Automaton-based Implicit Controlled Invariant Set Computation for Discrete-Time Linear Systems

Zexiang Liu Tzanis Anevlavis Necmiye Ozay Paulo Tabuada

***Abstract*— In this paper, we derive closed-form expressions for implicit controlled invariant sets for discrete-time controllable linear systems with measurable disturbances. In particular, a disturbance-reactive (or disturbance feedback) controller in the form of a parameterized finite automaton is considered. We show that, for a class of automata, the robust positively invariant sets of the corresponding closed-loop systems can be expressed by a set of linear inequality constraints in the joint space of system states and controller parameters. This leads to an implicit representation of the invariant set in a lifted space. We further show how the same parameterization can be used to compute invariant sets when the disturbance is not available for measurement.**

I. Introduction

When tasked with synthesizing a controller in order to ensure safety of a plant under uncertainties, the objective is to indefinitely keep the state of the plant within a set of safe states. A natural solution to this task is to initialize the state in a Robust Controlled Invariant Set (RCIS) within the set of safe states. RCISs have the property that any trajectory starting within an RCIS can always be forced to remain inside the RCIS and, therefore, inside the set of safe states. Consequently, RCISs are at the core of safety-critical applications.

Since the conception of the standard method for computing the Maximal RCIS of discrete-time systems [1], [2], is known to suffer from poor scaling with the system's dimension and no guarantees of termination, different approaches attempted to alleviate these drawbacks. In the case of bounded disturbances, when the set of safe states are polytopes, [3] computes inner and outer approximations of RCISs for linear systems with guarantees on finite-time termination. For the same system class, a different line of works [4]–[6] approximate the Maximal/Minimal RCIS by first closing the loop with a linear state feedback law and then computing the Robust Positively Invariant Set (RPIS) of the closed-loop system. This group of methods are typically very conservative since only linear state feedback controllers are considered.

Finally, several recent methods [7]–[10], including previous works from the authors, develop approaches for constructing implicit RCIS in closed form, represented as polytopes in a high-dimensional space whose projection onto the original state space is an RCIS. By avoiding computing RCISs explicitly, those methods can work for systems with higher dimensions. It is indeed the case in many practical applications, such as model predictive control and supervision control, that knowledge of the explicit RCIS is not required and the implicit representation suffices [11], [12].

Inspired by the recent progress on implicit RCISs, in this work we propose a novel approach to compute implicit RCISs for discrete-time linear systems. In addition, the aforementioned works consider non-measurable disturbances only, however, in many safety-critical applications, incoming disturbances can be measured in ahead [13] and, hence, are considered measurable [2]. Thus, unlike the existing works, we develop a method that works for both measurable and non-measurable disturbances, achieved by introducing an automaton-based controller whose input is exactly the measurable disturbance. More specifically, our contributions are as follows:

1) We propose an automaton-based method for computing *implicit RCISs* for a class of linear systems that contains the class of controllable linear systems, with measurable disturbances.

2) We derive conditions on the structure of the automaton such that the implicit RCIS is computed in closed-form.

3) We present a generic connection between measurable and non-measurable disturbances, enabling the proposed method to work with systems with non-measurable disturbances.

In addition to the above, we demonstrate the practicality of the implicit RCIS in the task of supervision control for the lane keeping problem. The goal is to modify nominal control inputs, as needed, to keep the system's trajectory within a set of safe states. We show that this is achieved by solving an optimization problem using the implicit RCIS.

The paper is organized as follows: In Section II, the problem is mathematically set up, along with the essential definitions and assumptions. Section III lays down the ideas for computing an implicit RCIS for systems with measurable disturbances. Subsequently, Section IV investigates when the implicit RCIS can be computed in closed-form, while Section V connects the proposed method to the case of non-measurable disturbances. Section VI provides a computational evaluation of the proposed method. Finally, conclusion is found in Section VII. To keep a streamlined presentation, the proofs of all theorems are found in Appendix.

Notation: The Minkowski sum of two sets A and B is

Z. Liu and N. Ozay are with the Dept. of Electrical Engineering and Computer Science, Univ. of Michigan, Ann Arbor, MI 48109, USA {zexiang, necmiye} @umich.edu. T. Anevlavis and P. Tabuada are with the UCLA Electrical and Computer Engineering Department, Los Angeles, CA 90095 {janis10, tabuada} @ucla.edu.

This work was partially supported by the NSF grants CNS-1931982 and CCF-1918123 and the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

denoted by $A+B=\{a+b \mid a\in A, b\in B\}$. For a singleton $\{x\}$, we write the Minkowski sum $\{x\}+B$ as $x+B$. We denote the convex hull of a set A by $\mathbf{CH}(A)$. For a sequence $(d(t))_{t=0}^{N}$, we denote the finite subsequence $(d(t))_{t=a}^{b}$ by $d(a:b)$ with $0\le a\le b\le N$. We denote the projection of a set C in $\mathbb{R}^{n+m}$ onto the first n coordinates by $\mathrm{Proj}_{1:n}(C)$ in $\mathbb{R}^n$. For two vectors $x\in\mathbb{R}^n$ and $y\in\mathbb{R}^m$, (x,y) denotes the concatenated vector in $\mathbb{R}^{n+m}$.

II. Preliminaries

In this work, we consider a discrete-time linear system Σ:

$$\Sigma : x(t+1) = Ax(t) + Bu(t) + d(t), \tag{1}$$

where $x\in\mathbb{R}^n$ is the state of Σ, $u\in\mathbb{R}^m$ is the input, and $d\in D\subseteq\mathbb{R}^n$ is a disturbance term. The set D contains all possible values of d. The disturbance d is *measurable* if $u(t)$ can be determined based on the measurement $d(t)$; otherwise, d is *non-measurable*.

Definition 1 (Safe set)**.** Let $S\subseteq\mathbb{R}^{n+m}$ be the set of desired state and input pairs (x,u), called the *safe set* of the system. That is, we want $(x(t),u(t))\in S$ for all $t\ge 0$.

The difference between measurable and non-measurable disturbances is reflected in the following definitions of a Robust Controlled Invariant Set (RCIS).

Definition 2 (RCIS with non-measurable disturbance)**.** Consider a *non-measurable* disturbance d. A set $C\subseteq\mathbb{R}^n$ is an RCIS for the system Σ within the safe set S if:

$$\forall x\in C, \exists u \text{ such that } (x,u)\in S, Ax+Bu+d\in C, \forall d\in D. \tag{2}$$

Definition 3 (RCIS with measurable disturbance)**.** For the disturbance d being *measurable*, a set $C\subseteq\mathbb{R}^n$ is an RCIS for the system Σ within the safe set S if:

$$\forall x\in C, \forall d\in D, \exists u \text{ such that } (x,u)\in S, Ax+Bu+d\in C. \tag{3}$$

Note that the order of the quantifiers $\forall d\in D$ and $\exists u$ is swapped in the definitions above, which means an RCIS C with respect to a non-measurable disturbance is guaranteed to be an RCIS for the same system with measurable disturbance but not vice versa. Finally, we call a set C_{max} the *Maximal RCIS* within S if it is controlled invariant and contains every RCIS in S.

Next, consider an autonomous system Σ_a:

$$\Sigma_a : x(t+1) = f(x(t), d(t)) \tag{4}$$

with state $x\in\mathbb{R}^n$ and a disturbance term $d\in D$.

Definition 4 (Robust Positively Invariant Set (RPIS))**.** A set $C\subseteq S$ is an *RPIS* of the autonomous system Σ_a within S if:

$$\forall x\in C \text{ we have that } f(x,d)\in C, \forall d\in D. \tag{5}$$

A set C_{max} is the *Maximal RPIS* within S if it is positively invariant and contains every robust positively invariant set in S.

Notice that, compared to systems with control inputs, there is no concept of measurable or non-measurable disturbances for autonomous systems.

Definition 5 (Reachable set for autonomous systems)**.** Let Σ_a be an autonomous system. The *reachable set* $\mathscr{R}(\Sigma_a, x_0)$ of Σ_a from state x_0 is the set of all possible states that the system may visit. Formally, $x\in\mathscr{R}(\Sigma_a,x_0)$ if and only if there exists a trajectory $(x(t))_{t=0}^{k}$, for some $k\ge 0$, of Σ_a under disturbance sequence $(d(t))_{t=0}^{k-1}\in D^k$ with $x(0)=x_0$ and $x(k)=x$.

Proposition 1. *The set $C=\{x\in\mathbb{R}^n \mid \mathscr{R}(\Sigma_a,x)\subseteq S\}$, i.e., the set of states whose corresponding reachable set is contained in the safe set S, is the Maximal RPIS for Σ_a within S.*

A. Problem Setup

In the first part of this work, we focus on the computation of RCISs for systems with measurable disturbances. More specifically, we propose a method that computes the desired implicit representation for an RCIS in closed-form based on the following assumptions. In the second part, we extend our method to compute RCISs for non-measurable disturbances.

Assumption 1: The matrix A is nilpotent. That is, there exists a non-negative integer $h\le n$ such that $A^h=0$.

Remark 1. Assumption 1 is satisfied by any controllable system, as there always exists a feedback gain K such that $A+BK$ is nilpotent. Thus, any controllable system satisfies Assumption 1 by pre-feedbacking the system with $u=Kx+v$ and taking v as the new control input[1] [14, Ch.3].

Assumption 2: The safe set $S\subset\mathbb{R}^{n+m}$ and the disturbance set $D\subset\mathbb{R}^n$ are both polytopes.

The next theorem shows that to compute an RCIS for systems with a measurable disturbance, we only need to consider the (finite) vertices of D.

Proposition 2. *Under Assumption 2, consider a system Σ with a measurable disturbance $d\in D$, and let D_v be the set of vertices of D. Let Σ' be the same system as Σ but with the measurable disturbance $d\in D_v$. Then, a convex set C is an RCIS for Σ if and only if C is an RCIS for Σ'.*

According to Proposition 2, given any polytopic disturbance set D, we can substitute D by the finite set D_v without loss of generality. Thus, for the remaining of this paper, we directly assume that D is a finite set, as stated below.

Assumption 3: The disturbance set $D\subset\mathbb{R}^n$ is given as a finite set of vertices.

Problem 1: For a linear system Σ, a safe set S, and a disturbance set D satisfying Assumptions 1, 2 and 3, compute a convex implicit RCIS of Σ within S in closed-form.

III. Controlled Invariant Set Computation Framework

The Maximal RCIS for a system in (1) with measurable and/or non-measurable disturbances can be computed by a standard iterative method ([1], [2]), which is not guaranteed to terminate in finite time and does not scale to high-dimensional systems. To reduce the computation burden, many existing works close the loop with a linear feedback controller and then compute an RPIS of the closed-loop

[1] Accordingly, given the safe set S_{xu} for (x,u), the safe set S_{xv} for the new state-input pair (x,v) should be $S_{xv}=\{(x,v) \mid (x,Kx+v)\in S_{xu}\}$.

system as an under-approximation of the Maximal RCIS. In this section, we extend this idea to a more general case: We close the loop with a parameterized nonlinear disturbance-feedback controller. Then, by computing an RPIS of an augmented closed-loop system, we search for the feasible initial states and controller parameters simultaneously such that the closed-loop trajectory satisfies the safety constraints. Lastly, we retrieve an RCIS from the RPIS of the augmented system.

First, we want to determine an appropriate controller structure. We draw some inspirations from Definition 3: Given any RCIS C for Σ under measurable disturbances, by definition, there exists a *memoryless state-disturbance feedback controller* $u(t) = k(x(t), d(t))$ such that C is the Maximal RPIS of the closed-loop system. In other words, any RCIS, including the Maximal RCIS, is the Maximal RPIS of a closed-loop system with respect to some memoryless state-disturbance controller. Thus, to minimize the conservativeness of the closed-loop RPIS, it is enough to consider the class of memoryless state-disturbance feedback controllers. Furthermore, it is well-known that any memoryless state-disturbance feedback controller is equivalent to a *disturbance feedback controller with memory*, explained by the following example.

Example 1. Consider a memoryless state-disturbance feedback controller $u(t) = \kappa(x(t), d(t))$. This controller can be equivalently expressed as:

$$\begin{cases} s(t+1) &= As(t) + B\kappa(s(t), d(t)) + d(t), \\ u(t) &= \kappa(s(t), d(t)), \end{cases} \tag{6}$$

with $s(0) = x(0)$. That is, the internal dynamics of the controller forms a state estimator. ■

For above reasons, we consider a *parameterized disturbance-feedback controller* Σ_c *with memory*:

$$\Sigma_c : \begin{cases} s(t+1) &= \mathscr{T}(s(t), d(t); \theta), \\ u(t) &= o(s(t), d(t); \theta). \end{cases} \tag{7}$$

In the above, s is the internal state (memory) of the controller that distills useful information from the disturbance input $d \in D$, and u is the output of the controller. The same d and u correspond to the disturbance and the control input of Σ respectively. The state transition function $\mathscr{T}$ and the output function o map the current state $s(t)$ and the disturbance input $d(t)$ into the next state $s(t+1)$ and the current output $u(t)$ respectively. Finally, θ is a constant vector that parameterizes the state transition function $\mathscr{T}$ and the output function o. The value of θ can depend on the initial state x_0 of the system Σ, such as in Example 1, $\theta = x(0)$. In what follows, we assume that the functions $\mathscr{T}$ and o in Σ_c are known. We discuss how to select $\mathscr{T}$ and o in the next section.

Closing the loop of Σ in (1) with the controller in (7), we obtain the following closed-loop system augmented with the controller internal state s and the constant vector θ:

$$\Sigma_{cl} : \begin{bmatrix} x(t+1) \\ \theta(t+1) \\ s(t+1) \end{bmatrix} = \begin{bmatrix} Ax(t) + Bo(s(t), d(t); \theta(t)) + d(t) \\ \theta(t) \\ \mathscr{T}(s(t), d(t); \theta(t)) \end{bmatrix}, \tag{8}$$

with the augmented state (x, θ, s) and the disturbance $d \in D$.

In the above augmented system, we can calculate feasible initial states x_0 and controller parameters θ simultaneously. The control input $u(t)$ of Σ is equal to $o(x(t), d(t); \theta(t))$, as a function of the augmented state. Then, given the safe set S of the system Σ, we define the safe set S_{cl} for the closed-loop system Σ_{cl} by:

$$S_{cl} = \{(x, \theta, s) \mid (x, o(s, d; \theta)) \in S, \forall d \in D)\}. \tag{9}$$

The following theorem connects the problem of computing an RCIS for the system Σ with the problem of computing an RPIS for the closed-loop system Σ_{cl}.

Theorem 1. *Let C_{cl} be an RPIS for the closed-loop system Σ_{cl} within the safe set S_{cl}. Then, the convex hull $\boldsymbol{CH}(Proj_{1:n}(C_{cl}))$ of the projection of C_{cl} onto the first n coordinates is a convex RCIS for the system Σ within the safe set S.*

According to Theorem 1, we propose a novel framework for computing RCISs: Given a controller specified by functions $\mathscr{T}$ and o, we first compute the Maximal RPIS C_{cl} of the augmented system Σ_{cl} within S_{cl}, and then take the convex hull of the projection of C_{cl}.

The size of the resulting RCIS depends on the choice of functions $\mathscr{T}$ and o. In Section VI-A, we demonstrate two classes of functions $\mathscr{T}$ and o that lead to larger RCISs as the number of parameters increases. In theory, there exist functions $\mathscr{T}$ and o such that the resulting RCIS meets the Maximal RCIS. However, how to find $\mathscr{T}$ and o achieving the Maximal RCIS is beyond the scope of this paper. One significant advantage of this framework, explored in the next section, is that by carefully designing $\mathscr{T}$ and o, this novel framework enables closed-form construction of RCISs.

IV. Closed-form construction of implicit robust controlled invariant sets

In the previous section, we presented a framework for computing RCISs. Still, there are two main questions to be addressed. First, can we compute the Maximal RPIS C_{cl} efficiently? Second, in practice the convex hull computation is expensive, can we avoid this operation? In this section, we show that by carefully designing $\mathscr{T}$ and o, we address both questions.

A. Computing C_{cl} in closed-form

From Proposition 1, we have that:

$$C_{cl} = \{(x, \theta, s) \mid \mathscr{R}(\Sigma_{cl}, (x, \theta, s)) \subseteq S_{cl}\}. \tag{10}$$

According to (10), if we express the reachable set $\mathscr{R}(\Sigma_{cl}, (x, \theta, s))$ in closed-form, then we obtain a closed-form expression of C_{cl}. The next theorem gives a sufficient

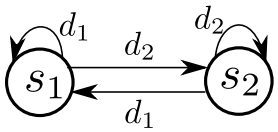

Fig. 1: A toy mealy machine controller.

condition for the reachable set $\mathscr{R}(\Sigma_{cl})$ to admit a closed-form expression.

Theorem 2. *Under Assumption 1 and 3, if the state s of the controller belongs to a finite set Q, then given any initial state (x,θ,s), the reachable set $\mathscr{R}(\Sigma_{cl},(x,\theta,s))$ is finite. Moreover, it can be expressed in closed-form.*

Note that Theorem 2 is the only result that requires Assumption 1 in this work. Given Theorem 2, we want to design the functions $\mathscr{T}$ and o such that the internal state s of the controller belongs to a finite set Q. Recall that by Proposition 2 and Assumption 3, the input d of the controller also belongs to a finite set D. Thus, the controller Σ_c is a system with finite states and inputs. In the literature, this type of system is called a *mealy machine*, a special class of automata.

Definition 6 (Mealy Machine)**.** A *Mealy Machine* Σ_{fts} is a quintuple $(Q,D,\mathscr{T},\Theta,o)$, where:

- Q is a finite set of discrete states;
- D is a finite set of actions;
- $\mathscr{T}: Q\times D\to Q$ is the state transition function that maps each state-action pair to the next state;
- Θ is a finite set of outputs;
- $o: Q\times D\to\Theta$ is the output function that maps each state-action pair to an element in the set Θ.

With slightly abusing notations, we denote both the action set of a mealy machine and the disturbance set of Σ by D, since in this work we only consider the disturbance set D as the action set. The transition function $\mathscr{T}$ and the output function o are designed by the user. We parameterize the output set Θ of a mealy machine by a parameter vector θ: Suppose that $\Theta=\{u_i\}_{i=1}^L$, where each u_i is a vector of variables in $\mathbb{R}^m$. Given a vector $\theta=(\overline{u}_1,\cdots,\overline{u}_L)\in\mathbb{R}^{mL}$, we define a parameterized output function $o(s,d;\theta): Q\times D\to\mathbb{R}^m$ such that $o(s,d;\theta)=\overline{u}_i$ for $o(s,d)=u_i$. A simple example of a mealy machine and the parameterized output function is shown below.

Example 2. Let $D=\{d_1,d_2\}\subset\mathbb{R}^n$, and $Q=\{s_1,s_2\}$. The state transition function $\mathscr{T}$ is shown in Fig. 1. The output set is $\Theta=\{u_1,u_2\}$. The output function is $o(s_1,d_1)=o(s_2,d_1)=u_1$ and $o(s_1,d_2)=o(s_2,d_2)=u_2$. Let the controller parameter be $\theta=(\overline{u}_1,\overline{u}_2)=(1,2)$. Then, the parameterized output function is $o(s_1,d_1;\theta)=o(s_2,d_1;\theta)=1$ and $o(s_1,d_2;\theta)=o(s_2,d_2;\theta)=2$. ■

Next, we provide guidance on how to construct C_{cl} efficiently. Since $s\in Q$, with Q finite, we can decompose C_{cl} into $|Q|$ subsets:

$$C_{cl}=\bigcup_{s_i\in Q}C_{sub}(s_i)\times\{s_i\}, \tag{11}$$

where:

$$C_{sub}(s_i)=\left\{(x,\theta)\in\mathbb{R}^{n+mL}\mid\mathscr{R}(\Sigma_{cl},(x,\theta,s_i))\subseteq S_{cl}\right\}. \tag{12}$$

For each state $(x',\theta,s')\in\mathscr{R}(\Sigma_{cl},(x,\theta,s_i))$:

$$(x',\theta,s')\in S_{cl}\Leftrightarrow(x',o(s',d;\theta))\in S,\ \forall d\in D. \tag{13}$$

As x' and $o(s',d;\theta)$ are linear functions of x, θ, and $d\in D$, the condition $(x',o(s',d;\theta))\in S$ is a set of linear inequality constraints on (x,θ). Thus, by (13) and the fact that $\mathscr{R}(\Sigma_{cl},(x,\theta,s_i))$ is finite, the set $C_{sub}(s_i)$ in (12) can be expressed by a set of linear inequality constraints, that is a polytope in $\mathbb{R}^{n+mL}$. We use the following example to illustrate the computation of $C_{sub}(s_i)$.

Example 3. Consider the mealy machine controller in Example 2. Suppose that the nilpotent matrix A of system Σ satisfies $A^2=0$. Then, the reachable set $\mathscr{R}(\Sigma_{cl},(x,\theta,s_i))$ contains 7 elements, that is

$$\mathscr{R}(\Sigma_{cl},(x,\theta,s_i))=\left\{\bigcup_{j,k=1,2}(ABu_j+Bu_k+Ad_j+d_k,\theta,s_k)\right\}$$
$$\cup\left\{(x,\theta,s_i)\right\}\cup\left\{\bigcup_{j=1,2}(Ax+Bu_j+d_j,\theta,s_j)\right\},$$

where $\theta=(u_1,u_2)$. Suppose that the safe set is $S=\{(x,u)\in\mathbb{R}^{n+m}\mid G_xx+G_uu\le h\}$. Then:

$$\begin{aligned}C_{sub}(s_1)=C_{sub}(s_2)=&\\ \{(x,u)\in\mathbb{R}^{n+m}\mid&G_xx+G_uu_i\le h,\\ &G_x(Ax+Bu_j+d_j)+G_uu_j\le h,\\ &G_x(ABu_j+Bu_k+Ad_j+d_k)+G_uu_i\le h,\\ &\forall i,j,k\in\{1,2\}\}.\end{aligned}$$

Finally, $C_{cl}=\bigcup_{i=1,2}C_{sub}(s_i)\times\{s_i\}$.

So far we constructed C_{cl} in closed-form. However, to obtain an RCIS from C_{cl}, we have to project C_{cl} onto the first n coordinates and then compute the convex hull of the projected set. Both projection and convex hull operations are time consuming and thus undesirable. In what follows we derive an *implicit expression* of the resulting RCISs.

B. Implicit Controlled Invariant Set Expression (Method 1)

Assumption 4: The safe set S of Σ is bounded.

Given that S is bounded, the projection of $C_{sub}(s_i)$ onto the first n coordinates is also bounded. Thus, we can always find a large enough hyperbox B such that:

$$\mathrm{Proj}_{1:n}(B\cap C_{sub}(s_i))=\mathrm{Proj}_{1:n}(C_{sub}(s_i)).$$

Denote the intersection of the hyperbox B and the polytope $C_{sub}(s_i)$ by $\overline{C}_{sub}(s_i)=B\cap C_{sub}(s_i)$. The projection of C_{cl} is

exactly the union of the projections of polytopes $\overline{C}_{sub}(s_i)$ over $s_i \in Q$, that is:

$$\text{Proj}_{1:n}(C_{cl}) = \bigcup_{s_i \in Q} \text{Proj}_{1:n}\left(\overline{C}_{sub}(s_i)\right). \tag{14}$$

Since the order of convex hull operation and the projection can be swapped, we have that:

$$\mathbf{CH}(\text{Proj}_{1:n}(C_{cl})) = \text{Proj}_{1:n}\left(\mathbf{CH}\left(\bigcup_{s_i \in Q} \overline{C}_{sub}(s_i)\right)\right). \tag{15}$$

Since $\overline{C}_{sub}(s_i)$ is a polytope, it can be written as:

$$C_{sub}(s_i) = \{(x,\theta) \mid G_i(x,\theta) \leq h_i\}.$$

Then, we construct the polytope:

$$\begin{aligned} C_\lambda = \Big\{ & \left(x, \theta, x_1, \theta_1, \cdots, x_{|Q|}, \theta_{|Q|}, \lambda_1, \cdots, \lambda_{|Q|}\right) \mid \\ & \lambda_i \geq 0, G_i(x_i,\theta_i) \leq \lambda_i h_i, \forall 1 \leq i \leq |Q|, \\ & \sum_{i=1}^{|Q|} \lambda_i = 1, \sum_{i=1}^{|Q|} x_i = x, \sum_{i=1}^{|Q|} \theta_i = \theta \Big\}. \end{aligned} \tag{16}$$

Under Assumption 4, given that $(x,\theta) \in R^{n+mL}$, we have:

$$\mathbf{CH}\left(\bigcup_{s_i \in Q} \overline{C}_{sub}(s_i)\right) = \text{Proj}_{1:(n+mL)}\left(C_\lambda\right), \tag{17}$$

$$\text{Proj}_{1:n}\left(\mathbf{CH}\left(\bigcup_{s_i \in Q} \overline{C}_{sub}(s_i)\right)\right) = \text{Proj}_{1:n}(C_\lambda). \tag{18}$$

By (15) and (18), the RCIS $\mathbf{CH}(\text{Proj}_{1:n}(C_{cl}))$ is the projection of C_λ onto the first n coordinates. In other words, C_λ is an *implicit expression* of the RCIS $\mathbf{CH}(\text{Proj}_{1:n}(C_{cl}))$.

Remark 2. Assumption 4 is only required if we want the equality $\text{Proj}_{1:n}(C_\lambda) = \mathbf{CH}(\text{Proj}_{1:n}(C_{cl}))$ to hold. In the next subsection, we introduce an alternative implicit expression which does not require a bounded safe set S.

C. Implicit Controlled Invariant Set Expression (Method 2)

In Example 3, the projection of C_{cl} is already convex and, thus, the convex hull computation is omitted. It turns out that the convexity of $\text{Proj}_{1:n}(C_{cl})$ is not a coincidence. We define the nested state transition function by:

$$\mathscr{T}^*(s,(d(t))_{t=0}^k) = \begin{cases} \mathscr{T}(s,d(0)), & k=0, \\ \mathscr{T}(\mathscr{T}^*(s,(d(t))_{t=0}^{k-1}),d(k)), & k>0. \end{cases} \tag{19}$$

Similarly, the nested output function $o^*(s,(d(t))_{t=0}^k)$ is:

$$o^*(s,(d(t))_{t=0}^k) = \begin{cases} o(s,d(0)), & k=0, \\ o(\mathscr{T}^*(s,(d(t))_{t=0}^{k-1}),d(k)), & k>0. \end{cases} \tag{20}$$

Define a preorder relation "$\succeq$" on Q as follows. For any $s_1, s_2 \in Q$, we have that $s_1 \succeq s_2$ if for all $(d_1(t))_{t=0}^{k_1} \in D^{k_1}$ and $(d_2(t))_{t=0}^{k_2} \in D^{k_2}$ with non-negative integers $k_1, k_2 \leq |Q|^2$, $o^*(s_1,(d_1(t))_{t=0}^{k_1}) = o^*(s_1,(d_2(t))_{t=0}^{k_2})$ implies $o^*(s_2,(d_1(t))_{t=0}^{k_1}) = o^*(s_2,(d_2(t))_{t=0}^{k_2})$.

Here, the "=" sign in $o^*(s_i,(d_1(t))_{t=0}^{k_1}) = o^*(s_i,(d_2(t))_{t=0}^{k_2})$ is interpreted as the function o^* mapping two inputs to the same element in Θ (regardless of the parameter θ). Given the definition of the relation $\succeq$ on Q, we can algorithmically check if two states s and s' satisfy $s \succeq s'$ with worst case time complexity $O(|Q|^2)$.

Algorithm 1 Compute Implicit Controlled Invariant Set

inputs: Σ, S, $\Sigma_c = (Q, D, \mathscr{T}, \Theta, o)$.

if a dominant state $s_{dom} \in Q$ exists **then**
  Compute $C_{sub}(s_{dom})$ as in (12).
  return $C_{sub}(s_{dom})$.
else
  for $s_i \in Q_0 \subseteq Q$ **do**
    Compute $C_{sub}(s_i)$ as in (12).
  end for
  Compute C_λ as in (16).
  return C_λ.
end if

Note that the "$\succeq$" relation is not a partial order as it does not satisfy the antisymmetry condition, namely it is possible to have $s_1 \succeq s_2$ and $s_2 \succeq s_1$ but $s_1 \neq s_2$. However, the following theorem shows that the $\succeq$ relation in Q actually implies the partial order on the sets $\{\text{Proj}_{1:n}(C_{sub}(s))\}_{s \in Q}$ defined by the set inclusion.

Theorem 3. *Given the $\succeq$ relation defined on the set Q, for any two states s_1, $s_2 \in Q$, $s_1 \succeq s_2$ implies that $Proj_{1:n}(C_{sub}(s_1)) \supseteq Proj_{1:n}(C_{sub}(s_2))$.*

We call a state $s_{max} \in Q$ a *maximal state* if for any $s' \in Q$, $s' \succeq s_{max}$ implies $s_{max} \succeq s'$, and call a state s_{dom} a *dominant state* if $s_{dom} \succeq s$ for all $s \in Q$. Denote Q_{max} as the set of all the maximal states in Q.

Corollary 1. *Suppose that there exists a dominant state $s_{dom} \in Q$. Then, $Proj_{1:n}(C_{cl}) = Proj_{1:n}(C_{sub}(s_{dom}))$.*

Corollary 1 explains our observation in Example 3.

Example 4. For the mealy machine in Example 2, $s_1 \succeq s_2$ and $s_2 \succeq s_1$. Thus, both s_1 and s_2 are dominant states. Then, $\text{Proj}_{1:n}(C_{cl}) = \text{Proj}_{1:n}(C_{sub}(s_1)) = \text{Proj}_{1:n}(C_{sub}(s_2))$.

Corollary 2. *Define a partition over Q_{max} as follows: For s and $s' \in Q$, s and s' belong to the same component if $s \succeq s'$ and/or $s' \succeq s$. Let a set $Q_0 \subseteq Q_{max}$ contain exactly one state from each component of this partition. Then, $Proj_{1:n}(C_{cl}) = \cup_{s_{max} \in Q_0} Proj_{1:n}(C_{sub}(s_{max}))$.*

According to Corollary 1, if a dominant state s_{dom} exists in Q, the RCIS $\mathbf{CH}(\text{Proj}_{1:n}(C_{cl}))$ is simply the projection of $C_{sub}(s_{dom})$ onto the first n coordinates. In this case, we can directly take $C_{sub}(s_{dom})$ as the implicit representation of the RCIS $\mathbf{CH}(\text{Proj}_{1:n}(C_{cl}))$; otherwise, by Section IV-B, we construct C_λ as the implicit RCIS. Note that according to Corollary 2, we can replace Q by Q_0 in the definition of C_λ. The overall procedure of computing implicit RCISs is summarized in Algorithm 1.

V. Bridge Measurable and Non-measurable Disturbances

In this section we prove a connection between measurable and non-measurable disturbances, which enables our method to compute RCISs for systems with any type of disturbances.

Suppose a system Σ in (1) has a non-measurable disturbance $d \in D$. We construct a system Σ' with a measurable disturbance by adding a one-step delay:

$$\Sigma' : \begin{bmatrix} x(t+1) \\ u(t+1) \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} + \begin{bmatrix} 0 \\ I_m \end{bmatrix} v(t) + \begin{bmatrix} I_n \\ 0 \end{bmatrix} d(t), \quad (21)$$

with state $(x,u) \in \mathbb{R}^{n+m}$, input $v \in \mathbb{R}^m$, a measurable disturbance $d \in D \subseteq \mathbb{R}^n$, and I_n, I_m being the $n \times n$ and $m \times m$ identity matrices respectively.

Let the safe set of Σ be $S \subset \mathbb{R}^{n+m}$. We want to compute an RCIS for Σ within S. The next theorem reveals that this can be achieved by computing an RCIS for Σ' within $S \times \mathbb{R}^m$.

Theorem 4. *Given the systems Σ in (1) and Σ' in (21), if C' is an RCIS for Σ' in $S \times \mathbb{R}^m$, the projection $Proj_{1:n}(C')$ of C' onto the first n coordinates is an RCIS for Σ in S.*

If C' is the maximal RCIS for Σ' in $S \times \mathbb{R}^m$, then $Proj_{1:n}(C')$ is the maximal RCIS for Σ in S.

Thanks to Theorem 4, in terms of computing RCISs, any method designed for measurable disturbances can be applied to systems with non-measurable disturbances.

VI. Case Study

A. Mealy machines with dominant states

We present two classes of mealy machines that contain at least one dominant state.

1) Simple Loop: Given an integer $L > 0$, let $Q = \{s_i\}_{i=1}^L$ and $\Theta = \{u_i\}_{i=1}^L$. Define, for all $d \in D$, the state transition and output functions as:

$$\mathscr{T}(s_i,d) = \begin{cases} s_{i+1} & i < L, \\ s_1 & i = L. \end{cases}, \; o(s_i,d) = \begin{cases} u_{i+1} & i < L, \\ u_1 & i = L. \end{cases} \quad (22)$$

For such a structure, any $s \in Q$ is a dominant state.

2) Tree Structure: Suppose the cardinality of the disturbance set $|D| = K$. Given an integer $L > 0$, define $N = (K^L - 1)/(K-1)$. Let $Q = \{s_0\} \cup \bigcup_{i=1}^L D^i$. That is, Q is the union of s_0 and all finite sequences of elements in D with length less than or equal to L. We assign one output for each $s \in Q$ denoted by $u(s)$. Thus, $\Theta = \{u(s)\}_{s \in Q}$. The state transition function is defined as for all $d \in D$:

$$\mathscr{T}(s,d) = \begin{cases} d & s = s_0, \\ sd & s \in D^k, k < L, \\ s(2:L)d & s \in D^L, \end{cases} \quad (23)$$

where $sd \in D^{k+1}$ denotes the concatenation of $s \in D^k$ and $d \in D$, and $s(2:L)d$ denotes the concatenation of the subsequence $s(2:L)$ of s and $d \in D$. For instance, if $s = d_1 d_2 \cdots d_L$, then $s(2:L)d = d_2 \cdots d_L d$.

For $L = K = 2$, the state transition function is shown in Fig. 2. We call this class of mealy machines *tree structure*

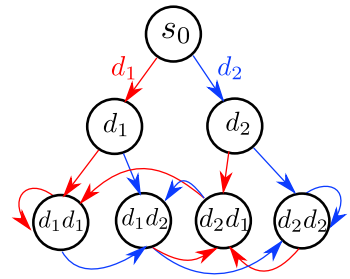

Fig. 2: The tree-structure mealy machine ($L = 3$, $D = \{d_1, d_2\}$). The red arrow and blue arrow indicate transitions under d_1 and d_2 respectively.

since the mealy machine transition graph, as shown in Fig. 2, embeds a tree with s_0 the root node.

Given the state transition function, the output function is simply defined as:

$$o(s,d) = u(\mathscr{T}(s,d)). \quad (24)$$

For any tree-structure mealy machine, s_0 is the dominant state. Intuitively, the tree-structure mealy machine memorizes the past L disturbance measurements and assigns a control input to each possible combination of the past L disturbances.

Finally, for both classes of mealy machines introduced here, it can be proven that by increasing the number of discrete states (complexity), that is increasing L, we tend to obtain larger RCISs.

B. Lane keeping supervision

Consider a 4-dimensional linearized bicycle vehicle dynamics with respect to a constant longitudinal velocity $30m/s$ in [15], discretized with time step $\Delta t = 0.1s$. The system states consist of the lateral displacement y, lateral velocity v, yaw angle $\Delta\Psi$ and yaw rate r. The control input u is the steering angle. The disturbance is $d = (0, 0, -r_d \Delta t, 0)$, where $r_d \in \mathbb{R}$ is the road curvature within a range $|r_d| \leq r_{d,max}$. The safe set is given by constraints $|y| \leq 0.9$, $|v| \leq 1.2$, $|\Delta\Psi| \leq 0.05$, $|r| \leq 0.3$ and $|u| \leq \pi/2$.

The future road curvature can be measured in ahead and thus d is a measurable disturbance [13]. We compare our method with Method 2 in [9], LMI-based low-complexity RCIS in [4] and the Maximal RCIS. Our method uses the tree structure with $L = 4$ in Section VI-A as the mealy machine controller. For Method 2 in [9], we set the parameter $L = 14$ and compute the lifted set in high dimensional space as an implicit RCIS. Note that Method 2 with parameter L is the same as our method equipped with simple loop controller with parameter L. For the LMI-based method in [4], we set the parameter $\rho = 1$ and run the iterative algorithm until convergence. The methods in [9], [4] consider non-measurable disturbances only. To make a fair comparison, our method computes RCISs for d being measurable and/or non-measurable respectively. We evaluate the algorithm performance by their computation time and the volume percentage of the resulting RCISs to the Maximal

TABLE I: Computation Time and Volume Percentage of Computed RCIS to the Maximal RCIS. (Lane Keeping)

$r_{d,max}$		0.01	0.015	0.03	0.05	0.07
Our method	Time (s)	0.042	0.035	0.037	0.035	0.032
(d meas.)	Vol (%)	100.00	99.99	99.89	98.91	74.75
Our method	Time (s)	0.071	0.062	0.072	0.063	0.060
(d non-meas.)	Vol (%)	100.00	100.00	100.00	99.89	0
Method 2 of	Time (s)	0.506	0.443	0.404	0.397	0.484
[9] ($L = 14$)	Vol (%)	99.82	87.64	0	0	0
LMI Method	Time (s)	0.449	0.519	0.562	0.500	0.564
[4] ($\rho = 1$)	Vol (%)	0	0	0	0	0
Maximal RCIS	Time (s)	13.084	18.918	15.525	15.698	21.513

RCIS. The volume percentage is estimated by monte carlo method with sample size $N = 10^4$.

The comparison results are shown in Table I: According to the 2nd, 3rd and 4th rows of Table I, when dealing with non-measurable disturbances only, our method outperforms Method 2 of [9] and LMI-based method in [4] in both the computation time and the volume of the resulting RCIS for all $r_{d,max}$, showing a strong robustness to non-measurable disturbances. The LMI-based method encounters an infeasible optimization problem in all test cases and thus has 0 volume percentage. Method 2 of [9], as a special case of the proposed method, has a decent volume percentage when the disturbance range is small. But as $r_{d,max} > 0.015$, the RCIS from Method 2 of [9] becomes empty, while our method still has volume percentage greater than 98% for both measurable and non-measurable cases.

Shown by the first 2 rows of Table I, when $r_{d,max} = 0.07$, our method returns a nonempty RCIS for d being measurable, but returns an empty RCIS for d being non-measurable. Thus, by considering d as a measurable disturbance, our method is robust to a larger range of disturbances. Finally, comparing the first 2 rows with the last row of Table I, when $r_{d,max} < 0.07$, our method computes implicit RCISs with almost the same size as the Maximal RCISs, using less than 0.3% computation time of the Maximal RCISs.

Next, we illustrate how the computed implicit RCIS can be used to supervise a nominal controller. Suppose the current state $x(t)$ belongs to the RCIS $\text{Proj}_{1:n}(C_{sub}(s_0))$. Given the nominal steering input $u_d(t)$ and the disturbance $d(t)$ at time t, we minimally change the input $u_d(t)$ such that the next state $x(t+1)$ stays in the RCIS $\text{Proj}_{1:n}(C_{sub}(s_0))$ by solving the following quadratic program:

$$\begin{aligned} &\min_{u(t),\theta} \|u_d(t) - u(t)\|_2^2 \\ &\text{subject to } (Ax(t) + Bu(t) + d(t), \theta) \in C_{sub}(s_0), \end{aligned} \tag{25}$$

where A, B are the system matrices and θ is a slack variable. We use the solution $u(t)$ of (25) as the actual steering input to the vehicle. The feasibility of (25) is guaranteed since $\text{Proj}_{1:n}(C_{sub}(s_0))$ is an RCIS and $x(t) \in \text{Proj}_{1:n}(C_{sub}(s_0))$.

We compare the supervised inputs obtained in (25) to the ones obtained based on the Maximal RCIS C_{max} via the following quadratic program:

$$\begin{aligned} &\min_{u(t)} \|u_d(t) - u(t)\|_2^2 \\ &\text{subject to } Ax(t) + Bu(t) + d(t) \in C_{max}. \end{aligned} \tag{26}$$

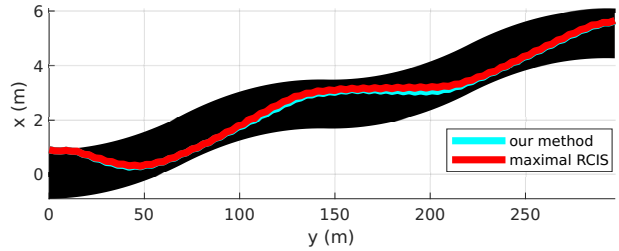

Fig. 3: Vehicle maneuvers under control inputs supervised by our implicit RCIS (cyan curve) and the Maximal RCIS (red curve). The black region indicates the road surface.

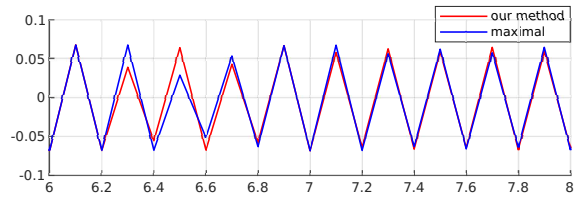

Fig. 4: Vehicle steering inputs supervised by our implicit RCIS (red curve) and the Maximal RCIS (blue curve) ($6 \leq t \leq 8$).

The nominal controller is $u_d(t) = -0.1812y - 0.0373v - 4.5996\Delta\Psi - 0.6649r$. We run two simulations with the same initial states and control inputs obtained from (25) and (26) respectively ($r_{d,max} = 0.015$). As shown in Fig. 3 and Fig. 4, the vehicle maneuvers and steering inputs supervised by our implicit RCIS $C_{sub}(s_0)$ and the Maximal RCIS are very close to each other. The maximal difference between the control inputs from (25) and (26) is around 0.035 at $t = 6.5s$. This observation is consistent with the results shown in Table I, where the volume of our implicit RCIS is approximately 99.96% of the volume of the Maximal RCIS.

C. Chain of integrators

Consider a discrete-time n-th order integrator:

$$x(t+1) = \left(I_n + \begin{bmatrix} 0 & I_{n-1} \\ 0 & 0 \end{bmatrix}\right) x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (u(t) + d(t)) \tag{27}$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}$ and $d \in \mathbb{R}$. I_n indicates the identity matrix in $\mathbb{R}^{n \times n}$. d is considered as a measurable disturbance within range $|d| \leq 0.1$. The safe set is $S = \{(x,u) \mid |x_i| \leq 1, \forall i = 1,\ldots,n,\ |u| \leq 1\}$.

The comparison results of our approach (tree structure, $L = 4$) with Method 2 of [9] ($L = 14$) and the LMI-based method in [4] ($\rho = 1$) are shown in Table II. For $n \leq 4$, our method outperforms the other 2 methods in computation time and volume percentage. For $n = 2$, our method returns exactly the Maximal RCIS, depicted in Fig. 5. For $n \geq 6$, the Maximal RCIS does not terminate within 1 hour. Thus we only check if the computed RCISs are empty or not instead of comparing their volume to the Maximal RCIS. When $n \geq 6$, our method is the only one that returns non-empty RCISs. Note that even though the implicit RCIS has closed-form expression, the number of constraints in the implicit RCIS grows exponentially as n increases. In this example,

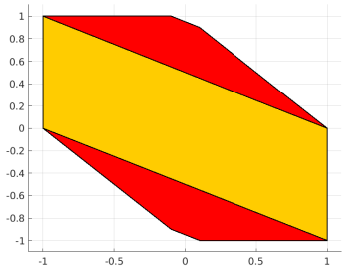

Fig. 5: RCISs for double integrator. Yellow: RCIS from LMI-based method [4]. Red: the Maximal RCIS computed by both our method and [9].

TABLE II: Computation Time and Volume Percentage of Computed RCIS to the Maximal RCIS (Chain of Integrators).

n		2	4	6	8	10
Our method	Time (s)	0.005	0.025	0.368	9.287	339.060
(d meas.)	Vol (%)	100	98.79	>0	>0	>0
Method 2 of	Time (s)	0.183	0.341	2.420	7.168	37.105
[9] ($L=14$)	Vol (%)	74.49	0	0	0	0
LMI Method	Time (s)	3.465	0.603	0.952	1.405	3.1402
[4] ($\rho=1$)	Vol (%)	66.85	≈ 0	0	0	0
Maximal RCIS	Time (s)	0.734	11.114	>3600	>3600	>3600

TABLE III: Computation Time and Volume Percentage of Computed RCIS to the Maximal RCIS (Truck with N trailers).

System dimension		$n=3$	$n=5$	$n=7$	$n=9$
Our method	Time (s.)	0.109	0.781	8.669	163.1
	Vol (%)	100	100	>0	0
Method 2 of	Time (s.)	0.547	0.814	1.352	6.577
[9] (L=14)	Vol (%)	100	98.90	0	0
Maximal RCIS	Time (s.)	0.746	13.76	>3600	>3600

for $n=10$, it takes about 339s for our method to generate the implicit RCIS, which is a polytope in $\mathbb{R}^{24}$ with about 36×10^4 constraints.

D. Truck with N trailers

Consider a continuous-time model for a truck with N trailers [16]. The state consists of the $N+1$ velocity values, each for the truck and the N trailers, and the N spring elongations in between them. Hence, N trailers correspond to dimension $n=2N+1$. The input is the velocity of the truck. We discretize the model with a sampling time of T_s seconds assuming piecewise constant inputs.

Table III shows the results of this case study for our method and the approach in [9]. For $n\geq 7$ the method computing the Maximal RCIS does not terminate after 1 hour, and, hence, we only check non-emptiness of sets instead of volume percentage. When the Maximal RCIS is computed, we see that our approach covers it, but due to the implicit representation, the running times are much faster. However, we see that in this example, after some point, as the dimension becomes large, the set our algorithm returns is empty. This can be understood as by adding more trailers the noise from each spring compounds towards the ones behind it, resulting in the shrinking of the RCIS.

VII. Conclusion

In this paper, we present a novel method of computing implicit RCISs in closed-form. The key insight is to construct a closed-loop system with a parameterized automaton-based controller. The implicit RCISs obtained by our method characterize the set of feasible initial states and controller parameters under which the system state-input trajectory stays in the safe set. Compared with the standard iterative methods [1], [2], all the computations of our method are done in one-shot, which guarantees finite-time termination and better scalability. Several numerical examples are provided to demonstrate the efficiency and practicality of the proposed method.

References

[1] D. Bertsekas, "Infinite time reachability of state-space regions by using feedback control," *Automatic Control, IEEE Transactions on*, vol. AC-17, pp. 604 – 613, 11 1972.

[2] L. P. Nilsson, "Correct-by-construction control synthesis for high-dimensional systems," Ph.D. dissertation, 2017.

[3] M. Rungger and P. Tabuada, "Computing robust controlled invariant sets of linear systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3665–3670, July 2017.

[4] F. Tahir and I. M. Jaimoukha, "Low-complexity polytopic invariant sets for linear systems subject to norm-bounded uncertainty," *IEEE Transactions on Automatic Control*, vol. 60, no. 5, pp. 1416–1421, 2014.

[5] A. Gupta and P. Falcone, "Full-complexity characterization of control-invariant domains for systems with uncertain parameter dependence," *IEEE control systems letters*, vol. 3, no. 1, pp. 19–24, 2018.

[6] N. Michel, S. Olaru, G. Valmorbida, S. Bertrand, and D. Dumur, "Invariant sets for discrete-time constrained linear systems using a sliding mode approach," in *2018 European Control Conference (ECC)*. IEEE, 2018, pp. 2929–2934.

[7] T. Anevlavis and P. Tabuada, "Computing controlled invariant sets in two moves," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 6248–6254. [Online]. Available: https://doi.org/10.1109/CDC40024.2019.9029610

[8] T. Anevlavis and P. Tabuada, "A simple hierarchy for computing controlled invariant sets," in *Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '20. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3365365.3382205

[9] T. Anevlavis, Z. Liu, N. Ozay, and P. Tabuada, "An enhanced hierarchy for (robust) controlled invariance," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 4860–4865.

[10] A. Wintenberg and N. Ozay, "Implicit invariant sets for high-dimensional switched affine systems," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 3291–3297.

[11] T. Anevlavis, Z. Liu, N. Ozay, and P. Tabuada, "Controlled invariant sets: implicit closed-form representations and applications," *arXiv preprint arXiv:2107.08566*, 2021.

[12] Z. Liu and N. Ozay, "Safe online planning in unknown nonconvex environments with implicit controlled invariant sets," *IFAC-PapersOnLine*, vol. 54, no. 5, pp. 163–168, 2021.

[13] S. Xu and H. Peng, "Design, analysis, and experiments of preview path tracking control for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 48–58, 2019.

[14] P. J. Antsaklis and A. Michel, *Linear Systems*, 1st ed. Birkhäuser Basel, 2006.

[15] S. W. Smith, P. Nilsson, and N. Ozay, "Interdependence quantification for compositional control synthesis with an application in vehicle safety systems," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 5700–5707.

[16] M. Rungger, M. Mazo, Jr., and P. Tabuada, "Specification-guided controller synthesis for linear systems and safe linear-time temporal logic," in *Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '13. New York, NY, USA: ACM, 2013, pp. 333–342.

APPENDIX

Proof of Proposition 1 . Let $x \in C$ and $d \in D$. By the Definition 5, $\mathscr{R}(\Sigma, f(x,d)) \subseteq \mathscr{R}(\Sigma_a, x) \subseteq S$ and thus $f(x,d) \in C$. Hence, C is an RPIS by definition.

Suppose x belongs to an arbitrary RPIS within S. By definition, $\mathscr{R}(\Sigma_a, x) \subseteq S$. Thus, $x \in C$ and C is the Maximal RPIS. □

Proof of Proposition 2. The "only if" direction is obvious. It is left to show the "if" direction. Suppose the safe set is S. Let C be an RCIS for the system Σ' in S and x be a point in C. We want to show that for all $d \in D$, there exists u such that $Ax+Bu+d \in S$. Since $D = \mathbf{CH}(D_v)$, there exists a finite $K > 0$ such that $d = \sum_{i=1}^{K} \alpha_i d_i$ for some d_1, ..., $d_K \in D_v$ and some α_1, ..., $\alpha_K \geq 0$ satisfying $\sum_{i=1}^{K} \alpha_i = 1$. Since C is controlled invariant for Σ', for each $d_i \in D_v$, there exists u_i such that $(x, u_i) \in S$ and $Ax + Bu_i + Ed_i \in C$. Define $u = \sum_{i=1}^{K} \alpha_i u_i$. It is easy to show that $(x,u) \in S$ and $Ax + Bu + Ed \in C$, by the convexity of S and C. Thus, C is an RCIS within S for Σ. □

Proof of Theorem 1. Denote $\mathbf{CH}(\mathrm{Proj}_{1:n}(C_{cl}))$ by C_p. Let $x \in C_p$ and $d \in D$. We want to show that there exists u such that $(x,u) \in S$ and $Ax + Bu + d \in C_p$.

By definition of convex hull, there exist a positive integer $k > 0$, k vectors $x_i \in \mathrm{Proj}_{1:n}(C_{cl})$ and k scalars $\alpha_i \in [0,1]$ for i from 1 to k such that $\sum_{i=1}^{k} \alpha_i = 1$ and $\sum_{i=1}^{k} \alpha_i x_i = x$. For each i, there exists θ_i and s_i such that $(x_i, \theta_i, s_i) \in C_{cl}$. We define $u_i = o(s_i, d)$. Note that by the definition of S_{cl}, $(x_i, u_i) \in S$. Also, since C_{cl} is an RPIS, $(Ax_i + Bu_i + d, \theta_i, \mathscr{T}(s_i, d)) \in C_{cl}$ and thus $Ax_i + Bu_i + d \in C_{proj}$. We define $u = \sum_{i=1}^{k} \alpha_i u_i$. Since S is convex and $(x_i, u_i) \in S$, $(x,u) = \sum_{i=1}^{k} \alpha_i (x_i, u_i) \in S$. Since C_p is convex and $Ax_i + Bu_i + d \in C_p$, $Ax + Bu + d = \sum_{i=1}^{k} \alpha_i (Ax_i + Bu_i + d) \in C_p$. Thus, C_p is an RCIS for the system Σ in S. □

Proof of Theorem 2. We want to show $\mathscr{R}(\Sigma_{cl}, (x, \theta, s))$ is finite. Let $((x(t), \theta(t), s(t)))_{t=0}^{\infty}$ be the trajectory of Σ_{cl} with initial state $(x(0), \theta(0), s(0)) = (x, \theta, s)$. Let $(d(t))_{t=0}^{\infty}$ be the disturbance sequence. Given A is nilpotent, that is $A^h = 0$ for some $h \geq 0$, we have that

$$x(t) = \begin{cases} A^t x + \sum_{i=0}^{t-1} A^{t-1-i}[Bo(s(i), d(i); \theta) + d(i)] & t < h, \\ \sum_{i=t-h}^{t-1} A^{t-1-i}[Bo(s(i), d(i); \theta) + d(i)] & t \geq h. \end{cases} \tag{28}$$

Since $s(t)$ and $d(t)$ belong to finite sets Q and D, $o(s(t), d(t); \theta)$ belongs to the finite set $U(\theta) = \{o(s, d; \theta)\}_{s \in Q, d \in D}$. Thus, according to (28), $x(t)$, as a function of $o(s(t), d(t); \theta)$ and $d(t)$ for $t \geq h$, must belongs to a finite set, denoted by $X(\theta)$. Thus, the reachable set $\mathscr{R}(\Sigma_{cl}, (x, \theta, s)) \subseteq X(\theta) \times \{\theta\} \times Q$ is a finite set. □

Proof of Theorem 3. We want to derive a sufficient condition under which $\mathrm{Proj}_{1:n}(C_{sub}(s_1)) \supseteq \mathrm{Proj}_{1:n}(C_{sub}(s_2))$. Note that if for all $(x, \theta_2) \in C_{sub}(s_2)$, there exists θ_1 such that $(x, \theta_1) \in C_{sub}(s_1)$, then we have $\mathrm{Proj}_{1:n}(C_{sub}(s_1)) \supseteq \mathrm{Proj}_{1:n}(C_{sub}(s_2))$.

Similar to how we define $o^*(s,d)$, we define the parameterized nested output function as

$$o^*(s, (d(t))_{t=0}^{k}; \theta) = \begin{cases} o(s, d(0); \theta) & k = 0, \\ o(\mathscr{T}^*(s, (d(t))_{t=0}^{k-1}), d(k); \theta) & k > 0. \end{cases} \tag{29}$$

Given s and θ, the parameterized nested output function $o^*(s, \cdot; \theta)$ becomes a function of $(d(t))_{t=0}^{k}$ in $\cup_{i=1}^{\infty} D^i$. If for any θ_2, we can always find a θ_1 such that the functions $o^*(s_2, \cdot; \theta_2) = o^*(s_1, \cdot; \theta_1)$, then for all $(x, \theta_2) \in C_{sub}(s_2)$, $(x, \theta_1) \in C_{sub}(s_1)$. Intuitively, recall that $(x, \theta_2) \in C_{sub}(s_2)$ if $(x(k), o^*(s_2, (d(t))_{t=0}^{k})) \in S$ for all $k \geq 0$ and $(d(t))_{t=0}^{k} \in D^k$. If we know that $(x(k), o^*(s_2, (d(t))_{t=0}^{k}; \theta_2)) \in S$ for all $k \geq 0$, then we know $(x(k), o^*(s_2, (d(t))_{t=0}^{k}; \theta_2)) \in S$ for all $k \geq 0$ since $o^*(s_2, \cdot; \theta_2) = o^*(s_1, \cdot; \theta_1)$. Thus, $(x, \theta_1) \in C_{sub}(s_1)$.

Now our goal is to derive a sufficient condition under which there exists a θ_1 such that $o^*(s_1, \cdot; \theta_1) = o^*(s_2, \cdot; \theta_2)$ for all θ_2.

Lemma 1. *Given s_1, $s_2 \in Q$ and θ_1, θ_2, the functions $o^*(s_1, \cdot; \theta_1) = o^*(s_2, \cdot; \theta_2)$ if and only if $o^*(s_1, (d(t))_{t=0}^{k}; \theta_1) = o^*(s_2, (d(t))_{t=0}^{k}; \theta_2)$ for all $k \leq |Q|^2$ and all $(d(t))_{t=0}^{k} \in D^k$.*

According to Lemma 1, given any θ_2, we can directly solve for a θ_1 satisfying $o^*(s_1, (d(t))_{t=0}^{k}; \theta_1) = o^*(s_2, (d(t))_{t=0}^{k}; \theta_2)$ for all $k \leq |Q|^2$ and all $(d(t))_{t=0}^{k} \in D^k$, which is a system of linear equations on θ_1. It can be checked that given any θ_2, the solvability of the system of equations on θ_1 is guaranteed if for all $(d_1(t))_{t=0}^{k_1}$ and $(d_2(t))_{t=0}^{k_2}$ with k_1, $k_2 \leq |Q|^2$, $o^*(s_1, (d_1(t))_{t=0}^{k_1}) = o^*(s_1, (d_2(t))_{t=0}^{k_2})$ implies $o^*(s_2, (d_1(t))_{t=0}^{k_1}) = o^*(s_2, (d_2(t))_{t=0}^{k_2})$, that is $s_1 \succeq s_2$ by definition. □

Proof of Lemma 1 . Given the mealy machine $(Q, D, \mathscr{T}, \Theta, o)$, we can construct a product mealy machine $(Q \times Q, D, \mathscr{T}_{pd}, \Theta \times \Theta, o_{pd})$ where for all s_i, $s_j \in Q$ and $d \in D$

$$\mathscr{T}_{pd}((s_i, s_j), d) = (\mathscr{T}(s_i, d), \mathscr{T}(s_j, d)), \tag{30}$$

$$o_{pd}((s_i, s_j), d) = (o(s_i, d), o(s_j, d)). \tag{31}$$

Given θ_1 and θ_2 as two value assignments of Θ, we define the parameterized output function $o_{pd}((s_i, s_j), d; \theta_1, \theta_2) = (o(s_i, d; \theta_1), o(s_j, d; \theta_2))$.

Given s_1, $s_2 \in Q$ and θ_1 and θ_2, by construction, $o^*_{pd}((s_1, s_2), \cdot; \theta)$ is equal to $(o^*(s_1, \cdot; \theta_1), o^*(s_2, \cdot; \theta_2))$. Thus, $o^*(s_1, \cdot; \theta_1) \neq o^*(s_2, \cdot; \theta_2)$ if and only if there exists a $(d(t))_{t=0}^{k}$ such that $(s_1', s_2') = \mathscr{T}^*_{pd}((s_1, s_2), (d(t))_{t=0}^{k-1})$ and $o_{pd}((s_1', s_2'), d(k); \theta_1, \theta_2) = (u_1, u_2)$ for some u_1, $u_2 \in \Theta$, $\overline{u}_1 \neq \overline{u}_2$. Since there are only $|Q|^2$ states in the product mealy machine, if (s_1', s_2') can be visited from (s_1, s_2) under action sequence $(d(t))_{t=0}^{k-1}$, the smallest k we need is less than or equal to $|Q|^2$. Thus, if $o^*(s_1, \cdot; \theta_1) \neq o^*(s_2, \cdot; \theta_2)$, there must exists a $(d(t))_{t=0}^{k}$ with $k \leq |Q|^2$ such that $o^*(s_1, (d(t))_{t=0}^{k}; \theta_1) \neq o^*(s_2, (d(t))_{t=0}^{k}; \theta_2)$ □

Proof of Theorem 4. Denote $C_p = \mathrm{Proj}_{1:n}(C')$. Suppose C' is an RCIS of Σ' in $S \times \mathbb{R}^m$. Let $x \in C_p$. We want to show that there exist u such that $(x,u) \in S$ and for all $d \in D$, $Ax + Bu + d \in C_p$.

By definition of C_p, there exists $u \in \mathbb{R}^m$ such that $(x,u) \in C' \subseteq S$. Furthermore, since C' is controlled invariant, there exists $v \in \mathbb{R}^m$ such that $(Ax+Bu+d,v) \in C'$ for all $d \in D$. Thus, $Ax+Bu+d \in C_p$ for all $d \in D$. Thus, we showed that C_p is an RCIS for the system Σ in S.

Next, suppose that C' is the Maximal RCIS for Σ' in $S \times \mathbb{R}^m$. Also, suppose that C_{max} is the Maximal RCIS for Σ in S. We want to show that $\text{Proj}_{1:n}(C') = C_{max}$. Note that $\text{Proj}_{1:n}(C') \subseteq C_{max}$ as $\text{Proj}_{1:n}(C')$ is controlled invariant for Σ in S. We need to show that $\text{Proj}_{1:n}(C') \supseteq C_{max}$, which is done in 3 steps.

First, define the set $C'_{max} = \{(x,u) \mid (x,u) \in S, Ax+Bu+d \in C_{max}, \forall d \in D\}$. We want to show that C'_{max} is controlled invariant for Σ' in $S \times \mathbb{R}^m$. Let $(x,u) \in C'_{max}$ and $d \in D$. By construction, $(x,u) \in S$ and $x^+ = Ax+Bu+d \in C_{max}$. Since C_{max} is controlled invariant for Σ, there exists $v \in \mathbb{R}^m$ such that $(x^+,v) \in S$ and $Ax^+ + Bu + d^+ \in C_{max}$ for all $d^+ \in D$. Thus, by definition of C'_{max}, $(x^+,v) = (Ax+Bu+d,v) \in C'_{max}$. Thus, C'_{max} is an RCIS for Σ' in $S \times \mathbb{R}^m$.

Second, as C' is the Maximal RCIS for Σ' in $S \times \mathbb{R}^m$, $C' \geq C'_{max}$. Thus, $\text{Proj}_{1:n}(C') \supseteq \text{Proj}_{1:n}(C'_{max})$.

Finally, note that for all $x \in C_{max}$, there exists u such that $(x,u) \in S$ and $Ax+Bu+d \in C_{max}$ for all $d \in D$, namely that $(x,u) \in C'_{max}$. Hence, $C_{max} \subseteq \text{Proj}_{1:n}(C'_{max}) \subseteq \text{Proj}_{1:n}(C')$. That is, $\text{Proj}_{1:n}(C') = C_{max}$ is the Maximal RCIS for Σ in S. $\square$