Controlled invariant sets: implicit closed-form representations and applications

Tzanis Anevlavis*, Zexiang Liu*, *Graduate Student Member, IEEE*, Necmiye Ozay, *Senior Member, IEEE*, and Paulo Tabuada *Fellow, IEEE*

***Abstract*— We revisit the problem of computing (robust) controlled invariant sets for discrete-time linear systems. Departing from previous approaches, we consider implicit, rather than explicit, representations for controlled invariant sets. Moreover, by considering such representations in the space of states and finite input sequences we obtain *closed-form* expressions for controlled invariant sets. An immediate advantage is the ability to handle high-dimensional systems since the closed-form expression is computed in a single step rather than iteratively. To validate the proposed method, we present thorough case studies illustrating that in safety-critical scenarios the implicit representation suffices in place of the explicit invariant set. The proposed method is complete in the absence of disturbances, and we provide a weak completeness result when disturbances are present.**

***Index Terms*— Constrained control, Controlled invariant sets, Linear systems, Optimization, Robust control**

I. Introduction

In an increasingly autonomous world, safety has recently come under the spotlight. A safety enforcing controller is understood as one that indefinitely keeps the state of the system within a set of safe states notwithstanding the presence of uncertainties. A natural solution that guarantees safety is to initialize the state of the system inside a Robust Controlled Invariant Set (RCIS) within the set of safe states. Any RCIS is defined by the property that any trajectory starting within, can always be forced to remain therein and, hence, inside the set of safe states. Consequently, RCISs are at the core of controller synthesis for safety-critical applications.

Since the conception of the standard method for computing the Maximal RCIS of discrete-time systems [6], which is known to suffer from poor scaling with the system's dimension and no guarantees of termination, numerous approaches attempted to alleviate these drawbacks. A non-exhaustive overview is found in Section VIII.

Submitted on July 5th, 2021.

This work was partially supported by the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, and also by NSF grants 1553873, 1918123, 1931982.

Tzanis Anevlavis and Paulo Tabuada are with the Department of Electrical and Computer Engineering at University of California, Los Angeles, CA 90095 USA (e-mail: {t.anevlavis, tabuada} @ ucla.edu).

Zexiang Liu and Necmiye Ozay are with the Department of Electrical Engineering and Computer Science at University of Michigan, Ann Arbor, MI 48109 USA (e-mail: {zexiang, necmiye}@umich.edu).

* T. Anevlavis and Z. Liu are co-first authors.

An alternative approach is to construct an implicit representation for an RCIS. The specific implicit representation used in this paper is a set in the higher dimensional space of states and finite input sequences. We argue that in many practical, safety-critical applications, such as Model Predictive Control (MPC) and supervisory control, knowledge of the explicit RCIS is not required and the implicit representation suffices. Consequently, by exploiting the efficiency of the implicit representation the aforementioned ideas are suitable for systems with large dimensions.

In this manuscript, we propose a general framework for *computing (implicit) RCISs* for discrete-time linear systems with additive disturbances, under polytopic state, input, and even mixed, constraints. We consider RCISs parameterized by collections of *eventually periodic* input sequences and prove that this choice leads to a closed-form expression for an implicit RCIS in the space of states and finite input sequences. Moreover, this choice results in a systematic way to obtain larger RCISs, which we term a *hierarchy*. Essentially, the computed sets include all states for which there exist eventually periodic input sequences that lead to a trajectory that remains within the safe set indefinitely. Once the (implicit) RCIS is computed, any controller rendering the RCIS invariant can be used in practice and a fixed periodic input is not chosen or used. Moreover, we show that this parameterization is rich enough, such that: 1) in the absence of disturbances, our method is complete and sufficient to approximate the Maximal CIS arbitrarily well; 2) in the presence of disturbances, a weak completeness result is established, along with a bound for the computed RCIS that can be approximated arbitrarily well. Finally we study, both theoretically and experimentally, safety-critical scenarios and establish that the efficient implicit representation suffices in place of computing the exact RCIS. In practice, the use of implicit RCISs can be done via optimization programs, e.g., a Linear Program (LP), a Mixed-Integer (MI) program, or a Quadratic Program (QP), and is only limited by the size of the program afforded to solve.

In order to make for a more streamlined presentation, a review of the existing related literature is found at the end of the manuscript.

Notation: Let $\mathbb{R}$ be the set of real numbers and $\mathbb{N}$ be the set of positive integers. For sets $P, Q \subseteq \mathbb{R}^n$, the Minkowski sum is $P + Q = \{x \in \mathbb{R}^n \mid x = p + q, p \in P, q \in Q\}$ and the Minkowski difference is $Q - P = \{x \in \mathbb{R}^n \mid x + P \subseteq Q\}$, where by slightly abusing the notation, we denote the Minkowski sum of a singleton $\{x\}$ and a set P by $x + P$. The

Hausdorff distance between P and Q, denoted by $d(P,Q)$, is induced from the Euclidean norm in $\mathbb{R}^n$. We denote a block-diagonal matrix M with blocks $M_1,\dots,M_N$ by $M=\text{blkdiag}(M_1,\dots,M_N)$. Moreover, given a matrix $A\in\mathbb{R}^{m\times n}$ and a set $P\subseteq\mathbb{R}^n$, the linear transformation of P through A is $AP=\{Ax\in\mathbb{R}^m|x\in P\}$. Given a set $S\subset\mathbb{R}^n\times\mathbb{R}^m$, its projection onto the first n coordinates is $\pi_n(S)$. For any $N\in\mathbb{N}$, let $[N]=\{1,2,\cdots,N\}$. Let $\mathbb{I}$ and $\mathbb{0}$ be the identity and zero matrices of appropriate sizes respectively, while $\mathbb{1}$ is a vector with all entries equal to 1.

II. Problem formulation

Let us begin by providing the necessary definitions.

Definition 1 (Discrete-time linear system): A *Discrete-Time Linear System* (DTLS) Σ is a linear difference equation:

$$x^+=Ax+Bu+Ew, \tag{1}$$

where $x\in\mathbb{R}^n$ is the state of the system, $u\in\mathbb{R}^m$ is the input, and $w\in W\subseteq\mathbb{R}^d$ is a disturbance term. Moreover, we have that $A\in\mathbb{R}^{n\times n}$, $B\in\mathbb{R}^{n\times m}$, and $E\in\mathbb{R}^{n\times d}$.

Definition 2 (Polytope): A *polytope* $S\subset\mathbb{R}^n$ is a bounded set of the form:

$$S=\{x\in\mathbb{R}^n\mid Gx\leq f\}, \tag{2}$$

where $G\in\mathbb{R}^{k\times n}$, $f\in\mathbb{R}^k$ for some $k>0$.

Definition 3 (Robust Controlled Invariant Set): Given a DTLS Σ and a safe set $S_{xu}\subset\mathbb{R}^n\times\mathbb{R}^m$, that is, the set defining the state-input constraints for Σ, a set $\mathcal{C}\subseteq\pi_n(S_{xu})$ is a *Robust Controlled Invariant Set* for Σ within S_{xu} if:

$$x\in\mathcal{C}\Rightarrow\exists u\in\mathbb{R}^m \text{ s.t. } (x,u)\in S_{xu}, Ax+Bu+EW\subseteq\mathcal{C}.$$

Definition 4 (Admissible Input Set): Given an RCIS $\mathcal{C}$ of a DTLS Σ within its safe set S_{xu}, the set $\mathcal{A}(x)$ of admissible inputs at a state x is:

$$\mathcal{A}(x)=\{u\in\mathbb{R}^m|(x,u)\in S_{xu}, Ax+Bu+EW\subseteq\mathcal{C}\}.$$

Assumption 1: In this manuscript we focus on systems and safe sets that satisfy the following:

1) There exists a suitable state feedback transformation that makes the matrix A of system Σ *nilpotent*. For a nilpotent matrix, there exists a $\nu\in\mathbb{N}$ such that $A^\nu=0$.

2) The *safe set* $S_{xu}\subset\mathbb{R}^n\times\mathbb{R}^m$ and the *disturbance set* $W\subset\mathbb{R}^d$ are both polytopes.

Remark 1: For any controllable system Σ, there exists a state feedback transformation satisfying Assumption 1 [4, Ch.3]. In this case, the nilpotency index ν is equal to the largest controllability index of Σ.

For any system Σ satisfying Assumption 1, let $K\in\mathbb{R}^{m\times n}$ be the feedback gain such that $A+BK$ is nilpotent. We construct a system Σ' by pre-feedbacking Σ with $u=Kx+u'$:

$$x^+=(A+BK)x+Bu'+Ew,$$

where $u'\in\mathbb{R}^m$ is the input of the system Σ'. The safe set for Σ' is the polytope induced from the safe set S_{xu} of Σ as $S'_{xu}=\{(x,u')\in\mathbb{R}^n\times\mathbb{R}^m\mid(x,Kx+u')\in S_{xu}\}$.

Proposition 2.1: Any RCIS $\mathcal{C}$ of Σ within S_{xu} is an RCIS of Σ' within S'_{xu} and vice versa.

Proof: The proof is based on the fact that the map from (x,u) to $(x,u')=(x,u-Kx)$ is a bijection from S_{xu} to S'_{xu}.

Consider an RCIS $\mathcal{C}$ of Σ within S_{xu}, and (x,u) with $x\in\mathcal{C}$ and $Ax+Bu+EW\subseteq\mathcal{C}$. Take $u'=u-Kx$. Then, $(x,u')\in S'_{xu}$ since $(x,Kx+u')=(x,u)\in S_{xu}$. Advancing the state x with input u' in Σ' gives $(A+BK)x+Bu'+EW=Ax+BKx-BKx+Bu+EW=Ax+Bu+EW\subseteq\mathcal{C}$. Hence, $\mathcal{C}$ is also an RCIS for Σ' within S'_{xu}. The other direction is shown in a similar way. ■

Based on Proposition 2.1, it can be seen that the problem of finding an RCIS of Σ within S_{xu} is *exactly equivalent* to the problem of finding an RCIS of Σ' within S'_{xv}. That is, for any procedure that takes in (Σ,S_{xu}) and produces a RCIS $\mathcal{C}$, there exists an equivalent procedure that takes in (Σ',S'_{xu}) and produces the same RCIS $\mathcal{C}$, and vice versa. Therefore, in the remainder of this work, we simply assume that the system in (1) (and its safe set S_{xu}) is already transformed to this equivalent form where the matrix A is nilpotent.

The main goal of this paper is to compute an *implicit representation of an RCIS in closed-form*. Hereafter, we refer to this representation as the *implicit RCIS*.

Definition 5 (Implicit RCIS): Given a DTLS Σ, a safe set $S_{xu}\subset\mathbb{R}^n\times\mathbb{R}^m$, and some integer $q\in\mathbb{N}$, a set $\mathcal{C}_{xv}\subseteq\mathbb{R}^n\times\mathbb{R}^q$ is an *Implicit RCIS* for Σ if its projection $\pi_n(\mathcal{C}_{xv})$ onto the first n dimensions is an RCIS for Σ within S_{xu}.

The following result stems directly from Definition 3.

Proposition 2.2: The union of RCISs and the convex hull of an RCIS are robustly controlled invariant.

For dynamical systems, i.e., systems Σ as in (1) where $B=0$, the analogous concept to RCISs is defined below.

Definition 6 (Robust Positively Invariant Set): Given a dynamical system $\Sigma:\ x^+=Ax+Ew$ and a safe set $S_x\subset\mathbb{R}^n$, a set $\mathcal{C}\subset\mathbb{R}^n$ is a *Robust Positively Invariant Subset* (RPIS) for Σ within S if $x\in\mathcal{C}\Rightarrow Ax+EW\subseteq\mathcal{C}$.

We define the *accumulated disturbance set* at time t by:

$$\overline{W}_t=\sum_{i=1}^{t}A^{i-1}EW. \tag{3}$$

By nilpotency of A we have that:

$$\overline{W}_\infty=\sum_{i=1}^{\infty}A^{i-1}EW=\sum_{i=1}^{\nu}A^{i-1}EW. \tag{4}$$

In the literature, $\overline{W}_\infty$ is called the *Minimal RPIS* of the system $x^+=Ax+Ew$ [31].

The next operator is used throughout this manuscript.

Definition 7 (Reachable set): Given a DTLS Σ and a set $X\subset\mathbb{R}^n$, define the *reachable set* from X under input sequence $\{u_i\}_{i=0}^{t-1}$ as:

$$\mathcal{R}_\Sigma\big(X,\{u_i\}_{i=0}^{t-1}\big)=A^tX+\sum_{i=1}^{t}A^{i-1}Bu_{t-i}+\overline{W}_t. \tag{5}$$

Intuitively, $\mathcal{R}_\Sigma\big(X,\{u_i\}_{i=0}^{t-1}\big)$ maps a set X and an input sequence $\{u_i\}_{i=0}^{t-1}$ to the set of all states that can be reached from X in t steps when applying said input sequence. Conventionally, $\mathcal{R}_\Sigma\big(X,\{u_i\}_{i=a}^{b}\big)=X$ if $b<a$, and when X is a singleton, i.e., $X=\{x\}$, we abuse notation to write $\mathcal{R}_\Sigma\big(x,\{u_i\}_{i=0}^{t-1}\big)$.

III. Implicit representation of controlled invariant sets for linear systems

The classical algorithm that computes the *Maximal* RCIS consists of an iterative procedure [6], [11] and theoretically works for any discrete-time system and safe set. However, this approach is known to suffer from the curse of dimensionality and its termination is not guaranteed. To alleviate these drawbacks, we propose an algorithm that is guaranteed to terminate and computes an implicit RCIS efficiently in closed-form, thus being suitable for high dimensional systems. Moreover, by optionally projecting the implicit RCIS back to the original state-space one computes an explicit RCIS. Overall, the proposed algorithm computes controlled invariant sets in one and two moves respectively.

The goal of this section is to present a *finite implicit representation* of an RCIS. That is, we provide a *closed-form* expression for an *implicit RCIS* characterized by constraints on the state and on a finite input sequence, whose length is the design parameter. This results in a polytopic RCIS in a higher dimensional space. Intuitively, the implicit RCIS contains the pairs of states and appropriate finite input sequences that guarantee that the state remains in the safe set indefinitely.

A. General implicit robust controlled invariant sets

We begin by discussing a general construction of a polytopic implicit RCIS. First, we consider inputs u_t to Σ that evolve as the output of a linear dynamical system, Σ_C, whose state is a *sequence of q inputs,* v, i.e.:

$$\Sigma_C \quad : \quad \begin{aligned} v_{t+1} &= Pv_t, \\ u_t &= Hv_t, \end{aligned} \tag{6}$$

where $v \in \mathbb{R}^{mq}$, $P \in \mathbb{R}^{mq\times mq}$, and $H \in \mathbb{R}^{m\times mq}$. The resulting input to Σ can be expressed as:

$$u_t = Hv_t = HP^t v_0, \tag{7}$$

for an initial choice of $v_0 \in \mathbb{R}^{mq}$. We can then lift system Σ, after closing the loop with Σ_C, to the following companion dynamical system:

$$\Sigma_{xv} \quad : \quad \begin{bmatrix} x^+ \\ v^+ \end{bmatrix} = \begin{bmatrix} A & BH \\ \mathbb{0} & P \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} E \\ \mathbb{0} \end{bmatrix} w. \tag{8}$$

Given the safe set S_{xu}, we construct the companion safe set $S_{xv} = \{(x, v) \in \mathbb{R}^n \times \mathbb{R}^{mq} \mid (x, Hv) \in S_{xu}\}$. The companion system of (1) is the closed-loop dynamics of (1) with a control input in (7). Then, the companion safe set simply constrains the closed-loop state-input pairs in the original safe set, i.e., $(x_t, Hv_t) \in S_{xu}$.

Theorem 3.1 (Generalized implicit RCIS): Let $\mathcal{C}_{xv}$ be an RPIS of the companion system Σ_{xv} within the companion safe set S_{xv}. The projection of $\mathcal{C}_{xv}$ onto the first n coordinates, $\pi_n(\mathcal{C}_{xv})$, is an RCIS of the original system Σ within S_{xu}. In other words, $\mathcal{C}_{xv}$ is *an implicit RCIS* of Σ.

Proof: Let $x \in \pi_n(\mathcal{C}_{xv})$. Then, there exists a $v \in \mathbb{R}^{mq}$ such that $(x, v) \in \mathcal{C}_{xv}$. Define $u = Hv$ and pick an arbitrary $w \in W$. By construction of S_{xv}, $(x, u) \in S_{xu}$. Since $\mathcal{C}_{xv}$ is an RPIS, we have that $(x^+, v^+) = (Ax + Bu + Ew, Pv) \in \mathcal{C}_{xv}$ and thus $x^+ \in \pi_n(\mathcal{C}_{xv})$. By Definition 3, $\pi_n(\mathcal{C}_{xv})$ is an RCIS of Σ in S_{xu}. ■

In principle, Theorem 3.1 holds even if Σ_C is nonlinear. However, the choice of a linear system Σ_C, as in (6), makes the computation of the Maximal RPIS of Σ_{xv} more efficient. In what follows, we study the conditions on P and H such that the Maximal RPIS of Σ_{xv} is represented in closed-form.

B. Finite reachability constraints

By definition of the companion safe set S_{xv} and Definition 6, we have that any state (x, v) belongs to the Maximal RPIS of Σ_{xv} within S_{xv}, if and only if, the input sequence $\{u_i\}_{i=0}^{t-1}$, with each input as in (7), satisfies:

$$\left(\mathcal{R}_\Sigma\left(x, \{u_i\}_{i=0}^{t-1}\right), u_t\right) \subseteq S_{xu}, \ t \geq 0, \tag{9}$$

where $\mathcal{R}_\Sigma\left(x, \{u_i\}_{i=0}^{t-1}\right) \subseteq \mathbb{R}^n$, $u_t \in \mathbb{R}^m$, and the pair $\left(\mathcal{R}_\Sigma\left(x, \{u_i\}_{i=0}^{t-1}\right), u_t\right) \subseteq \mathbb{R}^n \times \mathbb{R}^m$. By Theorem 3.1, the above constraints characterize the states and input sequences within an implicit RCIS of Σ, such that the pair (x, u) stays inside the safe set S_{xu} indefinitely. Notice that (9) defines an *infinite* number of constraints in general. In this section, we investigate under what conditions we can reduce the above constraints into a *finite* number and compute them explicitly. Then, we use these constraints to construct the promised implicit RCIS.

Definition 8 (Eventually periodic behavior): Consider two integers $\tau \in \mathbb{N} \cup \{0\}$ and $\lambda \in \mathbb{N}$. A control input u_t follows an *eventually periodic behavior* if:

$$u_{t+\lambda} = u_t, \ \text{for all } t \geq \tau. \tag{10}$$

We call τ the *transient* and λ the *period*.

Proposition 3.2 (Finite reachability constraints): Consider a DTLS Σ satisfying Assumption 1. If the input u_t follows an eventually periodic behavior with transient $\tau \in \mathbb{N} \cup \{0\}$ and period $\lambda \in \mathbb{N}$, then the infinite constraints in (9) are reduced to a finite number of constraints.

Proof: Under Assumption 1 the matrix A is nilpotent with nilpotency index ν. Consequently, given (5), the reachable set from a state x for $t \geq \nu$ depends only on the past ν inputs. We abuse notation to write $\mathcal{R}_\Sigma\left(\{u_i\}_{i=0}^{t-1}\right)$ and omit the state x to denote dependency only on the inputs. Then, for $t \geq \nu + \tau$:

$$\begin{aligned} \mathcal{R}_\Sigma\left(\{u_i\}_{i=0}^{t-1}\right) &= \sum_{i=1}^{\nu} A^{i-1} B u_{t-i} + \overline{W}_\infty \\ &\overset{(10)}{=} \sum_{i=1}^{\nu} A^{i-1} B u_{t+\lambda-i} + \overline{W}_\infty = \mathcal{R}_\Sigma\left(\{u_i\}_{i=0}^{t+\lambda-1}\right). \end{aligned}$$

Therefore, under inputs with eventually periodic behavior the reachability constraints repeat themselves after $t = \nu + \tau + \lambda$. As a result, we can split the constraints in (9) as:

$$\left(\mathcal{R}_\Sigma\left(x, \{u_i\}_{i=0}^{t-1}\right), u_t\right) \subseteq S_{xu}, \ t = 0, \ldots, \nu - 1, \tag{11}$$

$$\left(\mathcal{R}_\Sigma\left(\{u_i\}_{i=0}^{t-1}\right), u_t\right) \subseteq S_{xu}, \ t = \nu, \ldots, \nu + \tau + \lambda - 1. \tag{12}$$

The above suggests that $\left(\mathcal{R}_\Sigma\left(x, \{u_i\}_{i=0}^{t-1}\right), u_t\right) \subseteq S_{xu}$ for all $t \geq 0$ can be replaced with only $\nu + \tau + \lambda$ constraints. ■

Proposition 3.2 provides a finite representation of the constraints in (9) under the eventually periodic input behavior in (10). The next question we address concerns characterizing the classes of policies that guarantee the behavior in (10).

C. Implicit robust controlled invariant sets in closed-form

Recall that our goal is to derive a closed-form expression for an implicit RCIS of Σ, which is essentially the Maximal RPIS of the companion system Σ_{xv} by Theorem 3.1. So far we proved that, in general, inputs with eventually periodic behavior result in finite reachability constraints. Clearly, the parameterized input in (7) follows an eventually periodic behavior as in (10) if:

$$P^t = P^{t+\lambda},\ t \geq \tau, \tag{13}$$

i.e., P is an *eventually periodic* matrix with transient τ and period λ.

Proposition 3.3 (Structure of eventually periodic matrices): Any *eventually periodic* matrix $P \in \mathbb{R}^{n\times n}$ has eigenvalues that are either 0 or λ-th roots of unity. If $\tau \neq 0$, i.e., P is not purely periodic, then P has at least one 0 eigenvalue with algebraic multiplicity equal to τ and geometric multiplicity equal to 1. If $P^\tau \neq 0$, i.e., P is not nilpotent, then P has at least one eigenvalue that is a λ-th root of unity.

Proof: Let $\mathrm{v} \neq 0$ be an eigenvector of P and δ the corresponding eigenvalue, i.e., $P\mathrm{v} = \delta\mathrm{v}$. Then, (13) for $t \geq \tau$ yields:

$$\begin{aligned} P^t = P^{t+\lambda} &\Rightarrow P^t\mathrm{v} = P^{t+\lambda}\mathrm{v} \Leftrightarrow \delta^t\mathrm{v} = \delta^{t+\lambda}\mathrm{v} \\ &\overset{\mathrm{v}\neq 0}{\Leftrightarrow} \delta^t = \delta^{t+\lambda} \Leftrightarrow \delta^t\left(1-\delta^\lambda\right) = 0, \end{aligned}$$

that is, the eigenvalues δ of P are only 0 or λ-th roots of unity.

Consider now the Jordan normal form $P = MJM^{-1}$ [19]. This form is unique up to the order of the Jordan blocks, and $P^t = MJ^tM^{-1}$. Without loss of generality, we write:

$$J = \begin{bmatrix} J_1 & \mathbb{0} \\ \mathbb{0} & J_2 \end{bmatrix},$$

where J_1 is the Jordan block corresponding to the eigenvalues of P that are 0, and J_2 is the Jordan block corresponding to the eigenvalues of P that are the λ-th roots of unity. Thus, J_1 is nilpotent. Then, when $\tau \neq 0$, equality (13) is equivalent to:

$$P^t = P^{t+\lambda} \Leftrightarrow MJ^tM^{-1} = MJ^{t+\lambda}M^{-1},\ t \geq \tau.$$

Matrix J_1 vanishes in exactly τ steps, i.e., $J_1^\tau = 0$ and $J_1^t \neq 0$, for $t < \tau$. This implies that P has at least one 0 eigenvalue with algebraic multiplicity equal to τ and geometric multiplicity equal to 1, but no 0 eigenvalues of geometric multiplicity 1 and algebraic multiplicity greater than τ.

Moreover, when P is not nilpotent, i.e., $P^\tau \neq \mathbb{0}$, for $t \geq \tau$:

$$\begin{aligned} J^t = J^{t+\lambda} &\overset{J_1^t=0, t\geq\tau}{\Leftrightarrow} \begin{bmatrix} \mathbb{0} & \mathbb{0} \\ \mathbb{0} & J_2^t \end{bmatrix} = \begin{bmatrix} \mathbb{0} & \mathbb{0} \\ \mathbb{0} & J_2^{t+\lambda} \end{bmatrix} \\ &\Leftrightarrow J_2^t = J_2^{t+\lambda}. \end{aligned}$$

Thus, P has at least one eigenvalue that is a λ-th root of unity. ■

Corollary 3.4: The class of matrices described by Proposition 3.3 that satisfies (13) can be written, up to a similarity transformation, in the following form:

$$P = \begin{bmatrix} N & Q \\ \mathbb{0} & R \end{bmatrix}, \tag{14}$$

where N is a nilpotent matrix with nilpotency index τ, R is a matrix whose eigenvalues are all λ-th roots of unity, i.e., $R^\lambda = \mathbb{I}$, and Q is an arbitrary matrix.

Proposition 3.3 and Corollary 3.4 guide the designer to effortlessly select matrix P via its eigenvalues or its submatrices. Moreover, it is reasonable to select the projection matrix H to be *surjective* in order to obtain a non-trivial input in (7).

We now show that we can compute the desired *closed-form* expression for an implicit RCIS parameterized by collections of *eventually periodic* input sequences.

Theorem 3.5 (Closed-form implicit RCIS): Consider a DTLS Σ and a safe set S_{xu} for which Assumption 1 holds. Select an eventually periodic matrix $P \in \mathbb{R}^{mq\times mq}$ and a surjective projection matrix $H \in \mathbb{R}^{m\times mq}$. An *implicit RCIS* for Σ within S_{xu}, denoted by $\mathcal{C}_{xv}$, is defined by the constraints:

$$\begin{aligned} \left(A^t x + \sum_{i=1}^{t} A^{i-1}BHP^{t-i}v, HP^t v\right) &\subseteq S_{xu} - \overline{W}_t \times \{0\}, \\ &t = 0, \ldots, \nu-1, \\ \left(\sum_{i=1}^{\nu} A^{i-1}BHP^{t-i}v, HP^t v\right) &\subseteq S_{xu} - \overline{W}_\infty \times \{0\}, \\ &t = \nu, \ldots, \nu+\tau+\lambda-1. \end{aligned} \tag{15}$$

That is, the set $\mathcal{C}_{xv} \subset \mathbb{R}^n \times \mathbb{R}^{mq}$:

$$\mathcal{C}_{xv} = \{(x,v) \in \mathbb{R}^n \times \mathbb{R}^{mq} \mid (x,v) \text{ satisfy (15)}\}, \tag{16}$$

is computed in closed-form. Moreover, $\mathcal{C}_{xv}$ is the Maximal RPIS of the companion dynamical system in (8).

Proof: By Proposition 3.2, the set $\mathcal{C}_{xv}$ defined by (15) in closed-form satisfies the constraints in (9) and, thus, is the Maximal RPIS of the companion system Σ_{xv} in S_{xv}. Then, by Theorem 3.1, C_{xv} is an implicit RCIS of Σ in S_{xu}. ■

Theorem 3.5 provides an implicit RCIS, $\mathcal{C}_{xv}$, in closed-form. This set defines pairs of states and finite input sequences such that the state remains in the safe set indefinitely.

Remark 2 (On the choice of input behavior): Notice that the open-loop eventually periodic policy used to parameterize the implicit RCIS is only a means towards its computation in closed-form. In practice, after computing an RCIS, we can use any controller appropriate for the task at hand. This is illustrated in our case studies in Section VII, where the controller of the system is independent of the RCIS implicit representation. For instance, once an RCIS is available one defines a closed-loop non-periodic and memoryless controller $K : \mathbb{R}^n \to \mathbb{R}^m$ for which $Ax + BK(x)$ belongs to the RCIS when x is an element of the RCIS.

Corollary 3.6 (Computation of explicit RCIS): By selecting an eventually periodic matrix $P \in \mathbb{R}^{mq\times mq}$ and a projection matrix $H \in \mathbb{R}^{m\times mq}$, one computes an *explicit* RCIS $C_x = \pi_n(\mathcal{C}_{xv})$ with a single projection step.

The size of the lifted space leads to a trade-off: on the one hand it can result to larger RCISs, as we detail in the next

section, but on the other it requires more effort if the optional projection step is taken.

IV. A HIERARCHY OF CONTROLLED INVARIANT SETS

Our main result, Theorem 3.5, provides a closed-form expression for an implicit RCIS, $\mathcal{C}_{xv}$, with constraints on the state of the system, x, and on a finite sequence of inputs, v. The resulting sets depend on the choice of the eventually periodic matrix P in (6) and the projection matrix H.

In this section, we show how to *systematically* construct a sequence of RCISs that form a *hierarchy*, i.e., a non-decreasing sequence. Our goal is to provide a closed-form expression for the implicit RCISs corresponding to this hierarchy. Towards this, we identify special forms of matrices P and H.

Definition 9 ((τ,λ)-lasso sequence): Consider two integers $\tau \in \mathbb{N}\cup\{0\}$ and $\lambda \in \mathbb{N}$, and let $q = \tau+\lambda$. The control input u generated by the dynamical system Σ_C in (6) forms a (τ,λ)-*lasso sequence* with respect to the inputs v, if:

$$\begin{aligned} P &= P_{(\tau,\lambda)} = \text{blkdiag}\left(\bar{P},\dots,\bar{P}\right) \in \mathbb{R}^{mq\times mq},\\ H &= H_{(\tau,\lambda)} = \text{blkdiag}\left(\bar{H},\dots,\bar{H}\right) \in \mathbb{R}^{m\times mq}, \end{aligned} \tag{17}$$

with m blocks each and $\bar{P}$, $\bar{H}$ defined as:

$$\begin{aligned} \bar{P} &= \begin{bmatrix} \mathbb{0} & & \mathbb{I} & & \\ 0 & \cdots & 1 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{q\times q},\\ \bar{H} &= \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \in \mathbb{R}^{1\times q}. \end{aligned} \tag{18}$$

In the last row of $\bar{P}$ the 1 occurs at the τ-th position. It is easy to verify that $P_{(\tau,\lambda)}$ in (17) is of the form (14). A (τ,λ)-lasso sequence has a transient of τ inputs followed by periodic inputs with period λ.

We utilize the (τ,λ)-lasso sequence to formalize a hierarchy of RCISs with a single decision parameter q.

Definition 10 (Lassos of same length): Select $q \in \mathbb{N}$. Define the set of all pairs $(\tau,\lambda) \in \mathbb{N}\cup\{0\}\times\mathbb{N}$ corresponding to lassos of length q as:

$$\Theta_q = \{(\tau,\lambda) \in \mathbb{N}\cup\{0\}\times\mathbb{N} \mid \tau+\lambda = q\}. \tag{19}$$

The cardinality of Θ_q is exactly q.

The next result provides a way to systematically construct implicit RCISs in closed-form such that the corresponding explicit RCISs form a hierarchy.

Theorem 4.1 (Hierarchy of RCISs): Consider a DTLS Σ and a safe set S_{xu} for which Assumption 1 holds, and select an integer $q \in \mathbb{N}$. Given q, the set $\mathcal{C}_{xv,\mathrm{q}} \subset \mathbb{R}^n \times \mathbb{R}^{mq}$:

$$\mathcal{C}_{xv,\mathrm{q}} = \bigcup_{(\tau,\lambda)\in\Theta_q} \mathcal{C}_{xv,(\tau,\lambda)}, \tag{20}$$

is the implicit RCIS induced by the q-level of the hierarchy, where each $\mathcal{C}_{xv,(\tau,\lambda)}$ is computed in closed-form in (16) with P and H as in (17). In addition, the explicit RCIS:

$$\begin{aligned} \mathcal{C}_{x,\mathrm{q}} &= \pi_n\left(\mathcal{C}_{xv,\mathrm{q}}\right) = \bigcup_{(\tau,\lambda)\in\Theta_q} \pi_n\left(\mathcal{C}_{xv,(\tau,\lambda)}\right)\\ &= \bigcup_{(\tau,\lambda)\in\Theta_q} \mathcal{C}_{x,(\tau,\lambda)}, \end{aligned} \tag{21}$$

corresponding to the q-level of the hierarchy contains any RCIS lower in the hierarchy, i.e.:

$$\mathcal{C}_{x,\mathrm{q}} \supseteq \mathcal{C}_{x,\mathrm{q}'}, \text{ for any } q,q' \in \mathbb{N} \text{ with } q' < q. \tag{22}$$

Proof: First, the sets $\mathcal{C}_{xv,\mathrm{q}}$ and $\mathcal{C}_{x,\mathrm{q}}$ are implicit and explicit RCISs respectively as the unions of, implicit and explicit, RCISs by Proposition 2.2. Next we prove (22) for the case of q and $q+1$, while the more general statement follows by a simple induction argument.

For any $\lambda \in \mathbb{N}$ such that $(\tau,\lambda) \in \Theta_q$, we have by (19) that $(\tau+1,\lambda) \in \Theta_{q+1}$. It is easy to show that:

$$\mathcal{C}_{x,(\tau+1,\lambda)} \supseteq \mathcal{C}_{x,(\tau,\lambda)}, \tag{23}$$

as $\mathcal{C}_{x,(\tau,\lambda)}$ contains the set of states rendered invariant by a (τ,λ)-lasso sequence of inputs, and any (τ,λ)-lasso sequence is also a $(\tau+1,\lambda)$-lasso sequence. Hence, by (21):

$$\begin{aligned} \mathcal{C}_{x,(\mathrm{q}+1)} &= \bigcup_{(\tau,\lambda)\in\Theta_{q+1}} \mathcal{C}_{x,(\tau,\lambda)} = \left(\bigcup_{(\tau,\lambda)\in\Theta_q} \mathcal{C}_{x,(\tau+1,\lambda)}\right) \bigcup \mathcal{C}_{x,(0,q+1)}\\ &\overset{(23)}{\supseteq} \left(\bigcup_{(\tau,\lambda)\in\Theta_q} \mathcal{C}_{x,(\tau,\lambda)}\right) \bigcup \mathcal{C}_{x,(0,q+1)} \overset{(21)}{=} \mathcal{C}_{x,\mathrm{q}} \bigcup \mathcal{C}_{x,(0,q+1)}. \end{aligned}$$

The above entails that $\mathcal{C}_{x,(\mathrm{q}+1)} \supseteq \mathcal{C}_{x,\mathrm{q}}$. ∎

Corollary 4.2: Using the standard big-M formulation, the implicit RCIS $\mathcal{C}_{xv,\mathrm{q}}$ can be expressed as a projection of a higher-dimensional polytope:

$$\mathcal{C}_{xv\zeta,\mathrm{q}} = \left\{ (x,v,\zeta) \,\middle|\, \sum_{i=1}^{q} \zeta_i = 1, G_i(x,v) \le f_i + (1-\zeta_i)M\mathbb{1} \right\}, \tag{24}$$

where $\zeta \in \{0,1\}^q$, G_i and f_i describe each of the q polytopes $\mathcal{C}_{xv,(\tau,\lambda)}$ in (20), and $M \in \mathbb{R}_+$ is sufficiently large. The set $\mathcal{C}_{xv\zeta,\mathrm{q}}$ is a polytope in $\mathbb{R}^n \times \mathbb{R}^{mq} \times \{0,1\}^q$, and its projection on $\mathbb{R}^n \times \mathbb{R}^{mq}$ is exactly the union in (20), while its projection on $\mathbb{R}^n$ is exactly the explicit RCIS in (21).

Theorem 4.1 defines the promised hierarchy and provides an implicit RCIS for each level of the hierarchy that can also be computed in closed-form in (24) at the cost of an additional lift. Fig. 1 illustrates the relation in (22), that is, how the sets induced by each hierarchy level contain the ones induced by lower hierarchy levels.

Remark 3 (Convex hierarchy): We can replace the union in (20) by the convex hull $\text{conv}\left(\bigcup_{(\tau,\lambda)\in\Theta_q} \mathcal{C}_{xv,(\tau,\lambda)}\right)$. Then, in an analogous manner, all the above results go through resulting in a *hierarchy of convex RCISs*. Similarly to (24), by standard set-lifting techniques, one obtains a closed-form expression for the convex hull.

Remark 4 (Partial hierarchies without union): The proposed hierarchy involves handling a union of sets. However, one might prefer to avoid unions of sets and rather use a single convex set. As each implicit RCIS $\mathcal{C}_{xv,(\tau,\lambda)}$ involved in the hierarchy is computed in closed-form by Theorem 3.5, we provide two more refined guidelines for obtaining larger RCISs, based on the choice of (τ,λ):

1) Given any $\lambda \in \mathbb{N}$, it holds that $\mathcal{C}_{x,(\tau+1,\lambda)} \supseteq \mathcal{C}_{x,(\tau,\lambda)}$ for any $\tau \in \mathbb{N}\cup\{0\}$.

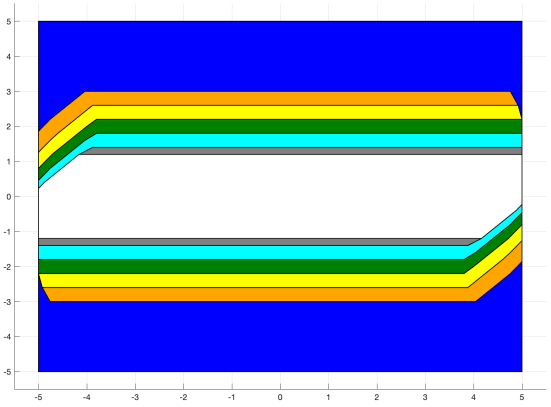


Fig. 1: RCIS corresponding to $q = 1$ (white), $q = 2$ (gray), $q = 3$ (teal), $q = 4$ (green), $q = 5$ (yellow), and $q = 6$ (orange) for a double integrator. Safe set in blue.

2) Given any $\tau \in \mathbb{N} \cup \{0\}$, it holds that $\mathcal{C}_{x,(\tau,\lambda)} \supseteq \mathcal{C}_{x,(\tau,\lambda')}$ for any $\lambda, \lambda' \in \mathbb{N}$ such that $\lambda = k\lambda'$, $k \in \mathbb{N}$, i.e., λ is a multiple of λ', see [3, Section 4.6] when $\tau = 0$.

The above can direct the designer in search of larger RCISs that are based on a single implicit RCIS.

V. Implicit invariant sets in practice: controlled invariant sets in one move

Using the proposed results, one has the option to project the implicit RCIS back to the original space and obtain an explicit RCIS as proposed in the two-move approach [1]–[3]. However, the required projection from a higher dimensional space becomes the bottleneck of this approach.

One of the goals of this manuscript is to establish that in a number of key control problems explicit knowledge of the RCIS is not required and the implicit RCIS suffices. We show how the proposed methodology can be used *online* as the implicit RCIS which admits a closed-form expression.

A. Extraction of admissible inputs

For many applications in this section, we need to extract a set of admissible inputs of the RCIS $\pi_n(C_{xv})$ at a given state x, i.e, $\mathcal{A}(x)$ as given in Definition 4. Given only the implicit RCIS C_{xv}, we provide here three linear encodings of $\mathcal{A}(x)$ or its nonempty subsets.

1) The first linear encoding of $\mathcal{A}(x)$ is given by the polytope:

$$\begin{aligned}\mathcal{U}_1(x) = \big\{(u, v_{1:N}) \in \mathbb{R}^{(1+Nq)m} \,\big|\, (x, u) \in S_{xu}, \\ (Ax + Bu + Ew_i, v_i) \in \mathcal{C}_{xv}, \forall i \in [N]\big\},\end{aligned} \tag{25}$$

where $v_{1:N}$ denotes the vector $(v_1, v_2, \cdots, v_N)$. It follows that $\pi_m(\mathcal{U}_1(x)) = \mathcal{A}(x)$.

2) The second linear encoding is:

$$\mathcal{U}_2(x) = \big\{v \in \mathbb{R}^{qm} \,\big|\, (x, v) \in \mathcal{C}_{xv}\big\}, \tag{26}$$

with H and P as in (6). Note that $\mathcal{U}_2(x)$ is the slice of $\mathcal{C}_{xv}$ at x and is nonempty for $x \in \pi_n(\mathcal{C}_{xv})$. Then, the linear transformation $H\mathcal{U}_2(x)$ is a nonempty subset of $\mathcal{A}(x)$.

3) Finally, define the polytope:

$$\begin{aligned}\mathcal{U}_3(x) = \big\{(u, v) \in \mathbb{R}^{(1+q)m} \,\big|\, (x, u) \in S_{xu}, \\ (Ax + Bu + Ew_i, v) \in \mathcal{C}_{xv}, \forall i \in [N]\big\},\end{aligned} \tag{27}$$

where $w_i \in \mathcal{V}$ with $\mathcal{V}$ the vertices of W. It follows that $\pi_m(\mathcal{U}_3(x)) \subseteq \mathcal{A}(x)$. It is easy to check that $(Hv, Pv) \in \mathcal{U}_3(x)$ for all $v \in \mathcal{U}_2(x)$, which implies that $\mathcal{U}_3(x)$ is guaranteed to be nonempty for any $x \in \pi_n(\mathcal{C}_{xv})$.

All three linear encodings are easily computed online given $\mathcal{C}_{xv}$. Moreover, it holds that:

$$H\mathcal{U}_2(x) \subseteq \pi_m(\mathcal{U}_3(x)) \subseteq \pi_m(\mathcal{U}_1)(x) = \mathcal{A}(x).$$

That is, $\mathcal{U}_2(x)$ is the most conservative encoding, while $\mathcal{U}_1(x)$ is the least conservative one. However, $\mathcal{U}_2$ is of lower dimension, while $\mathcal{U}_1$ has the highest dimension. More conservative encodings are easier to compute. Depending on the available compute, a user can select the most appropriate encoding.

B. Supervision of a nominal controller

In many scenarios, when synthesizing a controller for a plant, the objective is to meet a performance criterion while always satisfying a safety requirement. This gives rise to the problem of *supervision*.

Problem 1 (Supervisory Control): Consider a system Σ, a safe set S_{xu}, and a nominal controller that meets a performance objective. The supervisory control problem asks at each time step to evaluate if, given the current state, the input $\tilde{u}$ from the nominal controller keeps the next state of Σ in the safe set. If not, *correct* $\tilde{u}$ by selecting an input that does so.

To solve Problem 1 one has to guarantee *at every step* that the pairs of states and inputs respect the safe set S_{xu}. A natural way to do so is by using an RCIS. The supervision framework operates as follows. Given an RCIS $\mathcal{C}$, assume that the initial state of Σ lies in $\mathcal{C}$. The nominal controller provides an input $\tilde{u}$ to be executed by Σ. If $\tilde{u} \in \mathcal{A}(x)$, then its execution is allowed. Else $\tilde{u}$ is corrected by selecting an input $u_{safe} \in \mathcal{A}(x)$. Existence of u_{safe} is guaranteed in any RCIS by Definition 3.

In practice an explicit RCIS is not needed. One can exploit the three linear encodings of admissible inputs from the proposed implicit RCISs to perform supervision. Furthermore, the nominal controller can be designed independently of the implicit RCIS parameterization. Consider an implicit RCIS $\mathcal{C}_{xv}$ for Σ within S_{xu}, as in Theorem 3.5. Then supervision of an input $\tilde{u}$ is performed by solving the following QP:

$$\begin{aligned}\min_{u,v} \quad & ||u - \tilde{u}||_2^2 \\ \text{s.t.} \quad & (x, u) \in S_{xu} \\ & (Ax + Bu + Ew, v) \in \mathcal{C}_{xv}, \forall w \in W\end{aligned} \tag{28}$$

Notice that the feasible domain of the QP in (28) is equal to the third linear encoding $\mathcal{U}_3(x)$ of admissible inputs; similar QPs are easily formulated with the feasible domain being $\mathcal{U}_1(x)$ or $\mathcal{U}_2(x)$. By solving optimization problem (28) we compute the *minimally intrusive safe input*.

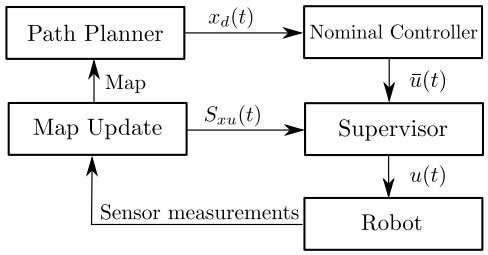


Fig. 2: The overall safe online planning framework.

C. Safe online planning

Based on the discussed supervision framework, we utilize the proposed implicit RCIS to enforce safety constraints in online planning tasks. The goal here is to navigate a robot through unknown environments without collision with any obstacles. The map is initially unknown, and it is built and updated online based on sensor measurements, such as LiDAR. The robot must only operate in the detected obstacle-free region. To ensure this, given a path planning algorithm and a tracking controller, we supervise the controller inputs based on the implicit RCIS. The overall framework is shown in Figure 2.

The safe set for the robot imposes bounds on states and inputs, which do not change over time, and also constraints, e.g., on the robot's position, which are given by the obstacle-free region in the current map. As the detected obstacle-free region expands over time, the corresponding part of the safe set does as well. Thus, differently from Section V-B, we have a time-varying safe set $S_{xu}(t)$ satisfying $S_{xu}(t) \subseteq S_{xu}(t+1)$, $t \geq 0$. As the implicit RCIS is constructed in closed-form, we can generate a new implicit RCIS $\mathcal{C}_{xv}(t)$ for each $S_{xu}(t)$. Then, at each time step t, for any $t' \leq t$, we supervise the nominal input $\tilde{u}(t)$ by solving the optimization problem:

$$\mathcal{P}(t,t'): \begin{aligned} \min_{u,v} \quad & ||u-\tilde{u}||_2^2 \\ \text{s.t.} \quad & (x,u) \in S_{xu}(t) \\ & (Ax+Bu+Ew,v) \in \mathcal{C}_{xv}(t'), \forall w \in W. \end{aligned}$$

As $S_{xu}(t) \subseteq S_{xu}(t+1)$, $\mathcal{C}_{xv}(t')$ is a valid implicit RCIS in $S_{xu}(t)$ for all $t \geq t'$. Thus, as long as $\mathcal{P}(t,t')$ is feasible, the optimizer v^* of $\mathcal{P}(t,t')$ is a safe input that guarantees the next state lies in the RCIS. Furthermore, if $P(t,t')$ is feasible, by definition of RCIS, $P(t+1,t')$ is also feasible. Thus, if $P(0,0)$ is feasible, for all $t>0$, there exists $t' \leq t$ such that $P(t,t')$ is feasible. That is, the recursive feasibility of $P(t,t')$ is guaranteed. In practice, to take advantage of the latest map, we always select t' to be the latest time instant t^* for which $P(t,t^*)$ is feasible.

To summarize, at each time step, we first construct the implicit RCIS $\mathcal{C}_{xv}(t)$ based on the current map. Then, given the state and nominal control input, we solve $\mathcal{P}(t,t^*)$ to obtain the minimally intrusive safe input. This input guarantees that the state of the robot stays within $S_{xu}(t)$ for all $t \geq 0$, provided that $\mathcal{P}(0,0)$ is feasible.

D. Safe hyper-boxes

For high dimensional systems, the exact representation of an RCIS $\mathcal{C}_x$ can be a set of thousands of linear inequalities. This reduces insight as it is quite difficult to clearly identify regions of each state that lie within the RCIS. In contrast, hyper-boxes are easy to grasp in any dimension and immediately provide information about the regions of states they contain. Based on this, we explore how implicit RCISs can be used to find hyper-boxes that can be considered *safe* in the following sense.

Definition 11 (Safe hyper-boxes): Consider a system Σ, a safe set S_x, and the Maximal RCIS $\mathcal{C}_{max} \subseteq S_x$. Define a hyper-box $\mathcal{B} = [\underline{b}_1, \overline{b}_1] \times \cdots \times [\underline{b}_n, \overline{b}_n] = [\underline{b}, \overline{b}] \subset \mathbb{R}^n$. We call a hyper-box $\mathcal{B}$ *safe* if $\mathcal{B} \subseteq \mathcal{C}_{max}$.

To simplify the presentation we only consider state constraints, S_x, instead of S_{xu}. Notice that by Definition 11, a safe hyper-box is not necessarily invariant. A safe hyper-box $\mathcal{B}$ entails the guarantee that the trajectory starting therein can remain in S_x forever, but not necessarily within $\mathcal{B}$. We now aim to address the following problem.

Problem 2: Find the largest[1] *safe* hyper-box $\mathcal{B}$ within $\mathcal{C}_x$.

A hyper-box $\mathcal{B}$ can be described by a pair of vectors $\left(\underline{b}, \overline{b}\right) \in \mathbb{R}^n \times \mathbb{R}^n$. Then, using similar arguments to Section III, we compute in closed-form an implicit RCIS $\mathcal{C}_{\mathcal{B}}$ characterizing all hyper-boxes $\left(\underline{b}, \overline{b}\right)$ that remain in S_x under eventually periodic inputs. The eventually periodic inputs are given by a vector $v \in \mathbb{R}^{mq}$ with $q = \tau + \lambda$. Then, the set $\mathcal{C}_{\mathcal{B}}$ lives in $\mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{mq}$ and is described by:

$$A^t \left[\underline{b}, \overline{b}\right] + \sum_{i=1}^{t} A^{i-1} B H P^{t-i} v \subseteq S_x - \overline{W}_t, t = 0, \ldots, \nu - 1,$$

$$\sum_{i=1}^{\nu} A^{i-1} B H P^{t-i} v \subseteq S_x - \overline{W}_\infty, t = \nu, \ldots, \nu + q - 1.$$

The above constraints can all be written as linear inequalities in $\left(\underline{b}, \overline{b}, v\right) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{mq}$. Then, the implicit RCIS $\mathcal{C}_{\mathcal{B}}$ is a polytope and one solves Problem 2 by the following convex optimization program:

$$\begin{aligned} \max_{\left(\underline{b}, \overline{b}, v\right)} \quad & \gamma\left(\overline{b} - \underline{b}\right) \\ \text{s.t.} \quad & \left(\underline{b}, \overline{b}, v\right) \in \mathcal{C}_{\mathcal{B}}, \end{aligned}$$

where $\gamma(y) = \left(\Pi_{i=1}^n y_i\right)^{\frac{1}{n}}$ is the geometric mean function, which is used as a heuristic for the volume of the hyper-box. Function γ is concave, and maximizing a concave function can be cast as a convex minimization problem [7].

Remark 5 (Invariant and recurrent hyper-boxes): Two special cases of the above are *invariant* hyper-boxes, when $\tau = 0$, $\lambda = 1$, see also [1], and *recurrent* hyper-boxes, when $\tau = 0$, $\lambda > 0$, see also [2], [3].

A related question to Problem 2 is to evaluate if a proposed hyper-box is safe. This is of interest when evaluating whether the initial condition of a problem or an area around a configuration point x_c where the system is required to operate is safe. If both the above are modeled by hyper-boxes $(\underline{b}, \overline{b})$, we can simply ask whether there exists a v, such that $(\underline{b}, \overline{b}, v) \in \mathcal{C}_{\mathcal{B}}$. Similarly, more complicated questions can be formulated, e.g., to find the largest safe box around a configuration point.

[1]The largest, as measured by volume, hyper-box within a set might not be unique. We choose a heuristic for maximizing the volume of a set that yields a well-defined convex optimization problem. Hence, the term "largest" refers to the heuristic used.

Remark 6 (Complexity when using implicit RCISs): In this section we showed how several key problems in control are solved without the need of projection and of an explicit RCIS, which results in extremely efficient computations since the implicit RCISs are computed in closed-form. The decision to be made is the size of the lift, i.e., the length of the input sequence. From a computational standpoint, this choice is only limited by how large an optimization problem one affords solving given the application.

VI. Performance bound for the proposed method

Numerical examples, to be presented later, will show that the projection of the proposed implicit RCIS onto the original state-space can coincide with the Maximal RCIS. However, this is not always the case. When there is a gap between our projected set and the Maximal RCIS, one may wonder if that gap is fundamental to our method. In other words, can we arbitrarily approximate the Maximal RCIS with the projection of our implicit RCIS by choosing better P and H matrices?

In this section we aim to answer the above question and provide insights into the completeness of our method. Given (4), define the nominal DTLS $\overline{\Sigma}$ and the nominal safe set $\overline{S}_{xu}$:

$$\overline{\Sigma} \ : \ x^+ = Ax + Bu, \tag{29}$$

$$\overline{S}_{xu} = S_{xu} - \overline{W}_\infty \times \{0\}, \tag{30}$$

where A and B are the same as in (1). Let $\overline{\mathcal{C}}_{\max}$ be the Maximal CIS of the nominal system $\overline{\Sigma}$ within $\overline{S}_{xu}$ and define:

$$\begin{aligned}\mathcal{C}_{outer,\nu} = \Big\{ x \in \mathbb{R}^n \ \Big| \ & \exists \{u_i\}_{i=0}^{\nu-1} \in \mathbb{R}^{m\nu}, \\ & \Big(\mathcal{R}_\Sigma\Big(x, \{u_i\}_{i=0}^{t-1}\Big), u_t\Big) \subseteq S_{xu}, t = 0, \dots, \nu-1, \\ & \mathcal{R}_\Sigma\Big(x, \{u_i\}_{i=0}^{\nu-1}\Big) \subseteq \overline{\mathcal{C}}_{\max} + \overline{W}_\infty \Big\},\end{aligned} \tag{31}$$

where ν is the nilpotency index of A.

Proposition 6.1: $\mathcal{C}_{outer,\nu}$ is an RCIS of Σ within S_{xu}.

Proof: In this proof, we use the order cancellation lemma, as a special case of [13, Thm. 4].

Lemma 6.2: Let $X, Y \subset \mathbb{R}^n$ be two closed convex sets with Y bounded. A point $x \in \mathbb{R}^n$ is in X if and only if $x + Y \subseteq X + Y$.

To prove that $\mathcal{C}_{outer,\nu}$ is an RCIS, we show that for any $x_0 \in \mathcal{C}_{outer,\nu}$, there exists u such that $(x_0, u) \in S_{xu}$ and for all $w \in W$, $Ax_0 + Bu + Ew \in \mathcal{C}_{outer,\nu}$. By definition of $\mathcal{C}_{outer,\nu}$, there exists a sequence $\{u_i\}_{i=0}^{\nu-1}$ that, along with x_0, satisfies the conditions in (31). We aim to show that u_0 in $\{u_i\}_{i=0}^{\nu-1}$ is a feasible choice for u. Given (31), the reachable set from x_0 at time ν is:

$$\mathcal{R}_\Sigma\Big(x_0, \{u_i\}_{i=0}^{\nu-1}\Big) = \sum_{i=0}^{\nu-1} A^{\nu-1-i} B u_i + \overline{W}_\infty \subseteq \overline{\mathcal{C}}_{\max} + \overline{W}_\infty,$$

with $\overline{W}_\infty$ and $\overline{\mathcal{C}}_{\max}$ convex and $\overline{W}_\infty$ bounded. By Lemma 6.2 we have that $\sum_{i=0}^{\nu-1} A^{\nu-1-i} B u_i \in \overline{\mathcal{C}}_{\max}$. Since $\overline{\mathcal{C}}_{\max}$ is controlled invariant within $\overline{S}_{xu}$ for the nominal DTLS $\overline{\Sigma}$, there exists u_ν such that:

$$\begin{aligned}&\left(\sum_{i=0}^{\nu-1} A^{\nu-1-i} B u_i, u_\nu\right) \in \overline{S}_{xu}, \\ &A\left(\sum_{i=0}^{\nu-1} A^{\nu-1-i} B u_i\right) + B u_\nu = \sum_{i=1}^{\nu} A^{\nu-1-i} B u_i \in \overline{\mathcal{C}}_{\max}.\end{aligned} \tag{32}$$

Consider any $w \in W$ and define $x_1 = Ax_0 + Bu_0 + Ew$:

$$\mathcal{R}_\Sigma(x_1, \{u_i\}_{i=1}^{\nu}) = \sum_{i=1}^{\nu} A^{\nu-1-i} B u_i + \overline{W}_\infty \subseteq \overline{\mathcal{C}}_{\max} + \overline{W}_\infty. \tag{33}$$

From (32) we have that:

$$(\mathcal{R}_\Sigma(x_1, \{u_i\}_{i=1}^{\nu-1}), u_\nu) \subseteq S_{xu}. \tag{34}$$

Finally, note that for $t = 0, \cdots, \nu-2$, we have:

$$\big(\mathcal{R}_\Sigma(x_1, \{u_i\}_{i=1}^{t}), u_{t+1}\big) \subseteq \big(\mathcal{R}_\Sigma(x_0, \{u_i\}_{i=0}^{t}), u_{t+1}\big) \subseteq S_{xu}. \tag{35}$$

From (33), (34), and (35) we verify that $x_1 \in \mathcal{C}_{outer,\nu}$. Thus, $\mathcal{C}_{outer,\nu}$ is an RCIS. ■

The following theorem shows that $\mathcal{C}_{outer,\nu}$ is an outer bound of the projection of the proposed implicit RCIS.

Theorem 6.3 (Outer bound on $\pi_n(\mathcal{C}_{xv})$): For a companion system Σ_{xv} as in (8), with arbitrary matrices P and H, let $\mathcal{C}_{xv}$ be an RPIS of Σ_{xv} within the companion safe set S_{xv}. The RCIS $\pi_n(\mathcal{C}_{xv})$ is a subset of $\mathcal{C}_{outer,\nu}$, that is $\pi_n(\mathcal{C}_{xv}) \subseteq \mathcal{C}_{outer,\nu}$.

Proof: Let $x \in \pi_n(\mathcal{C}_{xv})$. We show that $x \in \mathcal{C}_{outer,\nu}$. By definition of $\mathcal{C}_{xv}$, there exists a vector v such that:

$$\left(\mathcal{R}_\Sigma\left(x, \left\{HP^i v\right\}_{i=0}^{t-1}\right), HP^t v\right) \subseteq S_{xu}, \text{ for all } t \geq 0. \tag{36}$$

Define $u_t = HP^t v$. We want to verify that x and $\{u_i\}_{i=0}^{\nu-1}$ satisfy the two conditions in the definition of (31). The first condition is immediately satisfied by (36). It is left to show that $\mathcal{R}_\Sigma(x, \{u_i\}_{i=0}^{\nu-1}) \subseteq \overline{\mathcal{C}}_{\max} + \overline{W}_\infty$. That is:

$$\sum_{i=0}^{\nu-1} A^{\nu-1-i} B u_i + \overline{W}_\infty \subseteq \overline{\mathcal{C}}_{\max} + \overline{W}_\infty.$$

By Lemma 6.2, it is equivalent to prove that:

$$\overline{x} \equiv \sum_{i=0}^{\nu-1} A^{\nu-1-i} B u_i \in \overline{\mathcal{C}}_{\max}.$$

By (36), we have that for $t \geq 0$:

$$\begin{aligned}&\left(\sum_{i=0}^{\nu-1} A^{\nu-1-i} B u_{i+t} + \overline{W}_\infty, u_{v+t}\right) \subseteq S_{xu} \\ \Leftrightarrow &\left(\sum_{i=0}^{\nu-1} A^{\nu-1-i} B u_{i+t}, u_{\nu+t}\right) \in \overline{S}_{xu} \\ \Leftrightarrow &\left(\mathcal{R}_{\overline{\Sigma}}(\overline{x}, \{u_i\}_{i=\nu}^{\nu+t-1}), u_{\nu+t}\right) \in \overline{S}_{xu}\end{aligned} \tag{37}$$

According to (37), the control sequence $\{u_i\}_{i=\nu}^{\nu+t-1}$ guarantees that the trajectory of $\overline{\Sigma}$ starting at $\overline{x}$ stays within $\overline{S}_{xu}$ for all $t \geq 0$. Thus, $\overline{x}$ must belong to the Maximal CIS of Σ in $\overline{S}_{xu}$. That is, $\overline{x} \in \overline{\mathcal{C}}_{\max}$. ■

Note here that the set $\mathcal{C}_{outer,\nu}$, which serves as an outer bound for the set computed by our method, is as hard to compute as the Maximal RCIS. Given Theorem 6.3 we have:

$$\pi_n\left(\mathcal{C}_{xv}\right) \subseteq \mathcal{C}_{outer,\nu} \subseteq \mathcal{C}_{\max}. \tag{38}$$

Thus, the projection of our implicit RCIS can coincide with the Maximal RCIS, for appropriately selected matrices P and H, only if $\mathcal{C}_{outer,\nu} = \mathcal{C}_{\max}$ in (38). This potential gap between our approximation and the Maximal RCIS is due to the fact that our method uses open-loop forward reachability constraints under disturbances. Finally, the following theorem establishes weak completeness of our method.

Theorem 6.4 (Weak completeness): The set $\mathcal{C}_{outer,\nu}$ is nonempty, if and only if, there exist matrices P and H such that the corresponding implicit RCIS $\mathcal{C}_{xv}$ is nonempty. Specifically, $\mathcal{C}_{outer,\nu} \neq \emptyset$, if and only if, $\mathcal{C}_{xv,(0,1)} \neq \emptyset$, that is P and H are as in (17) with $(\tau, \lambda) = (0, 1)$.

Proof: We want to show that $\mathcal{C}_{outer,\nu}$ is nonempty if and only if $\mathcal{C}_{xv,(0,1)}$ is nonempty, where $\mathcal{C}_{xv,(0,1)}$ is defined in (20) with respect to system Σ and safe set S_{xu}.

Since $\pi_n\left(\mathcal{C}_{xv,(0,1)}\right) \subseteq \mathcal{C}_{outer,\nu}$, immediately nonemptyness of $\mathcal{C}_{xv,(0,1)}$ implies nonemptyness of $\mathcal{C}_{outer,\nu}$.

For the converse, suppose that $\mathcal{C}_{outer,\nu}$ is nonempty. Then $\overline{\mathcal{C}}_{\max}$ is nonempty. By [9, Theorem 12], we know that $\overline{\mathcal{C}}_{\max}$ is nonempty, if and only if, there exists a fixed point $x \in \overline{\mathcal{C}}_{\max}$ along with a u such that $(x, u) \in \overline{S}_{xu}$ and $Ax + Bu = x$. Also, note that $A\overline{W}_\infty + E\overline{W} = \overline{W}_\infty$. Thus, we have:

$$\begin{aligned} &(x + \overline{W}_\infty, u) \subseteq S_{xu}, \\ &A(x + \overline{W}_\infty) + Bu + E\overline{W} = x + \overline{W}_\infty. \end{aligned} \tag{39}$$

According to (39), for any $y \in x + \overline{W}_\infty$, we have $(y, u) \in S_{xu}$ and $Ay + Bu + E\overline{W} \subseteq x + \overline{W}_\infty$, which implies that $x + \overline{W}_\infty$ is an RCIS of Σ within S_{xu}. By the definition of $\mathcal{C}_{xv,(0,1)}$, it is easy to check that $(x + \overline{W}_\infty, u) \subseteq \mathcal{C}_{xv,(0,1)}$. Thus, $\mathcal{C}_{xv,(0,1)}$ is nonempty. ∎

Corollary 6.5 (Completeness in absence of disturbances): In the absence of disturbances, $\mathcal{C}_{outer,\nu} = \mathcal{C}_{max}$ and thus there exist P and H such that $\mathcal{C}_{xv}$ is nonempty, if and only if, $\mathcal{C}_{max}$ is nonempty. That is, the proposed method is complete.

The significance of Theorem 6.4 lies in allowing to quickly check nonemptiness of $\mathcal{C}_{outer,\nu}$ by computing $\mathcal{C}_{xv,(0,1)}$, which we can do in closed-form. Even though the gap between $\mathcal{C}_{outer,\nu}$ and $\mathcal{C}_{\max}$ is still an open question at the writing of this manuscript, we show that $\pi_n\left(\mathcal{C}_{xv}\right)$ can actually converge to its outer bound for a specific choice of H and P matrices.

Theorem 6.6 (Convergence to $\mathcal{C}_{outer,\nu}$): Assume that the disturbance set W contains 0, and the interior of $\overline{S}_{xu}$ contains a fixed point (x, u) of $\overline{\Sigma}$. There exist matrices H and P such that $\pi_n\left(\mathcal{C}_{xv}\right)$ approaches $\mathcal{C}_{outer,\nu}$. Specifically, if H and P are as in (17), by increasing τ in (17), $\pi_n\left(\mathcal{C}_{xv}\right)$ converges to $\mathcal{C}_{outer,\nu}$ in Hausdorff distance exponentially fast.

Proof: Without loss of generality, assume that the fixed point (x, u) of $\overline{\Sigma}$ in the interior of $\overline{S}_{xu}$ is the origin of the state-input space. We define a set operator $\mathcal{U}(\mathcal{C})$ that maps a subset $\mathcal{C}$ of $\mathbb{R}^n$ to a subset of $\mathbb{R}^{\nu m}$:

$$\mathcal{U}(\mathcal{C}) = \left\{ u_{0:\nu-1} \in \mathbb{R}^{\nu m} \;\middle|\; \sum_{i=1}^{\nu} A^{i-1} B u_{\nu-i} \in \mathcal{C} \right\}, \tag{40}$$

where $u_{0:\nu-1}$ denotes the vector $(u_0, u_1, \cdots, u_{\nu-1}) \in \mathbb{R}^{\nu m}$.

To maintain a streamlined presentation, we make the following claims that we prove in Appendix A.

Claim 1: The polytope $\mathcal{C}_{xv,0}$ contains the origin, where:

$$\begin{aligned} \mathcal{C}_{xv,0} = \{&(x, u_{0:\nu-1}) \in \mathbb{R}^{n+\nu m} \mid \\ &\left(\mathcal{R}_\Sigma\left(x, \{u_i\}_{i=0}^{t-1}\right), u_t\right) \subseteq S_{xu}, t = 0, \ldots, \nu - 1\}. \end{aligned}$$

Claim 2: For the set $\mathcal{C}_{outer,\nu}$ in (31) it holds that:

$$\mathcal{C}_{outer,\nu} = \pi_n(\mathcal{C}_{xv,max}), \tag{41}$$

where $\mathcal{C}_{xv,max} = \mathcal{C}_{xv,0} \cap (\mathbb{R}^n \times \mathcal{U}(\overline{\mathcal{C}}_{max}))$.

Claim 3: Let $\overline{\mathcal{C}}_{xv,(\tau,\lambda)}$ be the implicit CIS of the nominal system $\overline{\Sigma}$ within $\overline{S}_{xu}$ with H and P as in (17) and let $\overline{\mathcal{C}}_{x,(\tau,\lambda)} = \pi_n(\overline{\mathcal{C}}_{xv,(\tau,\lambda)})$. The implicit RCIS $\mathcal{C}_{xv,(\tau,\lambda)}$ of Σ within S_{xu} with H and P as in (17) satisfies:

$$\pi_n(\mathcal{C}_{xv,(\tau,\lambda)}) = \pi_n(\widehat{\mathcal{C}}_{xv,(\tau,\lambda)}), \text{ for any } \tau \geq \nu, \tag{42}$$

where $\widehat{\mathcal{C}}_{xv,(\tau,\lambda)} = \mathcal{C}_{xv,0} \cap (\mathbb{R}^n \times \mathcal{U}(\overline{\mathcal{C}}_{x,(\tau-\nu,\lambda)}))$.

Claim 4: There exist $c_0 > 0$, $a \in [0, 1)$, and some $\tau_1 \geq 0$ such that for any $\lambda \geq 1$ and for any $\tau \geq \tau_1$:

$$\overline{\mathcal{C}}_{x,(\tau,\lambda)} \supseteq (1 - c_0 a^\tau)\overline{\mathcal{C}}_{max}, \tag{43}$$

with τ_1 big enough such that $1 - c_0 a^{\tau_1} \geq 0$ and thereby the right hand side of (43) is well-defined.

We use these claims to prove the desired convergence rate. The operator $\mathcal{U}(\cdot)$ in (40) is linear with respect to scalar multiplication, i.e, $\mathcal{U}(\xi\mathcal{C}) = \xi\mathcal{U}(\mathcal{C})$, $\xi \geq 0$, and monotonic, i.e., $\mathcal{U}(\mathcal{C}_1) \supseteq \mathcal{U}(\mathcal{C}_2)$, $\mathcal{C}_1 \supseteq \mathcal{C}_2$. According to (43), for $\tau \geq \tau_1$:

$$\mathcal{U}(\overline{\mathcal{C}}_{x,(\tau,\lambda)}) \supseteq (1 - c_0 a^\tau)\mathcal{U}(\overline{\mathcal{C}}_{max}). \tag{44}$$

Note $\tau_0 = \nu + \tau_1$. By (42), for $\tau \geq \tau_0$:

$$\begin{aligned} \widehat{\mathcal{C}}_{xv,(\tau,\lambda)} &\supseteq \mathcal{C}_{xv,0} \cap (1 - c_0 a^{\tau-\nu})(\mathbb{R}^n \times \mathcal{U}(\overline{\mathcal{C}}_{max})) \\ &\supseteq (1 - c_0 a^{\tau-\nu})(\mathcal{C}_{xv,0} \cap (\mathbb{R}^n \times \mathcal{U}(\overline{\mathcal{C}}_{max}))) \\ &\supseteq (1 - c_0 a^{\tau-\nu})\mathcal{C}_{xv,max}. \end{aligned} \tag{45}$$

The second inclusion above holds since $0 \in \mathcal{C}_{xv,0}$ and thus $(1 - c_0 a^\tau)\mathcal{C}_{xv,0} \subseteq \mathcal{C}_{xv,0}$. Note that $\pi_n(\cdot)$ is also linear with respect to scalar multiplication. By (41), (42) and (45), for $\tau \geq \tau_0$:

$$\begin{aligned} \mathcal{C}_{x,(\tau,\lambda)} &= \pi_n(\mathcal{C}_{xv,(\tau,\lambda)}) = \pi_n(\widehat{\mathcal{C}}_{xv,(\tau,\lambda)}) \\ &\supseteq \pi_n((1 - c_0 a^{\tau-\nu})\mathcal{C}_{xv,max}) \\ &= (1 - c_0 a^{\tau-\nu})\mathcal{C}_{outer,\nu}. \end{aligned} \tag{46}$$

By Theorem 6.3 and (46), for any $\tau \geq \tau_0$:

$$(1 - c_0 a^{\tau-\nu})\mathcal{C}_{outer,\nu} \subseteq \mathcal{C}_{x,(\tau,\lambda)} \subseteq \mathcal{C}_{outer,\nu}. \tag{47}$$

Let $c_1 = \max_{x_1, x_2 \in \mathcal{C}_{outer,\nu}} \|x_1 - x_2\|_2$ be the diameter of $\mathcal{C}_{outer,\nu}$, which is finite since S_{xu} is bounded. Then, by (47), the Hausdorff distance between $\mathcal{C}_{x,(\tau,\lambda)}$ and $\mathcal{C}_{outer,\nu}$ satisfies:

$$d(\mathcal{C}_{x,(\tau,\lambda)}, \mathcal{C}_{outer,\nu}) \leq c a^\tau, \text{ for } c = c_0 c_1 a^{-\nu} \text{ and } \tau \geq \tau_0.$$

∎

Note that $\mathcal{C}_{outer,\nu}$ contains the union of the projections $\pi_n(\mathcal{C}_{xv})$ for all general implicit RCISs $\mathcal{C}_{xv}$ suggested by Theorem 3.1 (that is, the matrices H and P can be arbitrary, not necessarily the eventually periodic ones in Section III-C).

Hence, intuitively the set $\mathcal{C}_{outer,\nu}$ should be much larger than the projection of any specific implicit RCIS $\mathcal{C}_{xv}$ corresponding to an eventually periodic H and P in Section III-C. However, Theorem 6.6 shows that the proposed implicit RCIS can approximate $\mathcal{C}_{outer,\nu}$ arbitrarily well by just using the simple H and P matrices as in (18). Moreover, the approximation error decays exponentially fast as we increase the parameter τ in (18). This result implies that the eventually periodic input structure explored in Section III.B and III.C is rich enough, and not as conservative as what it may look at first sight.

Corollary 6.7: In the absence of disturbances, if the interior of S_{xu} contains a fixed point of Σ, then for any $\lambda > 0$, then $\mathcal{C}_{x,(\tau,\lambda)}$ converges to the Maximal CIS $\mathcal{C}_{max}$ in Hausdorff distance exponentially fast as τ increases.

The condition that the interior of S_{xu} (resp. $\overline{S}_{xu}$) contains a fixed point of Σ (resp. $\overline{\Sigma}$) in Corollary 6.7 (resp. Theorem 6.6) is critical to our method:

Example 1: Let Σ be $x_1^+ = x_2$, $x_2^+ = u$ and the safe set $S_{xu} = \{(x,u) | -1 \leq x_1,\ 1.5x_2 \leq x_1 \leq 2x_2, u \in [-1,1]\}$. The only fixed point of Σ in S_{xu} is the origin in $\mathbb{R}^3$, which is also a vertex of S_{xu}. It is easy to check that $\mathcal{C}_{max} = \pi_n(S_{xu})$, but the largest CIS $\pi_n(C_{xv})$ computed by our method is equal to the singleton set $\{0\}$.

If we expand S_{xu} slightly so that its interior contains the origin, there immediately exist H and P such that $\pi_n(\mathcal{C}_{xv})$ approximates $\mathcal{C}_{max}$ arbitrarily well, as expected by Corollary 6.7. Conversely, if we slightly shrink S_{xu} so that it does not contain any fixed point, then $\mathcal{C}_{max}$ is empty [9, Theorem12].

Remark 7: Under the assumption that $0 \in W$, let $\mathbb{S}_{xu}$ be the set of all the polytopic safe sets S_{xu} that have a nonempty $\mathcal{C}_{outer,\nu}$. Moreover, let $\partial\mathbb{S}_{xu}$ be the set of all safe sets $S_{xu} \in \mathbb{S}_{xu}$, whose corresponding nominal safe set $\overline{S}_{xu}$ does not contain a fixed point of $\overline{\Sigma}$ in the interior. It can be shown that $\partial\mathbb{S}_{xu}$ must be contained by the boundary of $\mathbb{S}_{xu}$ in the topology induced by Hausdorff distance. Consequently, for any safe set in the interior of $\mathbb{S}_{xu}$, there exists H and P such that $\pi_n(\mathcal{C}_{xv})$ approximates $\mathcal{C}_{outer,\nu}$ arbitrarily well.

VII. Case Studies

A MATLAB implementation of the proposed method, along with instructions to replicate our case studies, can be found at `https://github.com/tzanis-anevlavis/cis2m`. A C++ library is currently under development as well. Hence, we direct the interested reader to the above repository for the latest performance metrics.

A. Quadrotor obstacle avoidance using explicit RCIS

We begin by tackling the supervision problem, defined in Section V-B, for the task of quadrotor obstacle avoidance. That is, we filter nominal inputs to the quadrotor to ensure collision-free trajectories. The dynamics of the quadrotor can be modeled as a non-linear system with 12 states [26]. Nonetheless, this system is differentially flat, which implies that the states and inputs can be rewritten as a function of the so-called flat outputs and a finite number of their derivatives [36]. Exploiting this property, we obtain an equivalent linear system that expresses the motion of a quadrotor. Moreover,

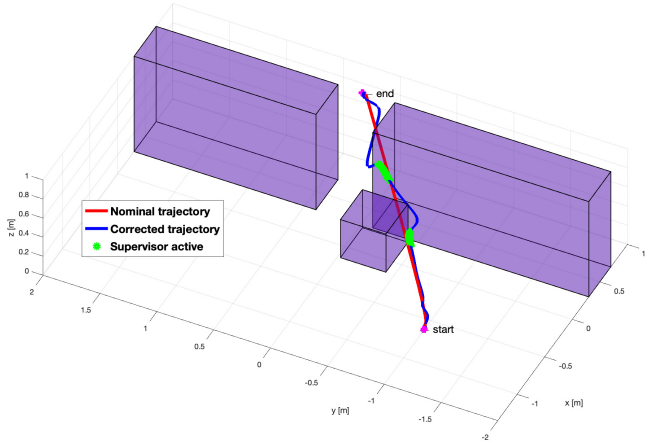


Fig. 3: Quadrotor operational region. Obstacles in purple transparent boxes. Nominal trajectory (red), corrected trajectory (blue), supervision active (green).

the original state and input constraints can be overconstrained by polytopes in the flat output space [29]. Then, the motion of a quadrotor can be described by:

$$x^+ = Ax + Bu + Ew,$$

with $A = \text{blkdiag}(A_1, A_2, A_3)$, $B = \text{blkdiag}(B_1, B_2, B_3)$, and:

$$A_i = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2!} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix},\ B_i = \begin{bmatrix} \frac{T_s^3}{3!} \\ \frac{T_s^2}{2!} \\ T_s \end{bmatrix}.$$

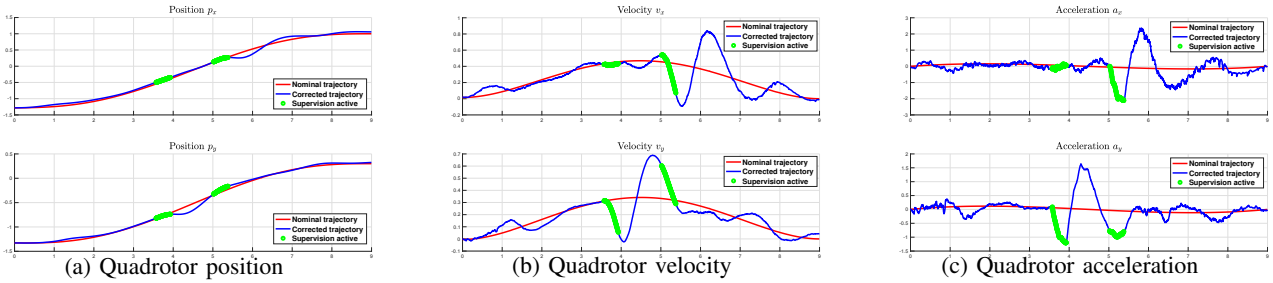
The state $x \in \mathbb{R}^9$ contains the 3-dimensional position, velocity, and acceleration, while the input $u \in \mathbb{R}^3$ is the 3-dimensional jerk. The matrix E and disturbance w are selected appropriately to account for various errors during the experiment.

The operating space for the quadrotor is a hyper-box with obstacles in $\mathbb{R}^3$, see Fig. 3. The safe set is described as the obstacle-free space, a union of overlapping hyper-boxes in $\mathbb{R}^3$, along with box constraints on the velocity and the acceleration:

$$S = \bigcup_{j=1}^{N} \left[\underline{p}_j, \overline{p}_j\right] \times [\underline{v}, \overline{v}] \times [\underline{a}, \overline{a}],$$

where $[\underline{p}_j, \overline{p}_j] \subset \mathbb{R}^3$, for $j = 1, \ldots, N$, is a hyper-box in the obstacle-free space, $[\underline{v}, \overline{v}] \subset \mathbb{R}^3$ and $[\underline{a}, \overline{a}] \subset \mathbb{R}^3$ denote the velocity and acceleration constraints respectively. The safe set is a union of polytopes, while our framework is designed for convex polytopes. Since we already know the obstacle layout, we compute an explicit RCIS for each polytope in the safe set. As these polytopes overlap we expect, and it is actually the case in our experiments, that the RCISs do so as well. This allows, when performing supervision, to select the input that keeps the quadrotor into the RCIS of our choice when in the intersection of overlaping RCISs and, hence, navigate safely.

Our goal is to ensure collision-free trajectory tracking. In Fig. 3, the nominal trajectory (red line) moves the quadrotor from a start point to an end point through the obstacles. As we can appreciate, the supervised trajectory (blue curve) takes the



(a) Quadrotor position (b) Quadrotor velocity (c) Quadrotor acceleration

Fig. 4: Quadrotor trajectory in x-y plane: nominal trajectory (red), corrected trajectory (blue), supervision active (green).

quadrotor around the obstacles and, safely, to the end point. When the supervision is active, the quadrotor performs more aggressive maneuvers to avoid the obstacle as shown in Fig. 4b and Fig. 4c, where we omit the z-axis as in this experiment the quadrotor maintains a relatively constant altitude. A video of the experiment is found at `https://tinyurl.com/drone-supervision-cis`. For visualizing the trajectory and the obstacles in the video, we used the Augmented Reality Edge Networking Architecture (ARENA) [10].

In this experiment we utilized the explicit RCIS $\mathcal{C}_{x,(\tau,\lambda)} = \pi_n\left(\mathcal{C}_{xv,(\tau,\lambda)}\right)$ with $(\tau,\lambda) = (0,6)$ and the one-step projection was done in just several seconds for this specific system. Our hardware platform is the open-source Crazyflie 2.0 quadrotor. The operating space for the position is $[-2,2]\times[-2,2]\times[0,1]$ (in m) and the obstacles are shown in Fig. 3. The velocity, acceleration, and jerk constraints are $[-1.0,1.0]$ (in m/s), $[-2.83,2.83]$ (in m/s^2), and $[-59.3,59.3]$ (in m/s^3) respectively. The sampling time is $T_s = 0.18s$. For the state estimation we use a Kalman filter, where the measurements are the quadrotor's position and attitude as obtained by the motion capture system OptiTrack. The nominal controller is a feedback controller stabilizing the error dynamics between the current state and a tracking point in the nominal trajectory. The optimization problems were solved by GUROBI [16].

B. Safe online planning using implicit RCIS

Next, we solve the safe online planning problem, discussed in Section V-C, for ground robot navigation. The map is initially unknown and is built online based on LiDAR measurements. While navigating the robot needs to avoid the obstacles, indicated by the dark area in Fig. 5, and reach the target point. This case study is inspired by the robot navigation problem in [5].

The robot's motion, using forward Euler discretization, is:

$$x^+ = \begin{bmatrix} \mathbb{I} & \mathbb{I}T_s \\ 0 & \mathbb{I} \end{bmatrix} x + \begin{bmatrix} 0 \\ \mathbb{I}T_s \end{bmatrix} u,$$

where the state $x = (p_x, p_y, v_x, v_y) \in \mathbb{R}^4$ is the robot's position and velocity and the input $u = (u_1, u_2) \in \mathbb{R}^2$ is the acceleration. The safe set consists of two parts:

1) The time-invariant constraints $v_x, v_y \in [-\overline{v}, \overline{v}]$ and $u_1, u_2 \in [-\overline{u}, \overline{u}]$.

2) The time-varying constraint of (p_x, p_y) within the obstacle-free region, shown by the white nonconvex area in

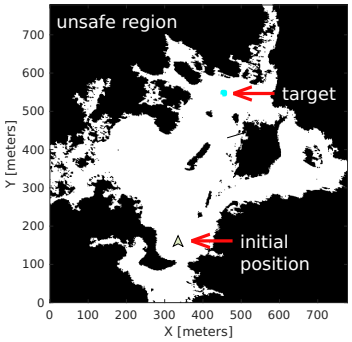


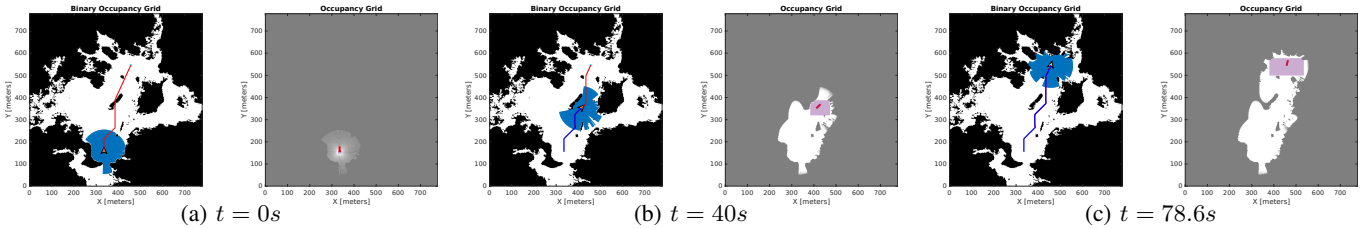
Fig. 5: Robot operational space: initial position (yellow arrowhead), target position (cyan), unsafe region (dark area).

Fig. 5. The obstacle-free region, denoted by $M(t) \subseteq R^2$, is determined by a LiDAR sensor using data up to time t. Combining the two constraints, the safe set at time t is:

$$S_{xu}(t) = \{(p_x, p_y, v_x, v_y, u_1, u_2) \mid (p_x, p_y) \in M(t), \\ v_x, v_y \in [-\overline{v}, \overline{v}],\ u_1, u_2 \in [-\overline{u}, \overline{u}]\}.$$

Since $M(t) \subseteq M(t+1)$, we have $S_{xu}(t) \subseteq S_{xu}(t+1)$, $t \geq 0$.

The overall control framework is shown in Fig. 2. Initially, the map is blank and the path planner generates a reference trajectory assuming no obstacles. At each time t, the map is updated based on the latest LiDAR measurements and the path planner checks if the reference trajectory collides with any obstacles in the updated map. If so, it generates a new, collision-free, reference path. Then, the nominal controller provides a candidate input $\tilde{u} = (\tilde{u}_1(t), \tilde{u}_2(t))$ tracking the reference path. When updating the reference trajectory, a transient period is needed for the robot to converge to the new reference. Moreover, the path planner cannot guarantee satisfaction of the input constraints. To resolve these issues, we add a supervisory control to the candidate inputs. Based on the updated obstacle-free region $M(t)$, we construct the safe set $S_{xu}(t)$ and compute an implicit CIS $\mathcal{C}_{xv,(\tau,\lambda)}(t)$ within $S_{xu}(t)$. To handle the nonconvexity of $S_{xu}(t)$, we first compute a convex composition of $S_{xu}(t)$. When constructing $\mathcal{C}_{xv,(\tau,\lambda)}(t)$, we let the reachable set at each time belong to one of the convex components in $S_{xu}(t)$, encoded by mixed-integer linear inequalities. For details see [22]. The convex decomposition of $S_{xu}(t)$ becomes more complex over time, which slows down the algorithm. To lighten the computational burden, we replace the full convex composition by the union of the 10 largest hyper-boxes in $S_{xu}(t)$ as the safe set. Given the



(a) $t = 0s$ (b) $t = 40s$ (c) $t = 78.6s$

Fig. 6: Simulation screenshots at times $t = 0s$, $40s$ and 78.6s. Left (a)-(b)-(c): reference path (red) and actual trajectory (blue); the disk of blue rays is the LiDAR measurements; the arrowhead indicates the position and moving direction of the robot. Right (a)-(b)-(c): obstacle-free region $M(t)$ (white) and unknown region (grey); purple boxes are the 10 largest boxes in $M(t)$ that contain the current robot position.

constructed implicit CIS $\mathcal{C}_{xv,(\tau,\lambda)}(t)$ at time t, we supervise the nominal control input $\tilde{u}(t)$ by solving $\mathcal{P}(t, t^*)$ as discussed in Section V-C. Note that $P(t, t^*)$ becomes a mixed-integer program as we introduced binary variables for the convex composition of the safe set and, therefore, in the implicit CIS.

In our simulations, we use a linear feedback controller as the nominal controller. The MATLAB Navigation Toolbox is used to simulate a LiDAR sensor with sensing range of 100 m, update the map, and generate the reference path based on the A* algorithm. The simulation parameters are $(\tau, \lambda) = (6, 4)$, $T_s = 0.1s$, $\overline{v} = 5m/s$, $\overline{u} = 5m/s^2$. The mixed-integer program $\mathcal{P}(t, t^*)$ is implemented via YALMIP [24] and solved by GUROBI [16]. The average computation time for constructing $\mathcal{C}_{xv,(\tau,\lambda)}(t)$ and solving $\mathcal{P}(t, t^*)$ at each time step is $2.87s$. The average computation time shows the efficiency of our method, considering the safe set is nonconvex and being updated at every time step.

The simulation results are shown in Fig. 6. The robot reaches the target region at $t = 78.6s$, and thanks to the supervisor, it satisfies the input and velocity constraints, while always staying within the time-varying safe region. As a comparison, when the supervisor is disabled, the velocity constraint is violated at time $t = 1.2s$. The full simulation video can be found at `https://youtu.be/mB9ir0R9bzM` .

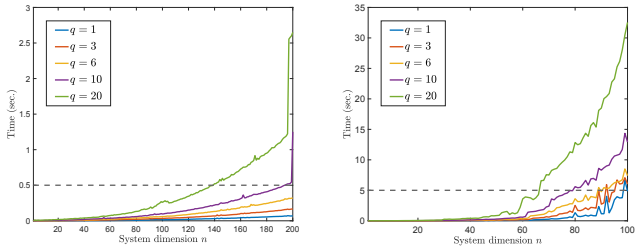
C. Scalability and quality

In this subsection we illustrate the scalability of the proposed method and compare with other methods in the literature. We consider a system of dimension n as in (1) that is already in Brunovsky normal form [8].

$$A_n = \begin{bmatrix} 0 & \mathbb{I} \\ 0 & 0 \end{bmatrix}, \; B_n = \begin{bmatrix} \mathbb{0} \\ 1 \end{bmatrix},$$

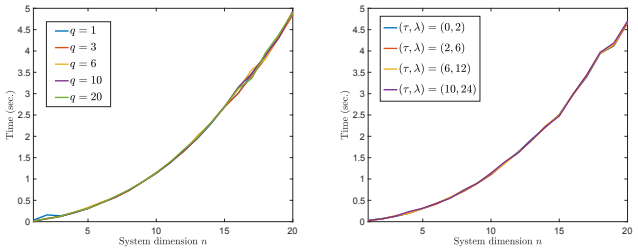
where $A_n \in \mathbb{R}^{n \times n}$ and $B_n \in \mathbb{R}^n$. This assumption does not affect empirical performance measurements as the transformation that brings a system in the above form is system-dependent and, thus, can be computed offline just once. To generalize the assessment of performance, we generate the safe set as a random polytope of dimension n and we average the results over multiple runs. Moreover, we constraint our input to $[-0.5, 0.5]$ and the disturbance to $[-0.1, 0.1]$.

1) Scalability of implicit invariant sets: We begin with the case of no disturbances. Fig. 7a and Fig. 7b show the times to compute the implicis CIS $\mathcal{C}_{xv,\mathrm{q}}$ for safe sets with $2n$ and n^2 constraints respectively. $\mathcal{C}_{xv,\mathrm{q}}$ can be computed in less than $0.5s$ for systems of size $n = 200$ when the safe set has $2n$



(a) (b)

Fig. 7: Absence of disturbances. Computation times for implicit CISs for different levels q of the full hierarchy, i.e., computing q Implicit CISs per level. Safe sets with (a) $2n$ constraints, $n \leq 200$, and (b) n^2 constraints, $n \leq 100$.



(a) (b)

Fig. 8: Presence of disturbances. Safe sets with $2n$ constraints, $n \leq 20$. Computation times for Implicit RCISs. (a) Different levels q of the hierarchy. (d) Individual implicit RCIS, $\mathcal{C}_{xv,(\tau,\lambda)}$, for different values of (τ, λ).

constraints, and in around $5s$ for $n = 100$ and safe sets with n^2 constraints, that is 10000 constraints in this example.

We now proceed to the case where system disturbances are present. In Fig. 8a and Fig. 8b, we observe that in the presence of disturbances computations are slower and, actually, are almost identical for different values of q. This is attributed to the presence of the Minkowsky difference in the closed-form expression (15) that dominates the runtime and depends on the nilpotency index of the system. Still, we are able to compute implicit RCISs in closed-form for systems with up to 20 states fairly efficiently in this experiment.

The above results suggest the efficiency and applicability of our approach to scenarios involving online computations, as shown already in Section VII-B. Moreover, in our experience, the numerical result of a projection operation, depending on

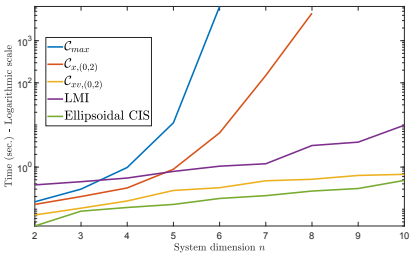


Fig. 9: Computation times for $\mathcal{C}_{xv,(0,2)}$, its projection $\mathcal{C}_{x,(0,2)}$, the LMI method in [34], the ellipsoidal CIS in [21], and $\mathcal{C}_{max}$. Logarithmic scale. Note: [21] is evaluated in the absence of disturbances as it considers only nominal systems. The other methods' performance without disturbance is similar or better.

the method used, can be sometimes unreliable. Contrary to this, our closed-form implicit representation does not suffer from such drawback.

Note that the runtimes in this section are derived using the MATLAB version of our approach. Even though it already shows the efficiency of our method, naturally, we expect the C++ library to further improve the presented runtimes.

2) Quality of the computed sets and comparison to other methods: We now compare our method with different methods in the literature, both in runtime and quality of the computed sets as measured by the percentage of their volume compared to the Maximal (R)CIS. Even though, we already provided a comprehensive analysis in terms of runtime for our method, we still present a few cases for the sake of comparison. We compare our approach to the Multi-Parametric Toolbox (MPT3) [17] that computes the Maximal (R)CIS, $\mathcal{C}_{max}$, the iterative approach in [34] that computes low-complexity (R)CISs, and the one in [21] that computes ellipsoidal CISs.

The runtimes of each method are reported in Fig. 9. The difficulty of computing $\mathcal{C}_{max}$ is apparent from the steep corresponding curve. The low-complexity methods in [34] and [21] are considerably faster, and [21] is slightly faster than even our implicit representation. However, our sets are superior in quality as we detail next.

First, in the absence of disturbances, the relative volume of the computed sets with respect to $\mathcal{C}_{max}$ is presented in Table I. Since for $n \geq 7$ MPT3 does not terminate after several hours and the computed set before termination is not invariant, we present the relative volumes only for $2 \leq n \leq 6$. Our method returns a very close approximation of $\mathcal{C}_{max}$ even with small values of (τ, λ) and computes substantially larger sets compared to the other techniques. This supports our theoretical result in Corollary 6.7. In other words, our implicit representation retains the best out of two worlds: computational efficiency and close approximations of $\mathcal{C}_{max}$.

In the presence of disturbances, the results are similar and are reported in Table II, where we omit [21] that only considers nominal systems. Theorem 6.6 proves that our method converges to its outer bound $\mathcal{C}_{outer,\nu}$. We can appreciate that empirically $\mathcal{C}_{outer,\nu}$ approximates very closely $\mathcal{C}_{max}$, even in the presence of disturbances, based on the size of the sets our method computes. However, the gap between $\mathcal{C}_{outer,\nu}$ and $\mathcal{C}_{max}$ depends on the size of the disturbance as shown next.

TABLE I: Absence of disturbances. Volume percentage with respect to the Maximal CIS. Algorithms: Our method for different implicit CISs $\mathcal{C}_{xv,(\tau,\lambda)}$, the LMI method in [34], and the ellipsoidal CIS in [21]. (S) denotes a singleton set.

	Our method		LMI method [34]	Ellipsoidal CIS method [21]
System dimension	$\mathcal{C}_{xv,(0,2)}$	$\mathcal{C}_{xv,(4,2)}$		
$n=2$	100	100	42.43	45.69
$n=3$	100	100	16.31	24.66
$n=4$	99.92	100	3.69	14.41
$n=5$	99.75	100	0.47	10.50
$n=6$	97.81	100	0 (S)	3.89

TABLE II: Presence of disturbances. Volume percentage with respect to the Maximal RCIS. Algorithms: Our method for different implicit RCISs $\mathcal{C}_{xv,(\tau,\lambda)}$ and the LMI method in [34]. (S) denotes a singleton set.

Volume (%)	Our method			LMI method [34]
System dimension	$\mathcal{C}_{xv,(0,2)}$	$\mathcal{C}_{xv,(2,2)}$	$\mathcal{C}_{xv,(4,2)}$	
$n=2$	100	100	100	31.99
$n=3$	98.24	99.67	99.96	16.35
$n=4$	99.02	99.42	99.88	4.36
$n=5$	98.75	99.74	99.81	3.64
$n=6$	91.17	96.07	97.91	0 (S)

TABLE III: Increasing the size of the disturbance set $W = [-\overline{w}, \overline{w}]$. Volume percentage of $\mathcal{C}_{x,(2,2)}$ with respect to the Maximal RCIS and volume percentage of $\overline{S}_{xu}$ with respect to S_{xu}. (NE) set is nonempty. (E) set is empty.

$\overline{w}$	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40
$\frac{vol\ \mathcal{C}_{x,(2,2)}}{vol\ \mathcal{C}_{max}}$	99.9	99.7	99.3	98.1	91.8	10.5	$\mathcal{C}_x$ empty	$\mathcal{C}_{max}$ empty
$\frac{vol\ \overline{S}_{xu}}{vol\ S_{xu}}$	63.9	38.2	20.5	9.2	2.6	0.1	$\overline{S}_{xu}$ empty	$\overline{S}_{xu}$ empty
$\overline{S}_{xu} \cap \overline{\Delta}_{xu}$	NE	NE	NE	NE	NE	NE	E	E

We illustrate how the size of the disturbance set affects our performance. We fix the safe set to be a random polytope in $\mathbb{R}^4$ and constrain the input to $[-0.5, 0.5]$. The disturbance set is $W = [-\overline{w}, \overline{w}]$ and we increase $\overline{w}$ as in Table III. Recall the nominal system $\overline{\Sigma}$ and the nominal safe set $\overline{S}_{xu} = S_{xu} - \overline{W}_{\infty}$, and let $\overline{\Delta}_{xu}$ be the set of fixed points of $\overline{\Sigma}$, which is in Brunovsky normal form. We can show that $\overline{\Delta}_{xu} = \{(x,u) \in \mathbb{R}^4 \times \mathbb{R} | x_1 = x_2 = x_3 = x_4 = u\}$. As Table III details, by increasing the size of W the our RCIS shrinks at a faster rate compared to C_{max}, until finally $\overline{S}_{xu}$ is empty and, hence, does not contain any fixed points from $\overline{\Delta}_{xu}$. This is when the set we compute becomes empty as well.

VIII. Related Literature

Recent works by the authors [1]–[3] develop methods constructing implicit RCISs in closed-form. These approaches consider different collections of periodic input sequences, which can be viewed as special instances of the parameterization proposed here. Furthermore, this work provides theoretical performance results, both for completeness and convergence, which extend to the previous methods as special cases. The concept of implicit RCISs is also explored in [12] for nominal systems and in [30], [32], [35] for systems with disturbances.

However, different from our method, these papers need to check a sufficient condition on set recurrence by LPs and do not provide completeness guarantees.

In addition to the aforementioned methods, a plethora of works have attempted to alleviate the poor scalability and the absence of termination guarantees of the standard method for computing the Maximal CIS of discrete-time systems introduced in [6]. The following list is not exhaustive.

One line of work [18], [28] focuses on outer and inner approximations of the Maximal CIS by solving either LPs or QPs. The resulting sets, however, are not always invariant. Comparing to these works, our method provides sets that are always guaranteed to be invariant and, given our closed-form expression, scales better with the system dimension.

Other methods compute exact ellipsoidal CISs efficiently and, thus, offer improved scalability, such as [21] which solves Semi-Definite Programs (SDP) for a class of hybrid systems. Nevertheless, the resulting ellipsoidal sets are generally small. This is backed by our comparison studies, which show that even though [21] computes very efficiently exact CISs, our implicit CIS offers similar computational performance, but substantially better quality in terms of approximating the Maximal CIS. In addition, for online control problems, like MPC and supervisory control, polytopes are preferred to ellipsoids, as they result in LPs or QPs, which are solved more efficiently compared to Quadratically Constrained Quadratic Programs (QCQP) that stem from ellipsoids.

In the presence of bounded disturbances, when the set of safe states are polytopes, [33] computes inner and outer approximations of the Maximal RCIS for linear systems. However, this iterative method suffers from the usual problem of performing an expensive projection operation in between iterations, which hinders its applicability in practice.

Ideas similar to ours, in the sense of using finite input sequences, were explored in the context of MPC [25]. Their goal is to establish asymptotic stability of a linear system, whereas we exploit finite input sequences that describe the proposed control behavior, leading to a closed-form expression for an implicit representation of controlled invariant sets. Other popular approaches first close the loop with a linear state-feedback control law, and then compute an invariant set of the closed-loop system. Under this umbrella, an idea close to ours is found in [20], where recurrent sets are computed in the context of MPC without disturbances. This can be understood as a special case of our eventually periodic approach.

In a similar spirit, i.e., by restricting to linear state-feedback control laws, the following works focus on reducing the computational cost and employ iterative procedures to compute low- or fixed-complexity RCISs and their associated feedback gains. In [34] low-complexity RCISs are found via SDPs under norm-bounded uncertainties. More recently, [14], [15] compute low- and fixed-complexity RCISs respectively for systems with rational parameter dependence. The complexity, i.e., the number of inequalities of the set, in [15] is twice the number of states, while [14] is more flexible as the complexity can be pre-decided. These methods assume the RCIS to be symmetric around the origin, whereas we make no assumptions on the RCIS. Arguably, pre-deciding the complexity is valuable for applications, such as MPC, but it can be very conservative. Increasing the set complexity to obtain larger sets hinders performance of said iterative methods. In comparison, we offer an alternative way to obtain larger sets, by increasing the transient and/or the period of the eventually periodic input parameterization. This bears minimal computation impact due to the derived closed-form expression.

The work of [27] computes larger controlled contractive sets of specified degree for nominal linear systems by solving Sum Of Squares (SOS) problems, but requires prior knowledge of a contractive set. Their scalability is also limited by the size of the SOS problems and so is its extension to handle polytopic uncertainty, which significantly increases the SOS problem size. Again, our method offers improved scalability along with the ability to increase the size of the computed set with minimal performance impact, as is backed by our experiments.

APPENDIX

A. Claims of Theorem 6.6

Proof of Claim 1: Since $\overline{S}_{xu}$ contains the origin, we have $\overline{W}_\infty \times \{0\} \subseteq S_{xu}$. Since $0 \in W$, it is easy to verify from (3) and (4) that $\overline{W}_k \subseteq \overline{W}_\infty$ for all $k \geq 1$. Thus, $\overline{W}_k \times \{0\} \subseteq S_{xu}$ for all $k \geq 1$. According to (5), if $x = 0$ and $u_t = 0$ for all $t \geq 0$, the reachable set $\left(\mathcal{R}_\Sigma\big(x, \{u_i\}_{i=0}^{t-1}\big), u_t\right) = \overline{W}_t \times \{0\} \subseteq S_{xu}$. Thus, $0 \in \mathcal{C}_{xv,0}$.

Proof of Claim 2: Recall from (31) that $\mathcal{C}_{outer,\nu}$ is:

$$\begin{aligned}\mathcal{C}_{outer,\nu} = \Big\{ & x \in \mathbb{R}^n \mid \exists \{u_i\}_{i=0}^{\nu-1} \in \mathbb{R}^{m\nu}, \\ & \big(\mathcal{R}_\Sigma\big(x, \{u_i\}_{i=0}^{t-1}\big), u_t\big) \subseteq S_{xu}, t = 0, \ldots, \nu-1, \\ & \mathcal{R}_\Sigma\big(x, \{u_i\}_{i=0}^{\nu-1}\big) \subseteq \overline{\mathcal{C}}_{\max} + \overline{W}_\infty \Big\}.\end{aligned}$$

Due to Lemma 6.2, $\mathcal{R}_\Sigma\big(x, \{u_i\}_{i=0}^{\nu-1}\big) \subseteq \overline{\mathcal{C}}_{\max} + \overline{W}_\infty$ if and only if $\sum_{i=0}^{\nu-1} A^{\nu-1-i} B u_i \in \overline{\mathcal{C}}_{max}$. Based on this observation, it is easy to verify that $\mathcal{C}_{outer,\nu} = \pi_n(\mathcal{C}_{xv,max})$ where $\mathcal{C}_{xv,max} = \mathcal{C}_{xv,0} \cap (\mathbb{R}^n \times \mathcal{U}(\overline{\mathcal{C}}_{max}))$.

Proof of Claim 3: We show that $\widehat{\mathcal{C}}_{xv,(\tau,\lambda)} = \pi_{n+\nu m}(\mathcal{C}_{xv,(\tau,\lambda)})$. Using the matrices H and P as in (17), the definition of $\mathcal{C}_{xv,(\tau,\lambda)}$, and Lemma 6.2, we write $\mathcal{C}_{xv,(\tau,\lambda)}$ as:

$$\begin{aligned}\mathcal{C}_{xv,(\tau,\lambda)} = \Big\{ & (x_0, u_{0:\tau+\lambda-1}) \mid \\ & \big(\mathcal{R}_\Sigma\big(x, \{u_i\}_{i=0}^{t-1}\big), u_t\big) \subseteq S_{xu}, t = 0, \cdots, \nu-1, \\ & (\mathcal{R}_{\overline{\Sigma}}(\sum_{i=1}^{v} A^{i-1} B u_{v-i}, \{u_{v+i}\}_{i=0}^{k-1}), u_{v+k}) \in \overline{S}_{xu}, \\ & k = 0, \cdots, \tau+\lambda-1 \Big\}.\end{aligned} \tag{48}$$

By (48) , the projection $\pi_{n+\nu m}(\mathcal{C}_{xv,(\tau,\lambda)})$ is:

$$\begin{aligned}\pi_{n+\nu m}(\mathcal{C}_{xv,(\tau,\lambda)}) = \Big\{ & (x_0, u_{0:\nu-1}) \mid \exists u_{\nu:\tau+\lambda-1}, \\ & \big(\mathcal{R}_\Sigma\big(x, \{u_i\}_{i=0}^{t-1}\big), u_t\big) \subseteq S_{xu}, t = 0, \cdots, \nu-1, \\ & (\mathcal{R}_{\overline{\Sigma}}(\sum_{i=1}^{\nu} A^{i-1} B u_{\nu-i}, \{u_{\nu+i}\}_{i=0}^{k-1}), u_{\nu+k}) \in \overline{S}_{xu}, \\ & k = 0, \cdots, \tau+\lambda-1 \Big\}.\end{aligned} \tag{49}$$

Again, using the matrices H and P as in (17), by the definition of $\overline{\mathcal{C}}_{x,(\tau-\nu,\lambda)}$, we have:

$$\begin{aligned}\overline{\mathcal{C}}_{x,(\tau-\nu,\lambda)} =&\Big\{x_0 \in \mathbb{R}^n \mid \exists u_{0:\tau-\nu+\lambda}, \\ &(\mathcal{R}_{\overline{\Sigma}}(x_0, \{u_i\}_{i=0}^{t-1}), u_t) \in \overline{S}_{xu}, \\ &t = 0, \cdots, \tau+\lambda-1\Big\}.\end{aligned} \tag{50}$$

Comparing the right hand sides of (49) and (50), we have:

$$\begin{aligned}&\pi_{n+\nu m}(\mathcal{C}_{xv,(\tau,\lambda)}) = \Big\{(x_0, u_{0:\nu-1}) \mid \\ &\left(\mathcal{R}_{\Sigma}\left(x, \{u_i\}_{i=0}^{t-1}\right), u_t\right) \subseteq S_{xu}, t = 0, \cdots, \nu-1, \\ &\sum_{i=1}^{\nu} A^{i-1} B u_{\nu-i} \in \overline{\mathcal{C}}_{x,(\tau-\nu,\lambda)}\Big\}.\end{aligned} \tag{51}$$

Note that $\mathcal{C}_{xv,0}$ and $\mathcal{U}(\overline{\mathcal{C}}_{x,(\tau-\nu,\lambda)})$ respectively impose the first and second constraints on $(x_0, u_{0:\nu-1})$ on the right hand side of (51). Thus, $\pi_{n+\nu m}(\mathcal{C}_{xv,(\tau,\lambda)})$ is equal to the intersection of $\mathcal{C}_{xv,0}$ and $\mathcal{U}(\overline{\mathcal{C}})_{x,(\tau-\nu,\lambda)}$. That is:

$$\widehat{\mathcal{C}}_{xv,(\tau,\lambda)} = \pi_{n+\nu m}(\mathcal{C}_{xv,(\tau,\lambda)}). \tag{52}$$

Since (52) implies (42), the third claim is proven.

Proof of Claim 4: We define the k-step null-controllable set $\mathcal{C}_k$ as the set of states of $\overline{\Sigma}$ that reach the origin at kth step under the state-input constraints S_{xu}:

$$\begin{aligned}\mathcal{C}_k = \Big\{&x \in \mathbb{R}^n \mid \exists u_{0:k-1} \in \mathbb{R}^{km}, \\ &(\mathcal{R}_{\overline{\Sigma}}(x, \{u_i\}_{i=0}^{t-1}), u_t) \in \overline{S}_{xu}, t = 0, \cdots, k-1, \\ &\mathcal{R}_{\overline{\Sigma}}(x, \{u_i\}_{i=0}^{k-1}) = 0\Big\}.\end{aligned} \tag{53}$$

Obviously, $\mathcal{C}_0 = \{0\}$. Since $A^\nu = 0$ and the fixed point $(0,0) \in \mathbb{R}^n \times \mathbb{R}^m$ is in the interior of $\overline{S}_{xu}$, there exists an $\epsilon > 0$ such that the ϵ-ball $B_\epsilon(0)$ at the origin satisfies that for $u_{0:\nu-1} = 0 \in \mathbb{R}^{\nu m}$ and for all $t \in [0, k-1]$:

$$\begin{aligned}&(\mathcal{R}_{\overline{\Sigma}}(B_\epsilon(0), \{u_i\}_{i=0}^{t-1}), u_t) = (A^t B_\epsilon(0), 0) \subseteq \overline{S}_{xu}, \\ &\mathcal{R}_{\overline{\Sigma}}(x, \{u_i\}_{i=0}^{\nu-1} = A^\nu B_\epsilon(0) = 0.\end{aligned} \tag{54}$$

By (54) and the definition of $\mathcal{C}_k$, $B_\epsilon(0)$ is contained by $\mathcal{C}_\nu$, and thus $\mathcal{C}_0 = \{0\}$ is contained in the interior of $\mathcal{C}_\nu$. Then, by Theorem 1 in [23], since $\mathcal{C}_0$ is contained in the interior of $\mathcal{C}_\nu$, there exists $\tau_2 \geq 0$, $c_2 \in [0,1]$ and $a \in [0,1)$ such that for all $k \geq \tau_2$, the Hausdorff distance $d(\mathcal{C}_k, \overline{\mathcal{C}}_{max})$ satisfies that:

$$d(\mathcal{C}_k, \overline{\mathcal{C}}_{max}) \leq c_2 a^k. \tag{55}$$

Furthermore, let $k = \tau$. For any $x \in \mathcal{C}_\tau$ and the corresponding $u_{0:\tau-1}$ satisfying the constraints on the right hand side of (53), it is easy to check that $(x, u_{0:\tau-1}, 0) \in \mathbb{R}^n \times \mathbb{R}^{(\tau+\lambda)m}$ is contained in $\overline{\mathcal{C}}_{xv,(\tau,\lambda)}$. Thus, we have for all $\tau \geq 0$:

$$\mathcal{C}_\tau \subseteq \overline{\mathcal{C}}_{x,(\tau,\lambda)} \subseteq \overline{\mathcal{C}}_{max}. \tag{56}$$

Thus, by (55) and (56), for any $\tau \geq \tau_2$, the Hausdorff distance $d(\overline{\mathcal{C}}_{x,(\tau,\lambda)}, \overline{\mathcal{C}}_{max})$ satisfies:

$$d(\overline{\mathcal{C}}_{x,(\tau,\lambda)}, \overline{\mathcal{C}}_{max}) \leq c_2 a^\tau. \tag{57}$$

From the properties of Hausdorff distance, (57) implies that:

$$\overline{\mathcal{C}}_{max} \subseteq \overline{\mathcal{C}}_{x,(\tau,\lambda)} + B_{c_2 a^\tau}(0), \tag{58}$$

where $B_{c_2 a^\tau}(0)$ is the ball at origin with radius $c_2 a^\tau$. Recall that $\mathcal{C}_\nu$ contains a ϵ-ball $B_\epsilon(0)$ for some $\epsilon > 0$. Since $\mathcal{C}_\nu \subseteq \overline{\mathcal{C}}_{max}$, we have $(c_2 a^\tau/\epsilon)\overline{\mathcal{C}}_{max} \supseteq B_{c_2 a^\tau}(0)$. Thus, by (58), we have for any $\tau \geq \tau_2$:

$$\overline{\mathcal{C}}_{max} \subseteq \overline{\mathcal{C}}_{x,(\tau,\lambda)} + \frac{c_2 a^\tau}{\epsilon}\overline{\mathcal{C}}_{max}. \tag{59}$$

Select a big enough τ_1 such that $\tau_1 \geq \tau_2$ and $c_2 a^{\tau_1} \leq \epsilon$. Then, by Lemma 6.2 and (59), we have for any $\tau \geq \tau_1$:

$$\overline{\mathcal{C}}_{x,(\tau,\lambda)} \supseteq (1 - c_0 a^\tau)\overline{\mathcal{C}}_{max},$$

where $c_0 = \frac{c_2}{\epsilon}$. Thus, the fourth claim is proven.

[1] 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

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

[3] 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

[4] P. J. Antsaklis and A. N. Michel, *Linear systems*. Springer, 1997, vol. 8.

[5] A. Bajcsy, S. Bansal, E. Bronstein, V. Tolani, and C. J. Tomlin, "An efficient reachability-based framework for provably safe autonomous navigation in unknown environments," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 1758–1765.

[6] 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.

[7] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[8] P. Brunovský, "A classification of linear controllable systems," *Kybernetika*, vol. 6, pp. 173–188, 1970.

[9] P. Caravani and E. De Santis, "Doubly invariant equilibria of linear discrete-time games," *Automatica*, vol. 38, no. 9, pp. 1531–1538, 2002.

[10] CONIX Research Center, "Augmented Reality Edge Networking Architecture – ARENA." [Online]. Available: https://conix.io/arena

[11] J. Duncan Glover and F. C. Schweppe, "Control of linear dynamic systems with set constrained disturbances," *Automatic Control, IEEE Transactions on*, vol. 16, pp. 411 – 423, 11 1971.

[12] M. Fiacchini and M. Alamir, "Computing control invariant sets is easy," *CoRR*, vol. abs/1708.04797, 2017.

[13] J. Grzybowski and R. Urbański, "Order cancellation law in the family of bounded convex sets," *Journal of Global Optimization*, pp. 1–12, 2019.

[14] 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, 2019.

[15] A. Gupta, H. Köroğlu, and P. Falcone, "Computation of low-complexity control-invariant sets for systems with uncertain parameter dependence," *Automatica*, vol. 101, pp. 330–337, 2019.

[16] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2020. [Online]. Available: http://www.gurobi.com

[17] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. of the European Control Conference*, Zürich, Switzerland, July 17–19 2013, pp. 502–510, http://control.ee.ethz.ch/~mpt.

[18] M. Korda, D. Henrion, and C. N. Jones, "Convex computation of the maximum controlled invariant set for polynomial control systems," *SIAM Journal on Control and Optimization*, vol. 52, no. 5, pp. 2944–2969, 2014. [Online]. Available: https://doi.org/10.1137/130914565

[19] A. J. Laub, *Matrix Analysis For Scientists And Engineers*. USA: Society for Industrial and Applied Mathematics, 2004.

[20] M. Lazar and V. Spinu, "Finite-step terminal ingredients for stabilizing model predictive control," *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 9–15, 2015, 5th IFAC Conference on Nonlinear Model Predictive Control NMPC 2015.

[21] B. Legat, P. Tabuada, and R. M. Jungers, "Computing controlled invariant sets for hybrid systems with applications to model-predictive control," in *6th IFAC Conference on Analysis and Design of Hybrid Systems, ADHS 2018, Oxford, UK*, 2018, pp. 193–198.
[22] Z. Liu and N. Ozay, "Safe online planning in unknown nonconvex environments with implicit controlled invariant sets," *IFAC-PapersOnLine*, 2021.
[23] ——, "On the convergence of the backward reachable sets of robust controlled invariant sets for discrete-time linear systems," 2022, accepted to CDC 2022. [Online]. Available: https://arxiv.org/abs/2207.04726
[24] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*. IEEE, 2004, pp. 284–289.
[25] D. Mayne, M. Seron, and S. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.
[26] M. W. Mueller and R. D'Andrea, "A model predictive controller for quadrocopter state interception," in *2013 European Control Conference (ECC)*, July 2013, pp. 1383–1389.
[27] S. Munir, M. Hovd, and S. Olaru, "Low complexity constrained control using higher degree lyapunov functions," *Automatica*, vol. 98, pp. 215 – 222, 2018.
[28] A. Oustry, M. Tacchi, and D. Henrion, "Inner approximations of the maximal positively invariant set for polynomial dynamical systems," *IEEE Control Systems Letters*, vol. 3, no. 3, pp. 733–738, July 2019.
[29] L. Pannocchi, T. Anevlavis, and P. Tabuada, "Trust your supervisor: quadrotor obstacle avoidance using controlled invariant sets," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 9219–9224.
[30] S. V. Rakovic and M. Baric, "Parameterized robust control invariant sets for linear systems: Theoretical advances and computational remarks," *IEEE Transactions on Automatic Control*, vol. 55, no. 7, pp. 1599–1614, 2010.
[31] S. V. Rakovic, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne, "Invariant approximations of the minimal robust positively invariant set," *IEEE Transactions on automatic control*, vol. 50, no. 3, 2005.
[32] S. V. Raković, E. C. Kerrigan, D. Q. Mayne, and K. I. Kouramas, "Optimized robust control invariance for linear discrete-time systems: Theoretical foundations," *Automatica*, vol. 43, no. 5, pp. 831–841, 2007.
[33] 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.
[34] 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, 2015.
[35] 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.
[36] D. Zhou and M. Schwager, "Vector field following for quadrotors using differential flatness," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6567–6572, 2014.

Tzanis Anevlavis was born in Athens, Greece. He received the Diploma degree in electrical and computer engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 2015, and the Ph.D. degree in electrical and computer engineering from the University of California, Los Angeles (UCLA), Los Angeles, CA, USA, in 2022.

Currently, his main research interests lie within the fields of safety-critical control, motion and trajectory planning, and formal methods in control of cyber-physical systems.

Zexiang Liu (Graduate Student Member, IEEE) was born in Beijing, China. He received the B.S. degree in Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2016, and the M.S. degree in Electrical and Computer Engineering from University of Michigan, Ann Arbor, MI, USA, in 2018. He is currently pursuing the Ph.D. degree in Electrical and Computer Engineering at the University of Michigan, Ann Arbor, MI, USA.

His current research interests lie in formal synthesis and verification for safety-critical systems, safe autonomy and system identification.

Necmiye Ozay (Senior Member, IEEE) received the B.S. degree in electrical engineering from Bogazici University, Istanbul, Turkey, in 2004, the M.S. degree in electrical engineering from Pennsylvania State University, University Park, PA, USA, in 2006 and the Ph.D. degree in electrical engineering from Northeastern University, Boston, MA, USA, in 2010. She was a Postdoctoral Scholar with the California Institute of Technology, Pasadena, CA, USA, between 2010 and 2013.

She joined the University of Michigan, Ann Arbor, MI, USA, in 2013, where she is currently an Associate Professor of electrical engineering and computer science. She is also a Member of the Michigan Robotics Institute. Her research interests include hybrid dynamical systems, control, optimization and formal methods with applications in cyber-physical systems, system identification, verification and validation, autonomy, and dynamic data analysis.

She was the recipient of the 1938E Award and a Henry Russel Award from the University of Michigan for her contributions to teaching and research, and five young investigator awards, including NSF CAREER, and the 2021 Antonio Ruberti Young Researcher Prize from the IEEE Control Systems Society for her fundamental contributions to the control and identification of hybrid and cyber-physical systems. Her papers have received several awards.

Paulo Tabuada (Fellow, IEEE) was born in Lisbon, Portugal, one year after the Carnation Revolution. He received the "Licenciatura" degree in aerospace engineering from the Instituto Superior Tecnico, Lisbon, Portugal, in 1998, and the Ph.D. degree in electrical and computer engineering in 2002 from the Institute for Systems and Robotics, a private research institute associated with Instituto Superior Tecnico.

Between January 2002 and July 2003, he was a Postdoctoral Researcher with the University of Pennsylvania. After spending three years at the University of Notre Dame, as an Assistant Professor, he joined the Electrical and Computer Engineering Department, University of California, Los Angeles, Los Angeles, CA, USA, where he is currently Vijay K. Dhir Professor of engineering.

Dr. Tabuada's contributions to control and cyber-physical systems have been recognized by multiple awards including the NSF CAREER award in 2005, the Donald P. Eckman award in 2009, the George S. Axelby award in 2011, the Antonio Ruberti Prize in 2015, the grade of fellow awarded by IEEE in 2017 and by IFAC in 2019. He has been a Program Chair and General Chair for several conferences in the areas of control and of cyber-physical systems such as NecSys, HSCC, and ICCPS. He is currently the Chair of HSCC's steering committee and served on the editorial board of IEEE Embedded Systems Letters and IEEE Transactions on Automatic Control.