionary contains the nonconvex transcendental function $\sin(\theta)$, which prevents a direct SDP formulation, but it can be factorized as $\sin(\theta) = \mathrm{sinc}(\theta)\theta$. As $\mathrm{sinc}(\theta)$ is concave over the domain, if the LMI evaluated at both end-points is non-negative then the interval is nonnegative everywhere; $f(s) \geq 0$ on $[a, b]$ if and only if $f(a) \geq 0$ and $f(b) \geq 0$. This statement is trivial but underutilized for CBCs. Thus $s = \mathrm{sinc}(\theta) \in [s_{\min}, 1]$ with $s_{\min} = \sin(0.3)/0.3 \approx 0.985$. Substituting $F = \mathbb{Z}_1\mathbb{D}_0^\dagger$ and $Q = P^{-1}$ in (11b) gives an LMI in Q, where every entry is linear in Q for fixed $\mathbf{z}$. Exploiting the structure of $\Theta_d(\mathbf{z})$, and the vector $F\Theta_d(\mathbf{z}) = C_0 + sC_1$ where $s = \mathrm{sinc}(\theta)$:

$$\begin{bmatrix} Q & (C_0 + sC_1)Q \\ Q(C_0 + sC_1)^\top & Q \end{bmatrix} \succeq 0, \quad \forall s \in [s_{\min}, 1].$$

No Taylor approximation is required. The guarantees are robust and the Lyapunov decrease (11b) holds for all $\mathbf{z} \in \mathcal{X}_\mathcal{I}$.

We calculate β_1 and β_2 using the same approach as for predator-prey.

Collecting the data according to Assumption 3, we consider $\mathbf{z}^+ = \tilde{A}\mathfrak{D}_d(\mathbf{z}) + \tilde{B}\mathbf{u}$ to be the system (15) projected onto its subspace $\mathcal{S}$ (15c), where $\mathbf{z} = \mathbf{x}_d$. Figure 2 demonstrates the results of running the case study. The system is initialized at $\mathbf{z}(0) = [0.2, 0.3]^\top$. The user then selects the following safe inputs (yellow line):

$$\begin{aligned}\mathbb{U}_0 = [&-4.744,\ -5.950,\ -4.306,\ -2.510,\ -6.005,\ -5.934,\\ &-2.238,\ -3.847,\ -6.280,\ -4.181,\ -6.149,\ -6.081,\\ &-4.594,\ -8.855,\ -8.365,\ -5.937,\ -6.770,\ -4.034,\\ &-6.436,\ -7.366,\ -1.519,\ -4.896,\ -4.251,\ -7.188,\\ &-5.336,\ -3.964,\ -6.445,\ -3.311,\ -5.230,\ -4.549,\\ &-5.117,\ -0.147,\ -3.890,\ -5.932,\ -2.097,\ -6.167,\\ &-3.229,\ -7.533,\ -6.172,\ -3.046],\end{aligned}$$

before the system declares it can complete *autonomous takeover* when $k = 40$ at $\mathbf{z}(40) = [0.126, -0.077]^\top$ marked by the purple box. The PE condition was satisfied at $k = 10$, but Step 8 of Algorithm 1 remained infeasible until $k = 40$ as additional data was required to sufficiently capture the nonlinear dynamics. The subsequent safe autonomous controller (green line) is given by,

$$u(k) = 1919.1\,\theta(k) - 0.031\,\omega(k) - 1958.4\,\sin(\theta(k)),$$

and certified by the CBC,

$$\mathfrak{B}(\mathbf{z}) = 0.0581\,\theta(k)^2 + 0.0100\,\theta(k)\omega(k) + 0.0105\,\omega(k)^2,$$

with level sets $\beta_1 = 0.0094$ and $\beta_2 = 0.0356$. Without human intervention and the safe autonomous takeover, the uncontrolled system diverges into the unsafe region within seconds as the pendulum falls over (grey-dashed trajectory in Figure 2).

VII. Conclusion

In this work we provide the first design of a formal safety controller for an unknown discrete-time nonlinear descriptor system certified by a data-driven control barrier certificate (CBC). We present an online algorithmic approach, including two controller designs, enabling *autonomous takeover* (safe transfer of control) from a user to an autonomous control policy guaranteeing safety. We demonstrate our approach on unstable descriptor systems, including an inverted pendulum with a trigonometric nonlinear term (without needing approximation methods).

References

[1] A. A. Belov, O. G. Andrianova, and A. P. Kurdyukov, *Control of discrete-time descriptor systems: An anisotropy-based approach*. Springer, 2018.

[2] L. Dai, *Singular control systems*. Springer, 1989.

[3] P. Kunkel and V. Mehrmann, *Differential-algebraic equations*. EMS Press, 2024.

[4] S. Campbell, A. Ilchmann, V. Mehrmann, and T. Reis, *Applications of differential-algebraic equations: examples and benchmarks*. Springer, 2019.

[5] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, 2007.

[6] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation." in *Robotics: Science and Systems*, vol. 13. Cambridge, MA, USA, 2017, pp. 1–10.

[7] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *2019 IEEE 58th conference on decision and control (CDC)*. IEEE, 2019, pp. 474–479.

[8] R. Pedersen, C. Sloth, and R. Wisniewski, "Verification of power grid voltage constraint satisfaction—A barrier certificate approach," in *2016 European Control Conference (ECC)*. IEEE, 2016, pp. 447–452.

[9] S. Kundu, S. Geng, S. P. Nandanoori, I. A. Hiskens, and K. Kalsi, "Distributed barrier certificates for safe operation of inverter-based microgrids," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 1042–1047.

[10] M. Althoff and B. H. Krogh, "Reachability analysis of nonlinear differential-algebraic systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 371–383, 2013.

[11] M. Althoff, "Formal and compositional analysis of power systems using reachable sets," *IEEE Transactions on Power Systems*, vol. 29, no. 5, pp. 2270–2280, 2014.

[12] Y. Park and C. Sloth, "Differential-Algebraic Equation Control Barrier Function for Flexible Link Manipulator," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 12 408–12 413.

[13] H. Zhang, M. H. Kazma, M. Ma, T. T. Johnson, and A. F. Taha, "Verification and forward invariance of control barrier functions for differential-algebraic systems," *arXiv preprint arXiv:2603.13509*, 2026.

[14] B. Samari, O. Akbarzadeh, M. Zaker, and A. Lavaei, "From a single trajectory to safety controller synthesis of discrete-time nonlinear polynomial systems," *IEEE Control Systems Letters*, 2024.

[15] B. Wooding and A. Lavaei, "Learning k-Inductive Control Barrier Certificates for Unknown Nonlinear Dynamics Beyond Polynomials," *arXiv preprint arXiv:2412.07232*, 2024.

[16] B. Wooding, H. Zhang, T. T. Johnson, and A. Lavaei, "k-inductive neural barrier certificates for unknown nonlinear dynamics," in *3rd International Conference on Neuro-symbolic Systems (NeuS)*, 2026, to appear. arXiv preprint arXiv:2605.20108.

[17] A. Edwards, A. Peruffo, and A. Abate, "Fossil 2.0: Formal certificate synthesis for the verification and control of dynamical models," in *Proceedings of the 27th ACM International Conference on Hybrid Systems: Computation and Control*, 2024, pp. 1–10.

[18] J. Gardner, B. Wooding, A. Nejati, and A. Lavaei, "TRUST: Stability and safety controller synthesis for unknown dynamical models using a single trajectory," in *Proceedings of the 28th ACM International Conference on Hybrid Systems: Computation and Control*, 2025, pp. 1–16.

[19] V. Murali, A. Trivedi, and M. Zamani, "A scenario approach for synthesizing k-inductive barrier certificates," *IEEE Control Systems Letters*, vol. 6, pp. 3247–3252, 2022.

[20] 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.

[21] P. Schmitz, T. Faulwasser, and K. Worthmann, "Willems' fundamental lemma for linear descriptor systems and its use for data-driven output-feedback MPC," *IEEE Control Systems Letters*, vol. 6, pp. 2443–2448, 2022.

[22] C. Verhoek, R. Tóth, S. Haesaert, and A. Koch, "Fundamental lemma for data-driven analysis of linear parameter-varying systems," in *2021 60th IEEE conference on decision and control (CDC)*. IEEE, 2021, pp. 5040–5046.

[23] 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.

[24] Y. Zhang, Y. Wang, J. Shang, and J. Zhang, "Data-driven analysis and predictive control of descriptor systems with application to power and water networks," *arXiv preprint arXiv:2508.16091*, 2025.

[25] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[26] C. De Persis, M. Rotulo, and P. Tesi, "Learning controllers from data via approximate nonlinearity cancellation," *IEEE Transactions on Automatic Control*, vol. 68, no. 10, pp. 6082–6097, 2023.

[27] T. Groß, S. Trenn, and A. Wirsen, "Solvability and stability of a power system DAE model," *Systems & Control Letters*, vol. 97, pp. 12–17, 2016.