Implicit Invariant Sets for High-Dimensional Switched Affine Systems

Andrew Wintenberg and Necmiye Ozay

***Abstract*— In this paper we consider the problem of robust control invariant set computation for discrete-time switched affine systems. We consider additive disturbances and joint state-input constraints. We provide several constructions of N-step recurrent sets from which we derive an implicit description of invariant sets. This is done with a linear program characterizing set recurrence and backwards reachability with affine feedback controllers. The focus of these methods is scalability to systems with high-dimension. The performance of these methods is evaluated on a system modeling consensus of unmanned aerial vehicles (UAVs) and a system modeling the large-scale control of thermostatically controlled loads (TCLs).**

I. INTRODUCTION

Set invariance is one of the main tools used to reason about constraint satisfaction for dynamical systems. As such, computation of invariant sets is crucial for safety analysis or safe-by-design control [1], [2]. There are a variety of techniques to compute invariant sets for different classes of systems (e.g., linear [3] vs. switched [4]), using different types of set representations (e.g., polytopic [5] vs. ellipsoidal [6]), or different computational approaches (e.g., set iterations [7] vs. optimization [8]). These methods generally exhibit a trade-off between conservativity and computational complexity. Within this trade-off space, there is a need for scalable methods that can be applied to high-dimensional systems. In this paper, we develop such a method for invariant set computation for switched affine systems.

Recently [9], [10] have proposed an efficient optimization-based computation of control invariant sets for constrained deterministic linear systems. They relax the one-step invariance condition and search for a set that can be driven into itself in N-steps, which we call N-step recurrent. They then construct invariant sets from these recurrent sets. Our main contributions are to extend the results of [9], [10] to systems with non-determinism in the form of additive disturbances and discrete switches. We also provide several constructions of recurrent sets. Although not presented in this way, [9], [10] essentially characterize N-step recurrence in terms of linear open-loop controllers. To account for non-determinism in our setting, we characterize these sets with affine feedback controllers. Using an appropriate parameterization of these controllers, we derive a linear program to check set recurrence. Then we derive an implicit linear representation of an invariant set from a given recurrent set. While these methods are conservative, they can be applied to high-dimensional systems due to the scalability of linear program solvers. We demonstrate this with several examples.

Andrew Wintenberg and Necmiye Ozay are with the Department of Electrical Engineering and Computer Science at The University of Michigan, Ann Arbor. This work is supported by an Early Career Faculty grant from NASA's Space Technology Research Grants Program. AW is also supported by NSF Grant CNS-1738103. Email: {awintenb, necmiye}@umich.edu

A. Notation and Preliminaries

A polytope is a set defined by linear constraints $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ for some $n \in \mathbb{R}^n$. In this definition, polytopes are convex but need not be bounded. The convex hull of a set S is denoted by $\operatorname{conv} S$. The Minkowski sum of sets S, T is denoted by $S \oplus T$. To avoid the computational complexity of vertex enumeration in high dimensions, all input polytopes are assumed to be in halfspace form. The following standard lemma provides linear conditions for the containment of polytopes in halfspace form.

Lemma 1.1 (H-Polytope in H-Polytope [11, Lemma 1]): Let $\mathcal{P} = \{x \in \mathbb{R}^{n_p} \mid Px \leq p\}$ and $\mathcal{Q} = \{y \in \mathbb{R}^{n_q} \mid Qy \leq q\}$ be polytopes. Then the containment $\mathcal{P} \subseteq \mathcal{Q}$ holds if and only if there exists a nonnegative matrix T such that

$$TP = Q, \tag{1}$$

$$Tp \leq q. \tag{2}$$

II. PROBLEM DESCRIPTION

Consider a discrete-time switched affine system:

$$x^+ = A_\sigma x + B_\sigma u + w + f_\sigma\,, \tag{3}$$

where $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ is the control input, $w \in \mathbb{R}^n$ is the disturbance, and $\sigma \in \Sigma$ is the discrete switch mode. The disturbance and switch mode are exogenous in this setting; they are not control inputs. We assume that Σ is a finite set. Initially we assume that the disturbance and switch mode are measured after the controller selects an input. We later relax this assumption in Section III-A.

We consider polytopic constraints on the state and input $XU \subseteq \mathbb{R}^{n+m}$ that should be satisfied when the disturbance is constrained to the polytope $W \subseteq \mathbb{R}^n$. We represent sets as polytopes because they can approximate any convex set. We would like to compute an invariant set of states $x(0)$ for which we can guarantee for all times t and all disturbances $w(t) \in W$ and switching modes $\sigma(t)$, it holds that $(x(t), u(t)) \in XU$ for some choice of $u(t)$. To formalize this notion, the predecessors of a set $\Omega \subseteq \mathbb{R}^n$ constrained by XU are defined by

$$\text{Pre}(\Omega) = \{x \mid \exists u \;\; (x,u) \in XU,\; \forall w \in W, \sigma \in \Sigma \;\; x^+ \in \Omega\}\,. \tag{4}$$

The states in $\text{Pre}(\Omega)$ can be guaranteed to reach Ω in one step subject to all constraints. Invariance is then commonly defined as follows.

Definition 2.1: A set $\Omega \subseteq \mathbb{R}^n$ is said to be *robust control invariant* or simply *invariant* with respect to the constraints XU if $\Omega \subseteq \text{Pre}(\Omega)$.

A set is invariant if it can return to itself in one step, so we say that invariant sets are one-step recurrent. As in [10], it can be useful to consider sets that can return to themselves in exactly N steps. These sets can be easier to construct with simpler representations. Let Pre^N denote the N-fold composition of Pre. We say $\Omega \subseteq \mathbb{R}^n$ is *N-step recurrent* if $\Omega \subseteq \text{Pre}^N(\Omega)$. We can relate N-step recurrent sets to invariant sets with the following key observation. If Ω is N-step recurrent and we define

$$\Omega_\tau = \bigcup_{i=1}^{\tau} \text{Pre}^i(\Omega) \quad \text{for} \quad \tau \geq N\,, \tag{5}$$

then Ω_τ is invariant. Here the set Ω_τ is the set of states that can reach Ω in exactly t steps for some $t \in \{1, \cdots, \tau\}$, i.e., it is a type of backwards reachable set [1]. Additionally using the properties of switched affine systems with convex constraints, if S is invariant then the convex hull $\text{conv}\, S$ is also invariant. So then $\text{conv}\, \Omega_\tau$ is a polytopic invariant set.

A. Invariant set computation approach

These observations motivate the following approach to compute invariant sets: first determine an N-step recurrent set Ω and then approximate the backwards reachable set Ω_τ for a horizon $\tau \geq N$. This is related to the approach used in [10] in the context of linear systems without disturbances. While this approach need not generate the maximal invariant set even as $\tau \to \infty$, it can provide efficient representations of invariant sets. However, direct computation of the Pre operator may be intractable in higher-dimensional systems. This is because the standard method of computation involves lifting to a higher-dimensional space followed by projection [12], [13]. Projection of polytopes with halfspace representations quickly becomes intractable in higher dimensions. In fact, there is no polynomial-time algorithm for projection [14].

To avoid this issue in directly verifying the set inclusion $\Omega \subseteq \text{Pre}^N(\Omega)$ and in computing Ω_τ, we consider the action of explicit controllers. The set Ω is recurrent if and only if there exists a controller that drives states of Ω into itself in N-steps. While such a controller may necessarily be nonlinear, we can perform efficient but conservative computations by restricting our attention to affine feedback controllers. We apply ideas from finite horizon control [2], [15] and system level synthesis [16] to first develop a linear program to conservatively verify N-step recurrence and then to construct an implicit representation of an invariant set that approximates Ω_τ.

[1] We may be able to guarantee a state can reach Ω within τ steps, but due to nondeterminism not be able to provide a specific time $t \leq \tau$ when this is accomplished. Such states are not contained in Ω_τ. While the set of states that can reach Ω within τ steps is invariant under the assumptions and contains Ω_τ, its computation would require computing the predecessors of unions of polytopes, which may be intractable.

III. AFFINE FEEDBACK CONTROL

In order to describe recurrent and invariant sets, we consider controllers that achieve recurrence or invariance. To this end, we consider a general controller design problem with parameters $\langle X_I, X_F, N\rangle$ representing the initial state polytope, target state polytope, and horizon, respectively. A solution to this problem is a controller that guarantees for any initial state in X_I and any realization of disturbances in W^N and switch modes in Σ^N, the system reaches X_F in N-steps while always satisfying the constraints in XU. A controller maps state, disturbance, and switching measurements into control inputs. To formalize the problem, we define

$$\mathbf{x} = (x(0), \cdots, x(N-1), x(N)) \tag{6}$$
$$\mathbf{u} = (u(0), \cdots, u(N-1)) \tag{7}$$
$$\mathbf{w} = (w(0), \cdots, w(N-1)) \tag{8}$$
$$\boldsymbol{\sigma} = (\sigma(0), \cdots, \sigma(N-1))\,. \tag{9}$$

We also define

$$\begin{aligned}\mathscr{Z} = \{(\mathbf{x}, \mathbf{u}) \mid \forall t \in \{0, \cdots, N-1\},& \\ (x(t), u(t)) \in XU, \quad x(N) \in X_F\}\,.&\end{aligned} \tag{10}$$

Note $\mathscr{Z}$ is simply a permutation of $XU^N \times X_F$ that represents admissible state-input trajectories and is also a polytope. As in [16], a controller uniquely induces a map $\mathbf{\Phi} : X_I \times W^N \times \Sigma^n \to X^{N+1} \times U^N$ representing the closed-loop system's state-input response to the initial state, disturbances, and switching over the horizon N so $(\mathbf{x}, \mathbf{u}) = \mathbf{\Phi}(x(0), \mathbf{w}, \boldsymbol{\sigma})$. We can then define solutions to the design problem in terms of the closed-loop response.

Definition 3.1: A controller inducing the closed-loop response $\mathbf{\Phi}$ is a solution to the design problem for $\langle X_I, X_F, N\rangle$ if

$$\forall \boldsymbol{\sigma} \in \Sigma^N \;\; \mathbf{\Phi}(X_I, W^N, \boldsymbol{\sigma}) \subseteq \mathscr{Z}. \tag{11}$$

That is for every initial state and sequence of disturbances and switch modes, the closed-loop dynamics produce an admissible trajectory satisfying the constraints. Furthermore, it is clear that the design problem has a solution if and only if $X_I \subseteq \text{Pre}^N(X_F)$. To make the problem tractable, we only consider controllers that are affine in $x(0)$ and $w(0), \cdots, w(t-1)$ for a fixed $(\sigma(0), \cdots, \sigma(t-1))$:

$$\begin{aligned}u(t) = K_t(\sigma(0), \cdots, \sigma(t-1))x(0) +& \\ \sum_{i=0}^{t-1} L_{t,i}(\sigma(0), \cdots, \sigma(t-1))w(i) +& \\ r_t(\sigma(0), \cdots, \sigma(t-1))\,.&\end{aligned} \tag{12}$$

Note that because the dynamics are switched affine, the state $x(t)$ can be written as an affine function of $x(0)$ and $u(\tau), w(\tau)$ for $\tau < t$ for a fixed switching sequence. Hence controllers that are affine in $x(\tau)$ for $\tau \leq t$ can be written in the form of (12). As such, we refer to the disturbance-feedback controllers in the form of (12) simply as *affine feedback controllers*. For any such controller, we can write

$$\mathbf{u} = \mathbf{K}(\boldsymbol{\sigma})x(0) + \mathbf{L}(\boldsymbol{\sigma})\mathbf{w} + \mathbf{r}(\boldsymbol{\sigma})\,, \tag{13}$$

where $\mathbf{L}$ is lower block triangular and the blocks of $\mathbf{K}, \mathbf{L}, \mathbf{r}$ are for $i < t < N$ given by

$$\mathbf{K}_t(\boldsymbol{\sigma}) = K_t(\sigma(0), \cdots, \sigma(t-1)) \tag{14}$$

$$\mathbf{L}_{t,i}(\boldsymbol{\sigma}) = L_{t,i}(\sigma(0), \cdots, \sigma(t-1)) \tag{15}$$

$$\mathbf{r}_t(\boldsymbol{\sigma}) = r_t(\sigma(0), \cdots, \sigma(t-1)) \tag{16}$$

This parameterization avoids the difficulty of multiplicative composition of gains in state feedback. For a more thorough discussion of this see [15]. The set of affine controllers solving the design problem for $\langle X_I, X_F, N \rangle$ is defined by

$$\mathcal{C} = \{(\mathbf{K}, \mathbf{L}, \mathbf{r}) \mid \forall \boldsymbol{\sigma} \in \Sigma^N \ \boldsymbol{\Phi}(X_I, W^N, \boldsymbol{\sigma}) \subseteq \mathcal{Z}\} \tag{17}$$

To characterize these solutions we observe the following.

Proposition 3.2: $\mathcal{C}$ is a polytope.

Proof: Consider an affine controller defined by $(\mathbf{K}, \mathbf{L}, \mathbf{r})$ and its induced system response $\boldsymbol{\Phi}$. By gathering terms, we can rewrite the dynamics over the horizon as

$$\mathbf{x} = G_x(\boldsymbol{\sigma})x(0) + G_u(\boldsymbol{\sigma})\mathbf{u} + G_w(\boldsymbol{\sigma})\mathbf{w} + g(\boldsymbol{\sigma}) \,. \tag{18}$$

Then we see that $\boldsymbol{\Phi}$ is affine in $x(0)$ and $\mathbf{w}$ for a fixed $\boldsymbol{\sigma}$:

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{u}\end{bmatrix} = \boldsymbol{\Phi}(x(0), \mathbf{w}, \boldsymbol{\sigma}) = \Phi_x(\boldsymbol{\sigma})x(0) + \Phi_w(\boldsymbol{\sigma})\mathbf{w} + \phi(\boldsymbol{\sigma}) \,, \tag{19}$$

where

$$\Phi_x(\boldsymbol{\sigma}) = \begin{bmatrix} G_x(\boldsymbol{\sigma}) + G_u(\boldsymbol{\sigma})\mathbf{K}(\boldsymbol{\sigma}) \\ \mathbf{K}(\boldsymbol{\sigma}) \end{bmatrix} \tag{20}$$

$$\Phi_w(\boldsymbol{\sigma}) = \begin{bmatrix} G_w(\boldsymbol{\sigma}) + G_u(\boldsymbol{\sigma})\mathbf{L}(\boldsymbol{\sigma}) \\ \mathbf{L}(\boldsymbol{\sigma}) \end{bmatrix} \tag{21}$$

$$\phi(\boldsymbol{\sigma}) = \begin{bmatrix} g(\boldsymbol{\sigma}) + G_u(\boldsymbol{\sigma})\mathbf{r}(\boldsymbol{\sigma}) \\ \mathbf{r}(\boldsymbol{\sigma}) \end{bmatrix} \,. \tag{22}$$

Given the polytopes $X_I, W^N, \mathcal{Z}$ in halfspace form as

$$X_I \times W^N = \{z \mid Hz \leq h\}, \quad \mathcal{Z} = \{q \mid Sq \leq s\}, \tag{23}$$

we define

$$\begin{aligned} A(\boldsymbol{\sigma}) &= \{(x, \mathbf{w}) \mid \boldsymbol{\Phi}(x, \mathbf{w}, \boldsymbol{\sigma}) \in \mathcal{Z}\} \\ &= \{(x, \mathbf{w}) \mid S(\Phi_x(\boldsymbol{\sigma})x + \Phi_w(\boldsymbol{\sigma})\mathbf{w}) \leq s - S\phi(\boldsymbol{\sigma})\} \end{aligned}$$

Rearranging the definition of $\mathcal{C}$, note that $(\mathbf{K}, \mathbf{L}, \mathbf{r}) \in \mathcal{C}$ holds if and only if $\forall \boldsymbol{\sigma} \in \Sigma^N \ X_I \times W^N \subseteq A(\boldsymbol{\sigma})$ which is a polytope containment. By expressing this with linear constraints using Lemma 1.1 we have

$$\mathcal{C} = \{(\mathbf{K}, \mathbf{L}, \mathbf{r}) \mid \forall \boldsymbol{\sigma} \in \Sigma^N \ \exists T(\boldsymbol{\sigma}) \geq 0, \tag{24}$$

$$T(\boldsymbol{\sigma})H = S\begin{bmatrix}\Phi_x & \Phi_w\end{bmatrix}, \tag{25}$$

$$T(\boldsymbol{\sigma})h \leq s - S\phi(\boldsymbol{\sigma})\} \,. \tag{26}$$

Because $(\mathbf{K}(\boldsymbol{\sigma}), \mathbf{L}(\boldsymbol{\sigma}), \mathbf{r}(\boldsymbol{\sigma}))$ enter linearly into $\Phi_x(\boldsymbol{\sigma}), \Phi_w(\boldsymbol{\sigma}), \phi(\boldsymbol{\sigma})$, $\mathcal{C}$ is the projection of a polytope defined over $(\mathbf{K}, \mathbf{L}, \mathbf{r}, T)$. Thus $\mathcal{C}$ is a polytope itself. ■

This result also shows that affine controller satisfying the design problem can be found by solving a linear program over $(\mathbf{K}, \mathbf{L}, \mathbf{r}, T)$. The number of constraints and decision variables in this linear program are polynomial n, m, l and in the number of halfspace constraints defining X_I, X_F, W, XU, and exponential in the horizon N. For systems without switching, these quantities are polynomial in the horizon. Despite the exponential complexity in the presence of switching, solving this linear program is tractable for small horizons and number of switching modes.

A. Unmeasured disturbance and switch mode

Now we consider the setting where the disturbance and switch mode are not measured. Given an affine feedback controller that satisfies the design problem, we construct a controller satisfying the design problem that only depends upon state measurements. Such a controller will then satisfy the design problem on the system without measurements. This is done by applying the affine feedback controller to an estimated history of disturbances and switch modes that is consistent with the state history,

Proposition 3.3: Given $(\mathbf{K}, \mathbf{L}, \mathbf{r})$ defines an affine controller inducing a system response $\boldsymbol{\Phi}$ that satisfies design problem for $\langle X_I, X_F, N \rangle$, then there exists a controller whose inputs only depend on state measurements inducing a system response $\hat{\boldsymbol{\Phi}}$ that satisfies the same problem.

Proof: At each time $t \in [N]$, the dynamics enforce that there is a solution $\hat{w}(t-1) = w(t-1), \ \hat{\sigma}(t-1) = \sigma(t-1)$ to the equation

$$x(t) = A_{\hat{\sigma}(t-1)}x(t-1) + B_{\hat{\sigma}(t-1)}u(t-1) + \hat{w}(t-1) + f_{\hat{\sigma}(t-1)} \,. \tag{27}$$

At time t, the controller knows $x(t), x(t-1), u(t-1)$ and can compute estimates $\hat{w}(t-1) \in W, \ \hat{\sigma}(t-1) \in \Sigma$ satisfying this equation by evaluating linear inequalities. Using the estimates, consider the controller defined by

$$\begin{aligned} u(t) = &K_t(\hat{\sigma}(0), \cdots, \hat{\sigma}(t-1))x(0) + \\ &\sum_{i=0}^{t-1} L_{t,i}(\hat{\sigma}(0), \cdots, \hat{\sigma}(t-1))\hat{w}(i) + \\ &r_t(\hat{\sigma}(0), \cdots, \hat{\sigma}(t-1)) \,. \end{aligned} \tag{28}$$

This controller depends only on the observed state history. Next we show that this controller satisfies the design requirements. Let $\hat{\boldsymbol{\Phi}}$ denote the system response induced by this controller. Consider any $x(0) \in X_I, \mathbf{w} \in W^N, \boldsymbol{\sigma} \in \Sigma^N$. By construction, it holds that $\hat{\boldsymbol{\Phi}}(x(0), \mathbf{w}, \boldsymbol{\sigma}) = \boldsymbol{\Phi}(x(0), \hat{\mathbf{w}}, \hat{\boldsymbol{\sigma}})$. As we assume that equation (11) holds for $\boldsymbol{\Phi}$ which states $\forall \boldsymbol{\sigma} \in \Sigma^N \ \boldsymbol{\Phi}(X_I, W^N, \boldsymbol{\sigma}) \subseteq \mathcal{Z}$, in particular it holds that $\boldsymbol{\Phi}(x(0), \hat{\mathbf{w}}, \hat{\boldsymbol{\sigma}}) \in \mathcal{Z}$. Thus $\hat{\boldsymbol{\Phi}}$ is a solution to the problem for $\langle X_I, X_F, N \rangle$. ■

So the linear program constructed in Proposition 3.2 can be used to solve the design problem with only state feedback. We can now characterize recurrent sets using this general controller design problem.

IV. RECURRENT SET ANALYSIS

While it is difficult to directly check if the set Ω is N-step recurrent with the set inclusion $\Omega \subseteq \text{Pre}^N(\Omega)$, we can effectively check the stronger condition that there exists an affine feedback controller driving Ω into itself. This is exactly the design problem of Section III for $\langle \Omega, \Omega, N \rangle$ which can be solved with a linear program. This result can be viewed as an extension of Theorem 2 in [9] to switched affine systems with additive disturbances. While we can test if a given set is recurrent with this result, we must now address the harder problem of constructing recurrent sets in order to compute invariant sets.

The construction of N-step recurrent sets is conceptually simpler than the construction of invariant sets, or 1-step recurrent sets. Even for simple systems, the maximal invariant set may be quite complex. As such, methods approximating the maximal invariant set often suffer from complex set representations. However for large N, N-step recurrent sets can be found with significantly simpler representations. For example, consider an autonomous system with an asymptotically stable equilibrium point. The region of attraction of this point is necessarily invariant, but may have arbitrarily complex geometry. Yet any sufficiently small neighborhood of the equilibrium point is a recurrent set. With this in mind, a naive but effective method for constructing recurrent sets is to select Ω with simple geometry, such as a hyperbox, about a desired operating point. We can efficiently iterate over a collection of simple candidates with the recurrent linear program to identify a recurrent set.

A. Heuristic generation of recurrent sets

In some cases, we can use known properties of the system to aid in the construction of recurrent sets. For example, in Section VI-B we can explicitly construct a recurrent set for a discrete-state system. We then use this set as a starting point to find a recurrent set for a linear relaxation of the system dynamics.

We can also use knowledge of the system to derive suitable parameterizations for candidate recurrent sets. For example in Section VI-A, we identify a number of linear constraints relevant to recurrence by analyzing the system dynamics. In such cases we can apply a heuristic algorithm inspired by the inside-out algorithm from [17] that tries to find a recurrent set with a given parameterization. While the original algorithm attempts to grow a fixed set into a invariant set containing it, our version grows a fixed set into the "smallest" recurrent set containing it.

Let $Q \in \mathbb{R}^{n_q \times n}$ be a set of halfspace normals such that for all $q \in \mathbb{R}^{n_q}$ with $q \geq 0$ the polytope

$$\Omega(q) = \{x \mid Qx \leq q\} \tag{29}$$

contains the point $x = 0$. We initialize the algorithm with a small set, for example $q(0) = 0$ corresponding to $\Omega(q(0)) = \{0\}$. In each iteration, we find a set $\Omega(q(i+1))$ that we can guarantee is reachable in N steps from $\Omega(q(i))$ with $q(i+1) \geq q(i)$. To do this, we consider the design problem for $\langle \Omega(q(i)), \Omega(q(i+1)), N \rangle$ and the associated linear program. Because $q(i+1)$ enters into the constraints linearly, we can use $q(i+1)$ as a decision variable in this linear program. As there is not a unique minimal choice of $q(i+1)$, we define costs $c \in \mathbb{R}^{n_q}$ with $c \geq 0$ associated with each constraint, and solve the linear program with the objective to minimize $c^T q(i+1)$. We terminate the algorithm when $q(i+1) = q(i)$ as this implies that $\Omega(q(i))$ is N-step recurrent, or when no feasible $q(i+1)$ is found. This method is shown in Algorithm 1. This heuristic algorithm is applicable to systems with natural parameterizations as seen in Section VI-A..

Algorithm 1 Inside-Out algorithm for computation of recurrent sets with a given parameterization

Input: Number of steps N, Normals Q, Costs $c \geq 0$
Output: A recurrent parameterization q^f

$q \leftarrow 0$
while true do
  $q^f \leftarrow \arg\min_{\tilde{q} \geq q} c^T \tilde{q}$ $s.t.$ $\Omega(q)$ can reach $\Omega(\tilde{q})$
  if Infeasible **then**
    return $\emptyset$
  end if
  if $q^f == q$ **then**
    return q^f
  end if
end while

V. BACKWARDS REACHABLE SET ANALYSIS

We can now construct invariant sets based upon the recurrent sets generated with the above methods. Recall if Ω is N-step recurrent, then the τ-step backwards reachable set Ω_τ defined by equation (5) is invariant for any $\tau \geq N$. Again direct computation of the Pre operator to construct Ω_τ may be intractable in higher dimension. However, by restricting to affine feedback control, we can construct an implicit representation of an invariant set that approximates Ω_τ. Consider a set of states $\Omega \subseteq \mathbb{R}^n$ and a horizon length N. Note that $x^* \in \text{Pre}^N(\Omega)$ if and only if there is a solution to the design problem $\langle \{x^*\}, \Omega, N \rangle$. We then consider the set of points that can reach Ω with an affine controller

$$P_N = \{x^* \mid \exists (\mathbf{K}, \mathbf{L}, \mathbf{r}) \; \forall \boldsymbol{\sigma} \in \Sigma^N \; \mathbf{\Phi}(\{x\}, W^N, \boldsymbol{\sigma}) \subseteq \mathcal{Z}\} \tag{30}$$

We can efficiently represent P_N using the following result.

Proposition 5.1: P_N is a polytope.

Proof: Given any $(\mathbf{K}, \mathbf{L}, \mathbf{r})$ defining a solution to the design problem for $\langle \{x^*\}, \Omega, N \rangle$, we can define another controller that is a solution by $(0, \mathbf{L}', \mathbf{r}')$ where $\mathbf{L}' = \mathbf{L}$ and $\mathbf{r}' = \mathbf{K}x^* + \mathbf{r}$. So it suffices to consider affine controllers with $\mathbf{K} = 0$. For such a $(0, \mathbf{L}, \mathbf{r})$, we can write the system response as in equation (19) as

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \mathbf{\Phi}(x^*, \mathbf{w}, \boldsymbol{\sigma}) = \Phi_x(\boldsymbol{\sigma})x(0) + \Phi_w(\boldsymbol{\sigma})\mathbf{w} + \phi(\boldsymbol{\sigma})\,. \tag{31}$$

Given the polytopes $W^N, \mathcal{Z}$ in halfspace form as

$$W^N = \{z \mid Hz \leq h\}, \quad \mathcal{Z} = \{q \mid Sq \leq s\}, \tag{32}$$

we define

$$A(\boldsymbol{\sigma}) = \{\mathbf{w} \mid \mathbf{\Phi}(\{x^*\}, \mathbf{w}, \boldsymbol{\sigma}) \in \mathcal{Z}\} \tag{33}$$

$$= \{\mathbf{w} \mid S\Phi_w(\boldsymbol{\sigma})\mathbf{w} \leq s - S(\phi(\boldsymbol{\sigma}) + \Phi_x(\boldsymbol{\sigma})x^*)\}. \tag{34}$$

So $x^* \in P_N$ if and only if there exists a solution $(0, \mathbf{L}, \mathbf{r})$ to the design problem or equivalently $\forall \boldsymbol{\sigma} \in \Sigma^N \; W^N \subseteq A(\boldsymbol{\sigma})$. So as in Proposition 3.2, we can use Lemma 1.1 to express P_N with linear constraints by

$$P_N = \{x^* \mid \exists (0, \mathbf{L}, \mathbf{r}) \; \forall \boldsymbol{\sigma} \in \Sigma^N \; \exists T(\boldsymbol{\sigma}) \geq 0, \tag{35}$$
$$T(\boldsymbol{\sigma}) H = S \Phi_w(\boldsymbol{\sigma}), \tag{36}$$
$$S\Phi_x(\boldsymbol{\sigma}) x^* + T(\boldsymbol{\sigma}) h \leq s - S\phi(\boldsymbol{\sigma})\}\,. \tag{37}$$

Because $\Phi_x(\boldsymbol{\sigma})$ is constant and $\Phi_w(\boldsymbol{\sigma}), \phi(\boldsymbol{\sigma})$ enter linearly into $\mathbf{L}(\boldsymbol{\sigma}), \mathbf{r}(\boldsymbol{\sigma})$ for a fixed $\boldsymbol{\sigma}$, we see P_N is the projection of a polytope. Hence P_N is a polytope itself. ■

Using the P_N we can construct an invariant set as a backwards reachable set restricted to affine control. Suppose that Ω is N-step recurrent with an affine controller, i.e., the linear program associated with the design problem $\langle \Omega, \Omega, N \rangle$ is feasible. Using the same reasoning for why Ω_τ is invariant, we observe that $\bigcup_{i=1}^{\tau} P_i$ is invariant for $\tau \geq N$. Although this set is a union of polytopes and may not be convex, we can again apply the result that the convex hull of an invariant set is invariant for these systems. So we define the set

$$\tilde{\Omega}_\tau = \operatorname{conv} \bigcup_{i=1}^{\tau} P_i\,, \tag{38}$$

which is invariant for $\tau \geq N$. We can now implicitly describe this set with linear constraints.

Proposition 5.2: Given representations of P_i as projections of polytopes as in Proposition 5.1

$$\forall i \; P_i = \{x \in \mathbb{R}^n \mid \exists y \in \mathbb{R}^{n_i}, \; Q_i x + R_i y \leq p_i\}\,, \tag{39}$$

where y is a vector encoding $\mathbf{L}, \mathbf{r}, T$ then

$$\begin{aligned} \tilde{\Omega}_\tau = \{x \in \mathbb{R}^n \mid & \exists \lambda \geq 0, \; \sum_{i=1}^{\tau} \lambda_i = 1 \\ & \forall i \in [\tau] \; \exists \tilde{x}_i \in \mathbb{R}^n \; \tilde{y}_i \in \mathbb{R}^{n_i}, \\ & \sum_{i=1}^{\tau} \tilde{x}_i = x, \\ & \forall i \in [\tau] \; Q_i \tilde{x}_i + R_i \tilde{y}_i \leq p_i \lambda_i\}\,. \end{aligned} \tag{40}$$

Proof: It is known that

$$\operatorname{conv} \bigcup_{i=1}^{\tau} P_i = \bigcup_{\substack{\lambda \geq 0 \\ \sum_{i=1}^{\tau} \lambda_i = 1}} \bigoplus_{i=1}^{\tau} \lambda_i P_i\,. \tag{41}$$

Next using the substitutions $\tilde{x}_i = \lambda_i x_i$ and $\tilde{y}_i = \lambda_i y_i$, observe that

$$\lambda_i P_i = \{\lambda_i x_i \mid \exists y_i, \; Q x_i + R y_i \leq p_i\} \tag{42}$$
$$= \{\tilde{x}_i \mid \exists \tilde{y}_i, \; Q\tilde{x}_i + R\tilde{y}_i \leq \lambda_i p_i\} \tag{43}$$

Using this form of $\lambda_i P_i$ in the right side of equation (41) yields the form in equation (40). ■

So we can represent the invariant set $\tilde{\Omega}_\tau$ as the projection of a polytope. Hence we can test if a given point is an element of $\tilde{\Omega}_\tau$ by solving a linear program. In the absence of switching, the size of the program is polynomial. Also using the arguments from section III-A, we see $\tilde{\Omega}_\tau$ is invariant for systems with only state measurements. This result can be viewed as a generalization of Theorem 3 in [9] to switched affine systems with additive disturbances. This type of implicit representation is useful for certain applications where it is only necessary to test if a point is in the invariant set, such as in supervisory control. We can recover an explicit representation of $\tilde{\Omega}_\tau$ as a polytope in halfspace form by projection, but this may be intractable due to the large number of constraints defining $\tilde{\Omega}_\tau$. Less conservative invariant sets can be achieved by selecting a larger horizon τ.

VI. CASE STUDIES

In this section we demonstrate our approach to compute invariant sets for two system models. The YALMIP interface [18] and the Gurobi solver [19] were used to solve the linear programs. The MPT3 library was used for polytopic computations [20]. All computations were performed on an Intel i7-8550 CPU laptop with 16GB of RAM. The code is available at `https://github.com/arw12625/padf`.

A. UAV consensus

In this example, we consider the dynamics of a group of d unmanned aerial vehicles (UAV) that employ an altitude consensus protocol. The UAVs attempt reach the same altitude by using the consensus protocol from [21] In this protocol, each UAV uses linear feedback on the difference between its and its neighbors' altitudes and velocities. These neighborhoods are determined by the UAVs' communication topology. We consider two cases where the topology is fixed and where the topology is subject to unknown dynamics, switching between the two topologies in Figure 2. Each of the UAVs are modeled by double integrator dynamics. For each UAV $i \in [d]$, we denote its altitude as h_i and its vertical velocity as v_i. Each UAV controls its velocity with an input $\hat{u}_i$ that is subject to a disturbance w_i so that its open-loop dynamics are described by

$$\begin{bmatrix} h_i(t+1) \\ v_i(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} h_i(t) \\ v_i(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \hat{u}_i(t) + w_i(t) \tag{44}$$

The communication topology of the system at time t is described by a graph with adjacency matrix $A_{adj}(t)$. The consensus protocol for each UAV is described by

$$\hat{u}_i(t) = u(t) + \sum_{j=1}^{d} \frac{A_{adj}(t)_{ij} K_f}{1 + d_i(t))} \left(\begin{bmatrix} h_j(t) \\ v_j(t) \end{bmatrix} - \begin{bmatrix} h_i(t) \\ v_i(t) \end{bmatrix} \right)\,, \tag{45}$$

where $K_f = \begin{bmatrix} 1 & 1 \end{bmatrix}$ is a fixed gain, $d_i(t)$ denotes the degree of node i in $A_{adj}(t)$, and $u(t)$ is a group reference velocity input. Modeling the dynamic communication topology as exogenous switching and this linear feedback protocol, the closed-loop system is a switched linear system with the single reference input u to be controlled. The absolute altitudes and velocities of each UAV are constrained to lie within a certain range $h_i(t) \in [-M_h, M_h]$ and $v_i(t) \in [-M_v, M_v]$.

First we consider the problem of computing an invariant set for the system with fixed grid topologies with $N =$

UAV Grid topology	1x1	2x2	3x3	3x4	2x7
Dimension (n)	2	8	18	24	28
Inside-Out Time	0.86	10.0	548	3415	10446
Membership test Time	1.4	6.0	51	342	671

TABLE I

COMPUTATION TIMES IN SECONDS FOR THE INSIDE-OUT ALGORITHM WITH $N = 10$ APPLIED TO THE UAV SYSTEM WITH A FIXED GRID COMMUNICATION TOPOLOGY.

$\tau = 10$. Recurrent sets were constructed using the inside-out algorithm with a parameterization Q constraining the absolute and relative altitudes and velocities of the UAVs. Then implicit descriptions of invariant sets were constructed. The applicability of these implicit descriptions are demonstrated by recording the time taken to test if randomly generated points for membership in the invariant set. These times are reported in Table I.

It is difficult to evaluate the conservativity of the invariant sets computed with this method as they are implicitly described in high dimensions. We can approximately evaluate these sets by sampling if points of the maximal invariant set are contained in the computed set. In the system with the 2x2 grid topology, it is possible to compute the maximal invariant set using set iteration methods. The test points were sampled from the vertices of the maximal invariant with a moderate tolerance. All points sampled were found to be contained in the invariant set $\tilde{\Omega}_\tau$ with $\tau = 20$ where Ω was constructed with the inside-out method with $N = 10$. So for long horizons, this method can approximate the maximal invariant set well.

Next we consider a system with 6 UAVs corresponding to a 12-dimensional system, switching between the two topologies depicted in Figure 2. We select horizons $N = \tau = 8$ and apply the inside-out algorithm with the parameterization from before to determine an N-step recurrent set. This procedure was performed in 560 seconds. The heights of the UAVs over several runs of the system under the recurrent controller are plotted in Figure 1. Using this set, an implicit description of an invariant set was constructed. Testing if a point in the invariant set was a member with the linear program was computed in an average of 1.93 seconds.

B. Large-scale TCL control

In this example, we consider large-scale coordination of thermostatically controlled loads (TCL). This includes appliances like air conditioners and water heaters that regulate temperatures within a desired range. Coordinating TCLs to provide a desired demand response while limiting power drawn from the distribution network is difficult due to the high dimension of the system.

Counting abstractions as in [22] have been used to reduce the dimension of this problem. These abstractions approximate the original system with a discrete transition graph where nodes represent a certain TCL mode and temperature range. A new system is constructed whose states are the counts of TCLs present in each node and control inputs represent the number of TCLs that should switch modes. This system's dynamics are linear, and furthermore counting constraints, like limiting the number of loads that are active, can be expressed as polytopes. However as states and inputs correspond to counts, this system has integral constraints. Discrete cycles in the transition graph of the abstracted system can be composed to construct closed trajectories of the linear system. Using this observation, [22] construct controllers to guarantee safety. However, these controllers are inherently open-loop and cannot account for uncertainty or disturbances in the TCL system.

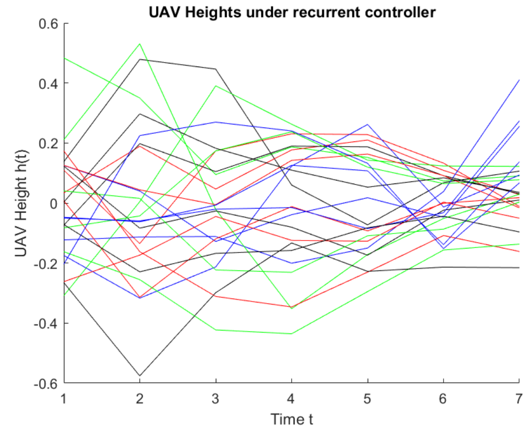

Fig. 1. The heights of the UAVs over several runs with the recurrent controller found for the system with topologies in Figure 2.

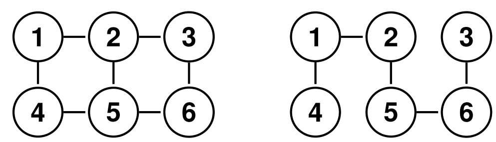

Fig. 2. The dynamic topologies used in the model of UAV consensus.

Invariant sets for TCL systems could be used to develop supervisors for tracking controllers with feedback to address this. However, while counting abstractions shift the complexity of the control problem from the number of TCLs, which can be thousands, to the size of abstraction graph, the resulting invariant set computation problem can be prohibitively large. Here we show that our method provides a preliminary solution to compute safe controllers for the linear TCL system. Although the computed set is implicit, requiring a solving a linear program to test membership, it is still useful in the construction of tracking controllers.

To address the integral constraints of the linear system, we consider a continuous relaxation. We can then apply the methods presented in this paper to construct invariant sets using recurrent sets for the relaxed system. By introducing a disturbance to model rounding of the control input to the nearest integral point, the integral points of an invariant set for the relaxed system will form an invariant set for the system with integral constraints. In order to construct recurrent sets for the relaxed system, we observe that the cycles considered in [22] form recurrent sets for the linear system without disturbances. To account for disturbances, we used enlarged versions of these sets as candidate recurrent sets.

We consider a counting abstraction system similar to the application example of [22] where the abstraction graph has 24 nodes. After pruning unused states, the resulting linear system has $n = 19$ states and $m = 16$ inputs. We consider polytopic constraints that bound the number of loads that are on, bound the number of loads switching from a given node by the number of loads in that mode, and ensure all states and inputs are nonnegative. The disturbance set modelling rounding is given by $W = [-0.5, 0.5]^m$. From the abstracted system, we can compute a set C of points forming a closed trajectories of the linear system as in [22]. We construct a candidate recurrent set as $\Omega = \text{conv}(C) \oplus \alpha W$ for some $\alpha \geq 1$. For the choice of $N = \tau = 8$ and $\alpha = 2$, the linear program to verify Ω is N-step recurrent was solved in 47.0 seconds. Next the implicit representation for the backwards reachable set $\tilde{\Omega}_\tau$ was computed. The linear program to test if a point is an element of the invariant set $\tilde{\Omega}_\tau$ was solved in an average of 55.3 seconds.

Remark 6.1: It is important to note that for both the high-dimensional UAV examples and the TCL example, the common backwards reachable set iteration to compute the maximal invariant set is intractable, requiring projections of polytopes in high dimensions. Additionally, a more recent method like [23] is not applicable as the systems are not controllable (even when restricted to non-switched version). Also due to the presence of disturbances and/or switching, the methods from [10], [9] cannot be used.

VII. CONCLUSION

We presented a linear programming based method to compute N-Step backward reachable sets via affine feedback for switched affine systems subject to polytopic constraints and disturbances. Given a recurrent sets, we showed how to construct implicit representations of invariant sets as a backwards reachable set. We also provided several constructions for recurrent sets. While the invariant sets are implicit, membership in the set can be efficiently tested with a linear program. The scalability of this approach is demonstrated with applications to physical system models. Future work includes improved methods for constructing recurrent sets as this remains a difficult and important problem for broader applicability of the approach. On the application side, we plan to interface the computed invariant sets with tracking controllers to build safety supervisors for hierarchical TCL control.

REFERENCES

[1] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[2] E. Kerrigan and J. Maciejowski, "Invariant sets for constrained non-linear discrete-time systems with application to feasibility in model predictive control," in *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*, vol. 5. Sydney, NSW, Australia: IEEE, 2000, pp. 4951–4956.

[3] M. Rungger and P. Tabuada, "Computing Robust Controlled Invariant Sets of Linear Systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3665–3670, Jul. 2017.

[4] E. D. Santis, M. D. D. Benedetto, and L. Berardi, "Computation of maximal safe sets for switching systems," *IEEE Transactions on Automatic Control*, vol. 49, no. 2, pp. 184–195, Feb. 2004.

[5] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[6] B. Legat, P. Tabuada, and R. M. Jungers, "Computing controlled invariant sets for hybrid systems with applications to model-predictive control," *arXiv:1802.04522 [math]*, Feb. 2018.

[7] D. Bertsekas, "Infinite time reachability of state-space regions by using feedback control," *IEEE Transactions on Automatic Control*, vol. 17, no. 5, pp. 604–613, 1972.

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

[9] M. Fiacchini and M. Alamir, "Computing control invariant sets in high dimension is easy," *arXiv:1810.10372 [cs]*, Oct. 2018.

[10] ——, "Computing control invariant sets is easy," *arXiv:1708.04797 [cs]*, Aug. 2017.

[11] S. Sadraddini and R. Tedrake, "Linear Encodings for Polytope Containment Problems," *arXiv:1903.05214 [cs, math]*, Mar. 2019.

[12] S. Keerthi and E. Gilbert, "Computation of minimum-time feedback control laws for discrete-time systems with state-control constraints," *IEEE Transactions on Automatic Control*, vol. 32, no. 5, pp. 432–435, May 1987.

[13] F. Blanchini, "Minimum-time control for uncertain discrete-time linear systems," in *[1992] Proceedings of the 31st IEEE Conference on Decision and Control*, Dec. 1992, pp. 2629–2634 vol.3.

[14] H. R. Tiwary, "On the Hardness of Computing Intersection, Union and Minkowski Sum of Polytopes," *Discrete & Computational Geometry*, vol. 40, no. 3, pp. 469–479, Oct. 2008.

[15] P. J. Goulart, E. C. Kerrigan, and J. M. Maciejowski, *Optimization over State Feedback Policies for Robust Control with Constraints*, 2005.

[16] J. Anderson, J. C. Doyle, S. Low, and N. Matni, "System Level Synthesis," *arXiv:1904.01634 [cs, math]*, Apr. 2019.

[17] Y. Chen, H. Peng, J. Grizzle, and N. Ozay, "Data-Driven Computation of Minimal Robust Control Invariant Set," in *2018 IEEE Conference on Decision and Control (CDC)*. Miami Beach, FL: IEEE, Dec. 2018, pp. 4052–4058.

[18] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in matlab," in *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

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

[20] 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.

[21] W. Ren, "On Consensus Algorithms for Double-Integrator Dynamics," *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1503–1509, Jul. 2008.

[22] P. Nilsson and N. Ozay, "Control synthesis for large collections of systems with mode-counting constraints," in *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control. HSCC'16*. ACM Press, 2016. [Online]. Available: https://doi.org/10.1145/2883817.2883831

[23] T. Anevlavis and P. Tabuada, "Computing controlled invariant sets in two moves," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 6248–6254.

[24] J. Skaf and S. P. Boyd, "Design of Affine Controllers via Convex Optimization," *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2476–2487, Nov. 2010.