Abstract—In this article, 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 nonexhaustive overview is found in Section VIII.

An alternative approach is to construct an implicit representation for an RCIS. The specific implicit representation used in this article 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 article, we propose a general framework for computing (implicit) RCISs for discrete-time linear systems (DTLSs) 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 existing related literature is found at the end of this article.
Notation: Let $\mathbb{R}$ be the set of real numbers and $\mathbb{N}$ be the set of positive integers. For sets $P, Q \subseteq \mathbb{R}^d$, the Minkowski sum is $P + Q = \{x \in \mathbb{R}^d \mid x = p + q, p \in P, q \in Q\}$ and the Minkowski difference is $Q - P = \{x \in \mathbb{R}^d \mid x + P \subseteq Q\}$, 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 $\mathbb{R}^d$. We denote a block-diagonal matrix $M$ with blocks $M_1, \ldots, M_N$ by $M = \text{blockdiag}(M_1, \ldots, M_N)$. 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 $AP = \{Ax \mid x \in P\}$. Given a set $S \subseteq \mathbb{R}^n$, its projection onto the first $n$ coordinates is $\pi_n(S)$. For any $N \in \mathbb{N}$, let $[N] = \{1, 2, \ldots, N\}$. Let $I$ and $0$ be the identity and zero matrices of appropriate sizes, respectively, while $1$ 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 DTLS $\Sigma$ is a linear difference equation

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

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 $S \subseteq \mathbb{R}^n$ is a bounded set of the form:

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

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 $\Sigma$ and a safe set $S_{xu} \subseteq \mathbb{R}^n \times \mathbb{R}^m$, that is, the set defining the state-input constraints for $\Sigma$, a set $C \subseteq \pi_n(S_{xu})$ is an RCIS for $\Sigma$ within $S_{xu}$ if:

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

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

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

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

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

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

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

For any system $\Sigma$ 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 $\Sigma'$ by prefeedback $\Sigma$ with $u = Kx + u'$:

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

where $u' \in \mathbb{R}^m$ is the input of the system $\Sigma'$. The safe set for $\Sigma'$ is the polytope induced from the safe set $S_{xu}$ of $\Sigma$ 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 $C$ of $\Sigma$ within $S_{xu}$ is an RCIS of $\Sigma'$ 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 $C$ of $\Sigma$ within $S_{xu}$, and $(x, u)$ with $x \in C$ and $Ax + Bu + EW \subseteq C$. 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 $\Sigma'$ gives $(A + BK)x + Bu' + EW = Ax + BKx - BKx + Bu + EW = Ax + Bu + EW \subseteq C$. Hence, $C$ is also an RCIS for $\Sigma'$ 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 $\Sigma$ within $S_{xu}$ is exactly equivalent to the problem of finding an RCIS of $\Sigma'$ within $S'_{xu}$. That is, for any procedure that takes in $(\Sigma, S_{xu})$ and produces a RCIS $C$, there exists an equivalent procedure that takes in $(\Sigma', S'_{xu})$ and produces the same RCIS $C$, 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 article 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 $\Sigma$, a safe set $S_{xu} \subseteq \mathbb{R}^n \times \mathbb{R}^m$, and some integer $q \in \mathbb{N}$, a set $C_{xu} \subseteq \mathbb{R}^n \times \mathbb{R}^d$ is an implicit RCIS for $\Sigma$ if its projection $\pi_n(C_{xu})$ onto the first $n$ dimensions is an RCIS for $\Sigma$ 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 is a robustly controlled invariant.

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

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

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

$$W_t = \sum_{i=1}^{t} A^{i-1} EW.$$  \hspace{1cm} (3)

By nilpotency of $A$ we have:

$$W_\infty = \sum_{i=1}^{\infty} A^{i-1} EW = \sum_{i=1}^{\nu} A^{i-1} EW.$$  \hspace{1cm} (4)

In literature, $W_\infty$ is called the minimal RPIS of the system $x^+ = Ax + Ew$ [31].

The next operator is used throughout this article.

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

$$\mathcal{R}_\Sigma(X, \{u_i\}_{i=0}^{\nu-1}) = A^\nu X + \sum_{i=1}^{\nu} A^{i-1} Bu_{t-i} + W_t.$$  \hspace{1cm} (5)
Intuitively, $\mathcal{R}_\Sigma(X, \{u_i\}_{i=0}^{\lfloor t\rfloor})$ maps a set $X$ and an input sequence $\{u_i\}_{i=0}^{\lfloor t\rfloor}$ to the set of all states that can be reached from $X$ in $t$ steps when applying said input sequence. Conventionally, $\mathcal{R}_\Sigma(X, \{u_i\}_{i=0}^{b}) = X$ if $b < a$, and when $X$ is a singleton, i.e., $X = \{x\}$, we abuse notation to write $\mathcal{R}_\Sigma(x, \{u_i\}_{i=0}^{\lfloor t\rfloor}).$

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 $\Sigma$ that evolve as the output of a linear dynamical system, $\Sigma_C$, whose state is a sequence of $q$ inputs, $v$:

$$\Sigma_C: \quad v_{t+1} = P v_t \quad u_t = H v_t$$

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 $\Sigma$ can be expressed as:

$$u_t = H v_t = H P^i v_0$$

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

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

Given the safe set $S_{xu}$, we construct the companion safe set $S_{xv} = \{(x, v) \in \mathbb{R}^n \times \mathbb{R}^{mq} \mid (x, H v) \in S_{xu}\}$. 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., $(x_t, H v_t) \in S_{xu}$.

Theorem 3.1 (Generalized implicit RCIS): Let $\mathcal{C}_{xv}$ be an RPIS of the companion system $\Sigma_{xv}$, 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 $\Sigma$ within $S_{xu}$. In other words, $\mathcal{C}_{xv}$ is an implicit RCIS of $\Sigma$.

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 $w = H v$ and pick an arbitrary $w \in W$. By construction of $S_{xv}$, $(x, w) \in S_{xu}$. Since $\mathcal{C}_{xv}$ is an RPIS, we have that $(x^{+}, v^{+}) = (\mathcal{A} x + B u + E w, P v) \in \mathcal{C}_{xv}$, and thus, $x^{+} = \pi_n(\mathcal{C}_{xv})$. By Definition 3, $\pi_n(\mathcal{C}_{xv})$ is an RCIS of $\Sigma$ in $S_{xu}$.

In principle, Theorem 3.1 holds even if $\Sigma_C$ is nonlinear. However, the choice of a linear system $\Sigma_C$, as in (6), makes the computation of the maximal RPIS of $\Sigma_{xv}$ more efficient. In what follows, we study the conditions on $P$ and $H$ such that the maximal RPIS of $\Sigma_{xv}$ 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 $\Sigma_{xv}$ within $S_{xv}$, if and only if, the input sequence $\{u_i\}_{i=0}^{\lfloor t\rfloor}$ with each input, as in (7), satisfies:

$$\mathcal{R}_\Sigma(x, \{u_i\}_{i=0}^{\lfloor t\rfloor}, u_t) \subseteq S_{xu}, \quad t \geq 0$$

where $\mathcal{R}_\Sigma(x, \{u_i\}_{i=0}^{\lfloor t\rfloor}, u_t) \subseteq \mathbb{R}^n$, $u_t \in \mathbb{R}^m$, and the pair $(\mathcal{R}_\Sigma(x, \{u_i\}_{i=0}^{\lfloor t\rfloor}, u_t)) \subseteq \mathbb{R}^n \times \mathbb{R}^m$. By Theorem 3.1, the above constraints characterize the states and input sequences within an implicit RCIS of $\Sigma$, 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:

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

We call $\tau$ the transient and $\lambda$ the period.

Proposition 3.2 (Finite reachability constraints): Consider a DTLS $\Sigma$ 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 $\nu$. Consequently, given (5), the reachable set from a state $x$ for $t \geq \nu$ depends only on the past $\nu$ inputs. We abuse notation to write $\mathcal{R}_\Sigma(\{u_i\}_{i=0}^{\lfloor t\rfloor})$ and omit the state $x$ to denote dependency only on the inputs. Then, for $t \geq \nu + \tau$:

$$\mathcal{R}_\Sigma(\{u_i\}_{i=0}^{\lfloor t\rfloor}) = \bigoplus_{i=1}^{\nu} A^{i-1} B u_{t-i} + \mathcal{W}_\infty$$

$$= \bigoplus_{i=1}^{\nu} A^{i-1} B u_{t+\lambda-i} + \mathcal{W}_\infty = \mathcal{R}_\Sigma(\{u_i\}_{i=0}^{\lfloor t+\lambda-1\rfloor}).$$

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:

$$\mathcal{R}_\Sigma(x, \{u_i\}_{i=0}^{\lfloor t\rfloor}, u_t) \subseteq S_{xu}, \quad t = 0, \ldots, \nu - 1$$
\[ \left( R_{\Sigma} \left( \{ u_i \}_{i=0}^{\nu-1}, u_\nu \right) \right) \subseteq S_{xu}, \quad t = \nu, \ldots, \nu + \tau + \lambda - 1. \] (12)

The above suggests that \( R_{\Sigma} \left( x, \{ u_i \}_{i=0}^{\nu-1}, u_\nu \right) \subseteq S_{xu} \) 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 \( \Sigma \), which is essentially the maximal RPPS of the companion system \( \Sigma_{xu} \), 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:

\[ P^t = P^{t+\lambda}, \quad t \geq \tau \] (13)

i.e., \( P \) is an eventually periodic matrix with transient \( \tau \) and period \( \lambda \).

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

Proof: Let \( \nu \neq 0 \) be an eigenvector of \( P \) and \( \delta \) the corresponding eigenvalue, i.e., \( P^\nu \delta = \delta \nu \). Then, (13) for \( t \geq \tau \) yields:

\[ P^t = P^{t+\lambda} \implies P^\nu \delta = P^{t+\lambda} \nu = \delta \nu = \delta \nu \]

\[ \nu \neq 0 \implies \delta \nu = \delta \nu \implies \delta = \delta \lambda \implies \delta = \lambda \]  

that is, the eigenvalues \( \delta \) of \( P \) are only 0 or \( \lambda \)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:

\[ J = \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} \]

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 \( \lambda \)th roots of unity. Thus, \( J_1 \) is nilpotent. Then, when \( \tau \neq 0 \), equality (13) is equivalent to:

\[ P^t = P^{t+\lambda} \implies MJ^tM^{-1} = MJ^{t+\lambda}M^{-1}, \quad t \geq \tau. \]

Matrix \( J_1 \) vanishes in exactly \( \tau \) 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 \( \sigma \) and geometric multiplicity equal to 1, but no 0 eigenvalues of geometric multiplicity 1 and algebraic multiplicity greater than \( \tau \).

Moreover, when \( P \) is not nilpotent, i.e., \( P^\tau \neq 0 \), for \( t \geq \tau \):

\[ J^t = J^{t+\lambda}, \quad J^0_{t=0, t \geq \tau} \]

Thus, \( P \) has at least one eigenvalue that is a \( \lambda \)-th root of unity.

Corollary 3.4: The class of matrices \( R_{\Sigma} \) that satisfies (13) can be written, up to a similarity transformation, in the following form:

\[ P = \begin{bmatrix} N & Q \\ 0 & R \end{bmatrix} \] (14)

where \( N \) is a nilpotent matrix with nilpotency index \( \sigma \), \( R \) is a matrix whose eigenvalues are all \( \lambda \)-th roots of unity, i.e., \( R^\lambda = I \), 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 nontrivial 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 \( \Sigma \) and a safe set \( S_{xu} \) for which Assumption 1 holds. Select an eventually periodic matrix \( P \in \mathbb{R}^{m \times m} \), and a surjective projection matrix \( H \in \mathbb{R}^{m \times m} \). An implicit RCIS for \( \Sigma \) within \( S_{xu} \), denoted by \( C_{xv} \), is defined by the constraints:

\[ A_i^t x + \sum_{i=1}^{t} A_i^t B H P^{t-i} v, H P^t v \subseteq S_{xu} - \mathcal{W}_t \times \{0\} \quad t = 0, \ldots, \nu - 1 \]

\[ \sum_{i=1}^{\nu} A_i^t B H P^{t-i} v, H P^t v \subseteq S_{xu} - \mathcal{W}_\infty \times \{0\} \quad t = \nu, \ldots, \nu + \tau + \lambda - 1. \] (15)

That is, the set \( C_{xv} \subset \mathbb{R}^n \times \mathbb{R}^{mq} \):

\[ C_{xv} = \{ (x, v) \in \mathbb{R}^n \times \mathbb{R}^{mq} \mid (x, v) \text{ satisfies (15)} \} \] (16)

is computed in closed-form. Moreover, \( C_{xv} \) is the maximal RPPS of the companion dynamical system in (8).

Proof: By Proposition 3.2, the set \( C_{xv} \) defined by (15) in closed-form satisfies the constraints in (9), and thus, is the maximal RPPS of the companion system \( \Sigma_{xv} \in S_{xv} \). Then, by Theorem 3.1, \( C_{xv} \) is an implicit RCIS of \( \Sigma \) in \( S_{xv} \).

Theorem 3.5 provides an implicit RCIS \( C_{xv} \) 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 toward 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 nonperiodic and memoryless controller \( K : \mathbb{R}^n \to \mathbb{R}^m \) for

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
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 $P \in \mathbb{R}^{m_q \times m_q}$ and a projection matrix $H \in \mathbb{R}^{m \times m_q}$, one computes an explicit RCIS $C_x = \pi_n(C_{xv})$ with a single projection step.

The size of the lifted space leads to a tradeoff: on the one hand, it can result to larger RCISs, as we detail in the next section, but on the other hand, it requires more effort if the optional projection step is taken.

### IV. Hierarchy of Controlled Invariant Sets

Our main result, Theorem 3.5, provides a closed-form expression for an implicit RCIS $C_{xv}$ 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 nondecreasing sequence. Our goal is to provide a closed-form expression for the implicit RCISs corresponding to this hierarchy. Toward 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 $\Sigma_C$ in (6) forms a $(\tau, \lambda)$-lasso sequence with respect to the inputs $v$, if:

$$ P = P_{(\tau, \lambda)} = \text{blkdiag}(\bar{P}, \ldots, \bar{P}) \in \mathbb{R}^{m_q \times m_q} $$

$$ H = H_{(\tau, \lambda)} = \text{blkdiag}(\bar{H}, \ldots, \bar{H}) \in \mathbb{R}^{m \times m_q} $$

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

$$ \bar{P} = \begin{bmatrix} 0 & 1 \\ 0 & \ddots & 1 \\ \vdots & \ddots & \ddots & 1 \\ 0 & \ddots & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{n_q \times q} $$

$$ \bar{H} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{1 \times q}. $$

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

We utilize the $(\tau, \lambda)$-lasso sequence to formalize a hierarchy of RCISs with a single decision parameter $\lambda$.

**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:

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

The cardinality of $\Theta_q$ 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 $\Sigma$ and a safe set $S_{xv}$ for which Assumption 1 holds, and select an integer $q \in \mathbb{N}$. Given $q$, the set $C_{xv,q} \subset \mathbb{R}^n \times \mathbb{R}^{m_q}$:

$$ C_{xv,q} = \bigcup_{(\tau, \lambda) \in \Theta_q} C_{xv,(\tau, \lambda)} $$

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

$$ C_{x,q} = \pi_n(C_{xv,q}) = \bigcup_{(\tau, \lambda) \in \Theta_q} \pi_n(C_{xv,(\tau, \lambda)}) $$

$$ = \bigcup_{(\tau, \lambda) \in \Theta_q} C_{x,(\tau, \lambda)} $$

(21)

corresponding to the $q$-level of the hierarchy contains any RCIS lower in the hierarchy:

$$ C_{x,q} \supseteq C_{x,q'}, \text{ for any } q, q' \in \mathbb{N} \text{ with } q' < q. $$

**Proof:** First, the sets $C_{xv,q}$ and $C_{x,q}$ 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:

$$ C_{x,(\tau + 1, \lambda)} \supseteq C_{x,(\tau, \lambda)} $$

(23)

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

$$ C_{x,(q + 1)} = \bigcup_{(\tau, \lambda) \in \Theta_{q + 1}} C_{x,(\tau, \lambda)} = \left( \bigcup_{(\tau, \lambda) \in \Theta_{q + 1}} C_{x,(\tau + 1, \lambda)} \right) \bigcup C_{x,(0,q + 1)} $$

(24)

$$ \supseteq \left( \bigcup_{(\tau, \lambda) \in \Theta_q} C_{x,(\tau, \lambda)} \right) \bigcup C_{x,(0,q + 1)} = \bigcup_{(\tau, \lambda) \in \Theta_q} C_{x,(\tau, \lambda)} \bigcup C_{x,(0,q + 1)} $$

The above entails that $C_{x,(q + 1)} \supseteq C_{x,q}$. $\blacksquare$

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

$$ C_{xv,q} = \left\{ (x, v, \zeta) \mid q \sum_{i=1}^{q} \zeta_i = 1, G_i(x, v) \leq f_i + (1 - \zeta_i) M_1 \right\} $$

(24)

where $\zeta \in \{0, 1\}^q$, $G_i$ and $f_i$ describe each of the $q$ polytopes $C_{xv,(\tau, \lambda)}$ in (20), and $M \in \mathbb{R}_+$ is sufficiently large. The set $C_{xv,q}$ 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 conv($U_{(\tau, \lambda) \in \Theta_q} C_{xv,(\tau, \lambda)}$). 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 $C_{xv}(\tau, \lambda)$ 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 $(\tau, \lambda)$ as follows:

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

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

The above can directly 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], [2], [3]. However, the required projection from a higher dimensional space becomes the bottleneck of this approach.

One of the goals of this article 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 $\pi_u(C_{xu})$ at a given state $x$, i.e., $A(x)$, as given in Definition 4. Given only the implicit RCIS $C_{xu}$, we provide here three linear encodings of $A(x)$ or its nonempty subsets:

1) The first linear encoding of $A(x)$ is given by the polytope

$$U_1(x) = \left\{ (u, v_{1:N}) \in \mathbb{R}^{(1+Nq)m} \mid (x, u) \in S_{xu}, \right. \\
(Ax + Bu + Ew, v_i) \in C_{xv}, \forall i \in [N] \}$$

(25)

where $v_{1:N}$ denotes the vector $(v_1, v_2, \ldots, v_N)$. It follows that $\pi_u(U_1(x)) = A(x)$.

2) The second linear encoding is:

$$U_2(x) = \left\{ v \in \mathbb{R}^m \mid (x, v) \in C_{xv} \right\}$$

(26)

with $H$ and $P$, as in (6). Note that $U_2(x)$ is the slice of $C_{xv}$ at $x$ and is nonempty for $x \in \pi_u(C_{xv})$. Then, the linear transformation $HU_2(x)$ is a nonempty subset of $A(x)$.

3) Finally, define the polytope:

$$U_3(x) = \left\{ (u, v) \in \mathbb{R}^{(1+q)m} \mid (x, u) \in S_{xu}, \\
(Ax + Bu + Ew, v) \in C_{xv}, \forall i \in [N] \}$$

(27)

where $w_i \in \mathcal{V}$ with $\mathcal{V}$ being the vertices of $W$. It follows that $\pi_m(U_3(x)) \subseteq A(x)$.

That is, $U_4(x)$ is the most conservative encoding, while $U_1(x)$ is the least conservative one. However, $U_2(x)$ is of lower dimension, while $U_1$ 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, the goal is to synthesize a controller that meets a performance criterion while always satisfying a safety requirement. This gives rise to the following problem.

Problem 1 (Supervisory Control): Consider a system $\Sigma$, 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 $\Sigma$ 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 $C$, assume that the initial state of $\Sigma$ lies in $C$. The nominal controller provides an input $u$ to be executed by $\Sigma$. If $u \in A(x)$, then its execution is allowed. Else $\tilde{u}$ is corrected by selecting an input $u_{safe} = A(x)$. 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 $C_{xv}$ for $\Sigma$ within $S_{xu}$, as in Theorem 3.5. Then supervision of an input $u$ is performed by solving the following QP:

$$\min_{u, v} \left\| u - \tilde{u} \right\|_2$$

s.t. \( (x, u) \in S_{xu}, \)

$$Ax + Bu + Ew, v_i \in C_{xv}, \forall i \in [N]$$

(28)
Notice that the feasible domain of the QP in (28) is equal to the third linear encoding $U_3(x)$ of admissible inputs; similar QPs are easily formulated with the feasible domain being $U_1(x)$ or $U_2(x)$. By solving optimization problem (28) we compute the minimally intrusive safe input.

### 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 Fig. 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 $C_{xu}(t)$ for each $S_{xu}(t)$. Then, at each time step $t$, for any $t' \leq t$, we supervise the nominal input $\bar{u}(t)$ by solving the optimization problem:

$$\min_{u,v} \quad ||u - \bar{u}||_2^2$$

$$\mathcal{P}(t, t') : \quad \text{s.t.} \quad (x, u) \in S_{xu}(t) \cap \mathcal{C}_{xu}(t') \forall w \in W.$$

As $S_{xu}(t) \subseteq S_{xu}(t+1)$, $C_{xu}(t')$ 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. If $\mathcal{P}(t, t')$ is feasible, by definition of RCIS, $\mathcal{P}(t+1, t')$ is also feasible. Thus, if $\mathcal{P}(0, 0)$ is feasible, for all $t > 0$, there exists $t' \leq t$ such that $\mathcal{P}(t, t')$ is feasible. That is, the recursive feasibility of $\mathcal{P}(t, t')$ is guaranteed. In practice, to take advantage of the latest map, we always select $t'$ to be the latest time instant $t^*$ for which $\mathcal{P}(t, t^*)$ is feasible.

To summarize, at each time step, we first construct the implicit RCIS $C_{xu}(t)$ 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 $t \geq 0$, if $\mathcal{P}(0, 0)$ 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 $\Sigma$, a safe set $S_x$, and the maximal RCIS $\mathcal{C}_{\max} \subseteq S_x$. Define a hyper-box $B = [b_1, \bar{b}_1] \times \cdots \times [b_n, \bar{b}_n] = [b, \bar{b}] \subset \mathbb{R}^n$. We call a hyper-box $B$ safe if $B \subseteq \mathcal{C}_{\max}$.

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 $B$ entails the guarantee that the trajectory starting therein can remain in $S_x$ forever, but not necessarily within $B$. We now aim to address the following problem.

**Problem 2:** Find the largest safe hyper-box $B$ within $S_x$.

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

$$A^t [b, \bar{b}] + \sum_{i=1}^t A^{t-i}BHP^{t-i}v \subseteq S_x - \overline{W}_t, t = 0, \ldots, \nu - 1$$

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

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

$$\max \gamma (\bar{b} - b)$$

$$\text{s.t.} \quad (b, \bar{b}, v) \in \mathcal{C}_B$$

where $\gamma(y) = (\prod_{i=1}^n y_i)^{\frac{1}{2}}$ is the geometric mean function, which is used as a heuristic for the volume of the hyper-box. Function $\gamma$ 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, for $\tau = 0, \lambda = 1$, see also [1], and recurrent hyper-boxes, for $\tau = 0, \lambda > 0$, 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

---

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.
point \( x_c \) where the system is required to operate is safe. If both the above are modeled by hyper-boxes \((\bar{b}, \bar{b}, v)\), we can simply ask whether there exists a \( v \), such that \((\bar{b}, \bar{b}, v) \in C_p\). Similarly, more complicated questions can be formulated, e.g., to find the largest safe box around a configuration point.

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 computational method. Given (4), define the nominal DTLSS \( \Sigma \) and the nominal safe set \( \overline{S}_{\Sigma} \):

\[
\Sigma : x^+ = Ax + Bu
\]

\[
\overline{S}_{\Sigma} = S_{\Sigma} - W_{\infty} \times \{0\}
\]

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

\[
C_{\text{outer},\nu} = \bigg\{ x \in \mathbb{R}^n \mid \exists \{u_{i \nu^{-1}}\} \in \mathbb{R}^{mv},
\]

\[
(R_{\Sigma}(x, \{u_{i \nu^{-1}}\}), u_{\nu^{-1}}) \subseteq S_{\Sigma}, t = 0, \ldots, \nu - 1,
\]

\[
R_{\Sigma}(x, \{u_{i \nu^{-1}}\}) \subseteq \overline{C}_{\max} + W_{\infty}
\]

where \( \nu \) is the nilpotency index of \( A \).

Proposition 6.1: \( C_{\text{outer},\nu} \) is an RCIS of \( \Sigma \) within \( \overline{S}_{\Sigma} \).

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

Lemma 6.2: Let \( X, Y \subseteq \mathbb{R}^n \) 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 \( C_{\text{outer},\nu} \) is an RCIS, we show that for any \( x_0 \in C_{\text{outer},\nu} \), there exists \( u \) such that \((x_0, u) \in S_{\Sigma} \) and for all \( w \in W \), \( Ax_0 + Bu + Ew \in C_{\text{outer},\nu} \). By definition of \( C_{\text{outer,\nu}} \), there exists a sequence \( \{u_{i \nu^{-1}}\} \) that, along with \( x_0 \), satisfies the conditions in (31). We aim to show that \( u_0 \) in \( \{u_{i \nu^{-1}}\} \) is a feasible choice for \( u \). Given (31), the reachable set from \( x_0 \) at time \( \nu \) is:

\[
R_{\Sigma}(x_0, \{u_{i \nu^{-1}}\}) = \sum_{i=0}^{\nu-1} A^{\nu-1-i} Bu_i + W_{\infty} \subseteq \overline{C}_{\max} + W_{\infty}
\]

with \( W_{\infty} \) and \( \overline{C}_{\max} \) convex and \( W_{\infty} \) bounded. By Lemma 6.2 we have that \( \sum_{i=0}^{\nu-1} A^{\nu-1-i} Bu_i \in \overline{C}_{\max} \). Since \( \overline{C}_{\max} \) is controlled invariant within \( \overline{S}_{\Sigma} \), there exists \( u_\nu \)

such that:

\[
\sum_{i=0}^{\nu-1} A^{\nu-1-i} Bu_i, u_\nu \in \overline{S}_{\max}
\]

\[
A \left( \sum_{i=0}^{\nu-1} A^{\nu-1-i} Bu_i \right) + Bu_\nu = \sum_{i=1}^{\nu} A^{\nu-1-i} Bu_i \in \overline{C}_{\max}.
\]

(32)

Consider any \( w \in W \). Define \( x_1 = Ax_0 + Bu_0 + Ew \) and:

\[
R_{\Sigma}(x_1, \{u_{i \nu^{-1}}\}) = \sum_{i=1}^{\nu} A^{\nu-1-i} Bu_i + W_{\infty} \subseteq \overline{C}_{\max} + W_{\infty}.
\]

(33)

From (32) we have:

\[
R_{\Sigma}(x_1, \{u_{i \nu^{-1}}\}, u_\nu) \subseteq S_{\Sigma}.
\]

(34)

Finally, note that for \( t = 0, \ldots, \nu - 2 \), we have:

\[
R_{\Sigma}(x_1, \{u_{i \nu^{-1}}\}, u_{i+1}) \subseteq \overline{C}_{\max} + W_{\infty}.
\]

(35)

From (33)–(35) we verify that \( x_1 \in C_{\text{outer,\nu}} \). Thus, \( C_{\text{outer,\nu}} \) is an RCIS.

The following theorem shows that \( C_{\text{outer,\nu}} \) is an outer bound of the projection of the proposed implicit RCIS.

Theorem 6.3 (Outer bound on \( \pi_{\nu}(C_{\nu}) \): For a companion system \( \Sigma_{\nu} \), as in (8), with arbitrary matrices \( P \) and \( H \), let \( C_{\nu} \) be an RPIS of \( \Sigma_{\nu} \) within the companion safe set \( S_{\Sigma} \). The RCIS \( \pi_{\nu}(C_{\nu}) \) is a subset of \( C_{\text{outer,\nu}} \), that is \( \pi_{\nu}(C_{\nu}) \subseteq C_{\text{outer,\nu}} \).

Proof: Let \( x \in \pi_{\nu}(C_{\nu}) \). We show that \( x \in C_{\text{outer,\nu}} \). By definition of \( C_{\nu} \), there exists a vector \( v \) such that:

\[
R_{\Sigma}(x, \{H P^i v\}_{i=0}^{\nu-1}), H P^i v) \subseteq S_{\Sigma}.
\]

for all \( t \geq 0 \). (36)

Define \( u_t = H P^t v \). We want to verify that \( x \) and \( \{u_{i \nu^{-1}}\} \) satisfy the two conditions in the definition of (31). The first condition is immediately satisfied by (36). It is left to show that:

\[
R_{\Sigma}(x, \{u_{i \nu^{-1}}\}) \subseteq \overline{C}_{\max} + W_{\infty},
\]

which is equivalent to:

\[
\sum_{i=0}^{\nu-1} A^{\nu-1-i} Bu_i + W_{\infty} \subseteq \overline{C}_{\max} + W_{\infty}.
\]

By Lemma 6.2, it is equivalent to prove that:

\[
\overline{\tau} \equiv \sum_{i=0}^{\nu-1} A^{\nu-1-i} Bu_i \in \overline{C}_{\max}.
\]

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

\[
\left( \sum_{i=0}^{\nu-1} A^{\nu-1-i} Bu_i + W_{\infty}, u_{\nu+t} \right) \subseteq S_{\Sigma}.
\]

(37)

According to (37), the control sequence \( \{u_{t \nu^{-1}}\} \) guarantees that the trajectory of \( \Sigma \) starting at \( \overline{\tau} \) stays within \( \overline{S}_{\Sigma} \) for all \( t \geq 0 \). Thus, \( \overline{\tau} \) must belong to the maximal CIS of \( \Sigma \) in \( \overline{S}_{\Sigma} \).
maximal RCIS. Given Theorem 6.3 we have:
\[ \pi_n(C_{x,v}) \subseteq C_{\text{outer},v} \subseteq C_{\text{max}}. \] (38)
Thus, the projection of our implicit RCIS can coincide with the maximal RCIS, for appropriately selected matrices \( P \) and \( H \), only if \( C_{\text{outer},v} = C_{\text{max}} \) 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 \( C_{\text{outer},v} \) is nonempty, if and only if, there exist matrices \( P \) and \( H \) such that the corresponding implicit RCIS \( C_{x,v} \) is nonempty. Specifically, \( C_{\text{outer},v} \neq \emptyset \), if and only if, \( C_{x,v,(0,1)} \neq \emptyset \), is that \( P \) and \( H \) as are in (17) with \( (\tau, \lambda) = (0, 1) \).

**Proof:** We want to show that \( C_{\text{outer},v} \) is nonempty if and only if \( C_{x,v,(0,1)} \) is nonempty, where \( C_{x,v,(0,1)} \) is defined in (20) with respect to system \( \Sigma \) and safe set \( S_{x,u} \).

Since \( \pi_n(C_{x,v,(0,1)}) \subseteq C_{\text{outer},v} \), immediately nonemptiness of \( C_{x,v,(0,1)} \) implies nonemptiness of \( C_{\text{outer},v} \).

For the converse, suppose that \( C_{\text{outer},v} \) is nonempty. Then \( C_{x,v} \) is nonempty. By [9, Th. 12], we know that \( C_{x,v} \) is nonempty, if and only if, there exists a fixed point \( x \in C_{x,v} \) along with a \( u \) such that \( (x, u) \in S_{x,u} \) and \( Ax + Bu = x \). Also, note that \( AW_\infty + EW = W_\infty \). Thus, we have:
\[ (x + W_\infty, u) \subseteq S_{x,u} \]
\[ A(x + W_\infty) + Bu + EW = x + W_\infty. \] (39)
According to (39), for any \( y \in x + W_\infty \), we have \( (y, u) \in S_{x,u} \) and \( Ay + Bu + EW \subseteq x + W_\infty \), which implies that \( x + W_\infty \) is an RCIS of \( \Sigma \) within \( S_{x,u} \). By the definition of \( C_{x,v,(0,1)} \), it is easy to check that \( (x + W_\infty, u) \subseteq C_{x,v,(0,1)} \). Thus, \( C_{x,v,(0,1)} \) is nonempty.

**Corollary 6.5 (Completeness in absence of disturbances):** In the absence of disturbances, \( C_{\text{outer},v} = C_{\text{max}} \), and thus, there exist \( P \) and \( H \) such that \( C_{x,v} \) is nonempty, if and only if, \( C_{x,v} \) is nonempty. That is, the proposed method is complete.

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

**Theorem 6.6 (Convergence to \( C_{\text{outer},v} \)):** Assume that the disturbance set \( W \) contains 0, and the interior of \( S_{x,u} \) contains a fixed point \( (x, u) \) of \( \Sigma \). There exist matrices \( H \) and \( P \) such that \( \pi_n(C_{x,v}) \) approaches \( C_{\text{outer},v} \). Specifically, if \( H \) and \( P \) are as in (17), by increasing \( \tau \) in (17), \( \pi_n(C_{x,v}) \) converges to \( C_{\text{outer},v} \) in the Hausdorff distance exponentially fast.

**Proof:** Without loss of generality, assume that the fixed point \( (x, u) \) of \( \Sigma \) is in the interior of \( S_{x,u} \) is the origin of the state-input space. We define a set operator \( U(\Sigma) \) that maps a subset \( \Sigma \) of \( R^n \) to a subset of \( R^{m \nu} \):
\[ U(\Sigma) = \left\{ u_{0,\nu-1} \in R^{m \nu} \mid \sum_{i=1}^{\nu} A^{i-1} B u_{i-1} \in \Sigma \right\} \] (40)
where \( u_{0,\nu-1} \) denotes the vector \( (u_0, u_1, \ldots, u_{\nu-1}) \) in \( R^{m \nu} \).

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

**Claim 1:** The polytope \( C_{x,v,0} \) contains the origin, where:
\[ C_{x,v,0} = \{ (x, u_{0,\nu-1}) \in R^{m \nu} \mid (R_{\Sigma}(x, \{ u_t \}_{t=0}^{\nu-1}), u_t) \subseteq S_{x,u}, t = 0, \ldots, \nu - 1 \}. \]

**Claim 2:** For the set \( C_{\text{outer},v} \) in (31) it holds that:
\[ c_{\text{outer},v} = \pi_n(C_{x,v,0}) \] (41)
where \( C_{x,v,0} = \bigcap (R^{m \nu} \times U(\Sigma_{C_{x,v}})) \).

**Claim 3:** Let \( C_{x,v,(\tau,\lambda)} \) be the implicit CIS of the nominal system \( \Sigma \) within \( S_{x,u} \) with \( H \) and \( P \), as in (17) and let \( C_{x,v,(\tau,\lambda)} = \pi_n(C_{x,v,(\tau,\lambda)}) \). The implicit RCIS \( C_{x,v,(\tau,\lambda)} \) of \( \Sigma \) within \( S_{x,u} \) with \( H \) and \( P \), as in (17) satisfies:
\[ \pi_n(C_{x,v,(\tau,\lambda)}) = \pi_n(C_{x,v,(\tau,\lambda)}), \] for any \( \tau \geq \nu \) (42)
where \( C_{x,v,(\tau,\lambda)} = \pi_n(C_{x,v,(\tau,\lambda)}), \) the eventually periodic ones in Section III-C. Hence, intuitively the set \( C_{\text{outer},v} \) should be much larger than the projection of any specific implicit RCIS \( C_{x,v} \) corresponding to an eventually periodic \( H \) and \( P \) in Section III-C. However, Theorem 6.6 shows that the proposed implicit RCIS can approximate \( C_{\text{outer},v} \).
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 \( \tau \) 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 \( \Sigma \), then for any \( \lambda > 0 \), then \( C_{\Sigma}(\tau,\lambda) \) converges to the maximal CIS \( C_{\max} \) in the Hausdorff distance exponentially fast as \( \tau \) increases.

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

Example 1: Let \( \Sigma \) be \( x_1^2 = x_2, x_2^2 = u \) 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 \( \Sigma \) in \( S_{xu} \) is the origin in \( \mathbb{R}^3 \), which is also a vertex of \( S_{xu} \). It is easy to check that \( C_{\max} = \pi_n(S_{xu}) \), but the largest CIS \( \pi_n(C_{\text{fav}}) \) 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 \( \pi_n(C_{\text{fav}}) \) approximates \( C_{\max} \) 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 \( C_{\max} \) is empty [9, Th. 12].

Remark 7: Under the assumption that \( 0 \in \mathcal{W} \), let \( \overline{S}_{xu} \) be the set of all the polytopic safe sets \( S_{xu} \) that have a nonempty \( \overline{\Sigma}_{xu} \). Moreover, let \( \partial S_{xu} \) be the set of all safe sets \( S_{xu} \subseteq \overline{S}_{xu} \) 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 \( \partial S_{xu} \) must be contained by the boundary of \( S_{xu} \) in the topology induced by the Hausdorff distance. Consequently, for any safe set in the interior of \( S_{xu} \), there exists \( H \) and \( P \) such that \( \pi_n(C_{\text{fav}}) \) approximates \( C_{\text{fav},\nu} \) 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. See 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, 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:

\[
A_i = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}, \quad B_i = \begin{bmatrix} \frac{T_s^3}{6} \\ \frac{T_s^2}{2} \\ T_s \end{bmatrix}.
\]

The state \( x \in \mathbb{R}^9 \) contains the 3-D position, velocity, and acceleration, while the input \( u \in \mathbb{R}^3 \) is the 3-D 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:

\[
S = \bigcup_{j=1}^{N} [p_j^x, p_j^y] \times [u_j, \pi] \times [a_j, \pi]
\]

where \( [p_j^x, p_j^y] \subset \mathbb{R}^3 \), for \( j = 1, \ldots, N \), is a hyper-box in the obstacle-free space, \( [u, \pi] \subset \mathbb{R}^3 \) and \( [a, \pi] \subset \mathbb{R}^3 \) 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 overlapping 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 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. 4(b) and (c), 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-
The robot’s motion, using forward Euler discretization, is:

\[
x^+ = \begin{bmatrix} 1 & T_s \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ T_s \end{bmatrix} u
\]

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 as follows:

1) The time-invariant constraints \( v_x, v_y \in [-\tau, \tau] \) and \( u_1, u_2 \in [-\bar{u}, \bar{u}] \).

2) The time-varying constraint of \( (p_x, p_y) \) within the obstacle-free region, shown by the white nonconvex area in Fig. 5. The obstacle-free region, denoted by \( M(t) \subseteq \mathbb{R}^2 \), is determined by a LiDAR sensor using data up to time \( t \).

Combining the two constraints, the safe set at time \( t \) is:

\[
S_{zu}(t) = \{(p_x, p_y, v_x, v_y, u_1, u_2) \mid (p_x, p_y) \in M(t), \quad v_x, v_y \in [-\tau, \tau], \quad u_1, u_2 \in [-\bar{u}, \bar{u}] \}.
\]

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

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 \( S_{zu}(t) \) and compute an implicit CIS \( C_{zu, (\tau, \lambda)}(t) \) within \( S_{zu}(t) \). To handle the nonconvexity of \( S_{zu}(t) \), we first compute a convex composition of \( S_{zu}(t) \). When constructing \( C_{zu, (\tau, \lambda)}(t) \), we let the reachable set at each time belong to one of the convex components in \( S_{zu}(t) \), encoded by MI linear inequalities. For details see [22]. The convex decomposition of \( S_{zu}(t) \) 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 ten largest hyper-boxes in \( S_{zu}(t) \) as the safe set. Given the

---

**Fig. 4.** Quadrotor trajectory in \( x-y \) plane: nominal trajectory (red), corrected trajectory (blue), supervision active (green). (a) Quadrotor position. (b) Quadrotor velocity. (c) Quadrotor acceleration.

**Fig. 5.** Robot operational space: initial position (yellow arrowhead), target position (cyan), and unsafe region (dark area).
constructed implicit CIS $C_{xv, (\tau, \lambda)}(t)$ at time $t$, we supervise the nominal control input $u(t)$ by solving $P(t, t^*)$ as discussed in Section V-C. Note that $P(t, t^*)$ becomes an MI 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 $(\tau, \lambda) = (6, 4)$, $T_s = 0.1 s$, $\tau = 5 m/s$, $\tau = 5 m/s^2$. The MI program $P(t, t^*)$ is implemented via YALMIP [24] and solved by GUROBI [16]. The average computation time for constructing $C_{xv, (\tau, \lambda)}(t)$ and solving $P(t, t^*)$ at each time step is 2.87 s. 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 $t = 78.6 s$, 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 $t = 1.2 s$. The full simulation video is made available at https://youtu.be/mB9ir0R9hzM.

### C. Scalability and Quality

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

$$A_n = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B_n = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

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 constrain 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. 7(a) and (b) show the times to compute the implicit CIS $C_{xv,q}$ for safe sets with $2n$ and $n^2$ constraints, respectively. $C_{xv,q}$ can be computed in less than 0.5 s for systems of size $n = 200$ when the safe set has $2n$ constraints, and in around 5 s for $n = 100$ and safe sets with $n^2$ constraints, that is 10 000 constraints in this example.

We now proceed to the case where system disturbances are present. In Fig. 8(a) and (b), 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 Minkowski 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 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 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 of our method, we still present a few cases for the sake of comparison. We compare our approach to the Multiparametric Toolbox (MPT3) [17] that computes the maximal (R)CIS $C_{\max}$, 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 $C_{\max}$ is apparent from the steep corresponding curve. The low-complexity methods in [34] and [21] are considerably faster, and the one in [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 $C_{\max}$ 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 $C_{\max}$ even with small values of $(\tau, \lambda)$ 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 $C_{\max}$.

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 $C_{outer,\nu}$. We can appreciate that empirically $C_{outer,\nu}$ approximates very closely $C_{\max}$, even in the presence of disturbances, based on the size of the sets our method computes. However, the gap between $C_{outer,\nu}$ and $C_{\max}$ depends on the size of the disturbance as shown next.

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

VIII. RELATED LITERATURE

Recent works by the authors [1], [2], [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], and [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 semidefinite 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 QPs (QCQP) that stem from addition, for online control problems, like MPC and supervisory control, polytopes are preferred to ellipsoids, as they result in better quality in terms of approximating the maximal CIS. In these cases, it is easy to verify from the derived closed-form expression.

A. Claims of Theorem 6.6

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

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

$$C_{outer,\nu} = \{ x \in \mathbb{R}^n \mid \exists \{u_i\}_{i=0}^{\nu-1} \in \mathbb{R}^{nu},$$

$$\big\{ (x, \{u_i\}_{i=0}^{\nu-1}), u_t \big\} \in \mathcal{S}_{xu}, t = 0, \ldots, \nu - 1, \bigwedge$$

$$\mathcal{R}_\Sigma(x, \{u_i\}_{i=0}^{\nu-1}) \subseteq \mathcal{S}_{xu}, t = 0, \ldots, \nu - 1,$$

Due to Lemma 6.2, $\mathcal{R}_\Sigma(x, \{u_i\}_{i=0}^{\nu-1}) \subseteq C_{max} + W_k$ if and only if $\sum_{i=0}^{\nu-1} A^{\nu-1-i} Bu_i \in C_{max}$. Based on this observation, it is easy to verify that $C_{outer,\nu} = \pi_{\nu}(C_{xv,max})$, where $C_{xv,max} = C_{xv,0} \cap \{ x \in \mathbb{R}^n \mid u \in (C_{max}) \}$. Using the matrices $H$ and $P$, as in (17), the definition of $C_{xv,\nu}$, and Lemma 6.2, we write $C_{xv,\nu}$ as:

$$C_{xv,\nu} = \{ \{x_0, \{u_t\}_{t=\nu+1}^{\infty}\} \}$$

$$\big\{ \{x_0, \{u_t\}_{t=\nu+1}^{\infty}\}, u_t \big\} \subseteq \mathcal{S}_{xu}, t = 0, \ldots, \nu - 1,$$

$$\bigwedge$$

$$\mathcal{R}_\Sigma\left( \sum_{i=1}^{\nu} A^{\nu-1-i} Bu_{t-i}, \{u_{t-i}\}_{i=0}^{\nu-1}, u_{\nu+k} \right) \subseteq \mathcal{S}_{xu},$$

$$k = 0, \ldots, \tau + \lambda - 1 \}.$$

(48)

By (48), the projection $\pi_{\nu+\nu}(C_{xv,\nu})$ is:

$$\pi_{\nu+\nu}(C_{xv,\nu}) = \{ \{x_0, \{u_t\}_{t=\nu+1}^{\infty}\} \mid \exists u_{t=\nu+\nu},$$

$$\big\{ \{x_0, \{u_t\}_{t=\nu+1}^{\infty}\}, u_t \big\} \subseteq \mathcal{S}_{xu}, t = 0, \ldots, \nu - 1,$$

$$\bigwedge$$

$$\mathcal{R}_\Sigma\left( \sum_{i=1}^{\nu} A^{\nu-1-i} Bu_{t-i}, \{u_{t-i}\}_{i=0}^{\nu-1}, u_{\nu+k} \right) \subseteq \mathcal{S}_{xu},$$

$$k = 0, \ldots, \tau + \lambda - 1 \}.$$

(49)

Again, using the matrices $H$ and $P$, as in (17), the definition of $C_{xv,\nu}$, we have:

$$\mathcal{C}_{xv,\nu} = \{ x_0 \in \mathbb{R}^n \mid \exists u_{t=\nu+\nu},$$

$$\mathcal{R}_\Sigma\left( \sum_{i=1}^{\nu} A^{\nu-1-i} Bu_{t-i}, \{u_{t-i}\}_{i=0}^{\nu-1}, u_{\nu+k} \right) \subseteq \mathcal{S}_{xu},$$

$$k = 0, \ldots, \tau + \lambda - 1 \}.$$
Comparing the right-hand sides of (49) and (50), we have:

\[
\pi_{n+vm}(C_{x(v,\tau)}) \subseteq \left\{ (x_0, u_{0:u-1}) \mid \begin{array}{l}
\sum_{i=1}^{\nu} A^{-1}B_{u_{i-1}} \\
\widetilde{R}_S(x, \{u_i\}_{i=0}^{\nu-1}), \{u_i\}_{i=0}^{\nu-1} \subseteq S_{xu}, t = 0, \ldots, \nu - 1,
\end{array} \right\}.
\]  

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

\[
\widetilde{C}_{x,v,(\nu,\tau)} = \pi_{n+vm}(C_{x(v,\tau)}).
\]  

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

Proof of Claim 4: We define the \( k \)-step null-controllable set \( C_k \) as the set of states of \( \Sigma \) that reach the origin at the \( k \)th step under the state-input constraints \( S_{xu} \):

\[
C_k = \{ x \in \mathbb{R}^n \mid \exists 0:k-1 \in \mathbb{R}^{km} \}.
\]  

From (49) and (50), we have:

\[
\pi_{n+vm}(C_{x(v,\tau)}) \subseteq \left\{ (x_0, u_{0:u-1}) \mid \begin{array}{l}
\sum_{i=1}^{\nu} A^{-1}B_{u_{i-1}} \\
\widetilde{R}_S(x, \{u_i\}_{i=0}^{\nu-1}), \{u_i\}_{i=0}^{\nu-1} \subseteq S_{xu}, t = 0, \ldots, \nu - 1,
\end{array} \right\}.
\]  

Obviously, \( C_0 = \emptyset \). Since \( A^T = 0 \) and the fixed point \( (0,0) \in \mathbb{R}^n \times \mathbb{R}^m \) is in the interior of \( S_{xu} \), there exists an \( \epsilon > 0 \) such that the \( \epsilon \)-ball \( B_{\epsilon}(0) \) at the origin satisfies that for \( u_{0:u-1} = 0 \in \mathbb{R}^{vm} \) and for all \( t \in [0, \nu - 1] \):

\[
\widetilde{R}_S(B_{\epsilon}(0), \{u_i\}_{i=0}^{\nu-1}), \{u_i\}_{i=0}^{\nu-1} = (A^T B_{\epsilon}(0), 0) \subseteq S_{xu}
\]  

By (54) and the definition of \( C_k \), \( B_k(0) \) is contained by \( C_k \), and thus, \( C_0 = \emptyset \) is contained in the interior of \( C_k \). Then, by (23, Th. 1), since \( C_0 \) is contained in the interior of \( C_k \), 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 \( d(C_k, \overline{C}_{x,v}) \) satisfies that:

\[
d(C_k, \overline{C}_{x,v}) \leq c_2 a^k.
\]  

Furthermore, let \( k = \tau_2 \). For any \( x \in C_{\tau_2} \) and the corresponding \( u_{0:u-1} \) satisfying the constraints on the right-hand side of (53), it is easy to check that \( (x, u_{0:u-1}) \in \mathbb{R}^n \times \mathbb{R}^{(\tau_2+1)m} \) is contained in \( C_{x,v,(\nu,\tau_2)} \). Thus, we have for all \( \tau > 0 \):

\[
C_{\tau} \subseteq \overline{C}_{x,(\nu,\tau,m)} \subseteq \overline{C}_{x,v}.
\]  

Select a big enough \( \tau_1 \) 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 \( \tau \geq \tau_1 \):

\[
\overline{C}_{x,(\nu,\tau,m)} \supseteq (1 - c_2 a^\tau) \overline{C}_{x,v}.
\]  

where \( c_0 = c_2 / \epsilon \). Thus, the fourth claim is proven.

REFERENCES

[1] T. Anevlavis and P. Tabuada, “Computing controlled invariant sets in two moves,” in Proc. IEEE 58th Conf. Decis. Control, 2019, pp. 6248–6254.

[2] T. Anevlavis, Z. Liu, N. Ozay, and P. Tabuada, “An enhanced hierarchy for (robust) controlled invariance,” in Proc. Amer. Control Conf., 2021, pp. 4860–4865.

[3] T. Anevlavis and P. Tabuada, “A simple hierarchy for computing controlled invariant sets,” in Proc. 23rd Int. Conf. Hybrid Syst.: Comput. Control, 2020, pp. 1–11.

[4] P. J. Antsaklis and A. N. Michel, Linear Systems, vol. 8. Berlin, Germany: Springer, 1997.

[5] A. Başçık, S. Bansal, E. Brontstein, V. Tolani, and C. J. Tomlin, “An efficient reachability-based framework for provably safe autonomous navigation in unknown environments,” in Proc. IEEE 58th Conf. Decis. Control, 2019, pp. 1758–1765.

[6] D. Bertsekas, “ Infinite time reachability of state-space regions by using feedback control automatic control,” IEEE Trans. Autom. Control, vol. 17, no. 5, pp. 604–613, Oct. 1972.

[7] S. Boyd and L. Vandenberghe, Convex Optimization. New York, NY, USA: Cambridge Univ. 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] N. Pereira, A. Rowe, M. Farb, I. Liang, E. Lu, and E. Riebling, “ARENA - the augmented reality edge networking architecture,” IEEE Int. Sympos. Mixed Augmented Reality, Oct. 2021.

[11] J. D. Glover and F. C. Schweppe, “Control of linear dynamic systems with set constrained disturbances automatic control,” IEEE Trans. Automat. Control, vol. 16, no. 5, pp. 411–423, Oct. 1971.

[12] Z. Liu and N. Ozay, “Safe online planning in unknown nonconvex environments with implicit controlled invariant sets,” in Proc. 54th IEEE Conf. Decis. Control, 2015, pp. 284–289.

[13] Z. Liu and N. Ozay, “On the convergence of the backward reachable sets of geometric control systems,” in Proc. 54th IEEE Conf. Decis. Control, 2015.

[14] M. Lazar and V. Spinu, “Finite-step terminal ingredients for stabilizing unknown environments,” in Proc. 54th IEEE Conf. Decis. Control, 2015.

[15] M. Lazar and V. Spinu, “Finite-step terminal ingredients for stabilizing unknown environments,” in Proc. 54th IEEE Conf. Decis. Control, 2015.

[16] M. Lazar and V. Spinu, “Finite-step terminal ingredients for stabilizing unknown environments,” in Proc. 54th IEEE Conf. Decis. Control, 2015.

[17] M. Lazar and V. Spinu, “Finite-step terminal ingredients for stabilizing model predictive control,” IFAC-PapersOnLine, vol. 48, no. 23, pp. 9–15, 2015.

[18] M. Lazar and V. Spinu, “Finite-step terminal ingredients for stabilizing model predictive control,” IFAC-PapersOnLine, vol. 48, no. 23, pp. 9–15, 2015.

[19] M. Lazar and V. Spinu, “Finite-step terminal ingredients for stabilizing model predictive control,” IFAC-PapersOnLine, vol. 48, no. 23, pp. 9–15, 2015.

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

His research interests include safety-critical control, motion and trajectory planning, and formal methods in control of cyber-physical systems.

Zexiang Liu (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, in 2018, from the University of Michigan, Ann Arbor, MI, USA, where he is currently working toward the Ph.D. degree in electrical and computer engineering.

His research interests include 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 the Chen-Luan Family Faculty Development Professor of electrical and computer engineering, Associate Professor of electrical engineering and computer science, and Associate Professor of robotics. 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. 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 from the Institute for Systems and Robotics (a private research institute associated with Instituto Superior Tecnico), Lisbon, Portugal, in 2002.

Between 2002 and 2003, he was a Postdoctoral Researcher with the University of Pennsylvania, Philadelphia, PA, USA. After spending three years with the University of Notre Dame, Notre Dame, IN, USA, 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 a Vijay K. Dhir Professor of Engineering. Dr. Tabuada is a Fellow of IFAC. He was the recipient of multiple awards including the NSF CAREER award in 2005, the Donald P. Eckman award in 2009, the George S. Axelby award in 2011, and the Antonio Ruberti Prize in 2015, for his contributions to control and cyber-physical systems. He has been a Program Chair and General Chair for several conferences in the areas of control and of cyber-physical systems, such as IFAC Conference on Networked Systems, ACM International Conference on Hybrid Systems: Computation and Control (HSCC), and International Conference on Cyber-Physical Systems. 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.