

Computing controlled invariant sets for hybrid systems with applications to model-predictive control

Benoît Legat * Paulo Tabuada ** Raphaël M. Jungers ***

* ICTEAM, Université catholique de Louvain, 4 Av. G. Lemaître, 1348 Louvain-la-Neuve, Belgium (e-mail: benoit.legat@student.uclouvain.be)

** Department of Electrical and Computer Engineering, UCLA, (e-mail: tabuada@ee.ucla.edu)

*** ICTEAM, Université catholique de Louvain, 4 Av. G. Lemaître, 1348 Louvain-la-Neuve, Belgium (e-mail: raphael.jungers@uclouvain.be)

Abstract In this paper, we develop a method for computing controlled invariant sets using Semidefinite Programming. We apply our method to the controller design problem for switching affine systems with polytopic safe sets. The task is reduced to a semidefinite programming problem by enforcing an invariance relation in the dual space of the geometric problem. The paper ends with an application to safety critical model predictive control.

Keywords: Controller Synthesis; Set Invariance; LMIs; Scalable Methods.

1. INTRODUCTION

The problem of computing a controlled invariant set is a paradigmatic challenge in the broad field of Hybrid Systems control. Indeed, it is for instance crucial in safety-critical applications, such as the control of a platoon of vehicles or air traffic management; see Tomlin et al. (1998), where firm guarantees are needed on our ability to maintain the state in a safe region (e.g., with a certain minimal distance between vehicles). In other situations, the dynamical system might be too complicated to analyze exactly in every point of the state space, but yet it can be possible to confine the state within a guaranteed set. Such situations occur frequently in hybrid, embedded, event-triggered systems, because of the complexity of the dynamics.

A set is *controlled invariant* (sometimes also referred to as *viable*) if, any trajectory whose initial point is in the set can be kept inside it by means of a proper control action. Given a system with constraint specifications on the states and/or input, the controlled invariant set can be used to determine initial states such that trajectories with these initial conditions are guaranteed to meet the specifications. Moreover, in some situations, a state feedback control law can be derived from the knowledge of the controlled invariant set; see Blanchini (1999) for a survey.

The computation of invariant sets is usually achieved using either polyhedral computations or semidefinite programming. Polyhedral computations are typically restricted to affine constraint specifications but it has been recently shown that it can also be applied to algebraic constraints; see Athanasopoulos and Jungers (2016). If the system contains a control input, the computational complexity

of the problem becomes even more challenging. Indeed, this requires (see e.g., the procedure p. 201 in Blanchini and Miani (2015)) the computation of projections of polytopes when using polyhedral computations and semidefinite programming techniques are not directly applicable. Parametrization of the set have been proposed to improve the scalability of the polyhedral approach; see Rakovic and Baric (2010).

Methods based on polyhedral computations for hybrid control systems have been developed in Rungger et al. (2013); Smith et al. (2016); Rungger and Tabuada (2017). Unfortunately, the problem of polyhedral projection is well known to severely suffer from the curse of dimensionality, see Avis et al. (1995), and the additional complexity of the discrete dynamics in hybrid systems makes the problem even less scalable for these systems.

The semidefinite programming approach sacrifices exactness of the solution for the sake of algorithmic tractability. In the case of an uncontrolled system $x_{k+1} = Ax_k$, it consists in searching for an ellipsoidal set

$$\mathcal{E}_P = \{x \in \mathbb{R}^n \mid x^\top P x \leq 1\}$$

such that if $x^\top P x \leq 1$ then $x^\top A^\top P A x \leq 1$. Indeed, one can verify that it implies invariance of the set \mathcal{E}_P . The S-procedure allows to formulate the search of P as a semidefinite program; see Pólik and Terlaky (2007) for a survey on the S-procedure.

With the presence of the control u in the system $x_{k+1} = Ax_k + Bu_k$, the condition becomes:

$$x^\top P x \leq 1 \Rightarrow \exists u, (Ax + Bu)^\top P (Ax + Bu) \leq 1.$$

The control term u , or more precisely the existential quantifier \exists prevents the S-procedure to be directly applied.

Kurzanski and Varaiya (2005) show how to compute an over- and under-approximation of the reachable sets of a hybrid control system. While they approximate *reachable sets* and do not compute *controlled invariant sets*, their approach bears similarities with the method presented in this paper. However, their technique does not rely on semidefinite programming as they propagate ellipsoidal sets and do not need to enforce any invariance property.

In Korda et al. (2014), a semidefinite programming method is proposed for the computation of an outer approximation of the maximal controlled invariant sets. While the set computed with this method can be a good approximation of the maximal controlled invariant set, it is an outer approximation and is not controlled invariant unless the approximation is exact.

In this paper, we give a general method that circumvents this issue. A key ingredient in our technique is that we work in the dual space of the geometric problem. We detail the application of the method to two classes of hybrid systems: Discrete-Time Affine Hybrid Control System (HCS for short) and Discrete-Time Affine Hybrid Algebraic System (HAS for short). HAS are not control systems but the computation of invariant sets for such systems presents the same features than for HCS. As a matter of fact, we show how to reduce the computation of controlled invariant sets for HCS to the computation of invariant sets for HAS.

In this paper we break the problem into four subproblems, which we solve separately. In Section 2.2, we show how to reduce the computation of controlled invariant sets of a HCS with *constrained* input to controlled invariant sets of a HCS with *unconstrained* input. Then in Section 2.3, we give the reduction of the computation of controlled invariant sets of a HCS with unconstrained input to invariant sets of a HAS. In Section 3.1, we detail the relation between the algebraic invariance condition of HAS on a convex set and its polar set and we discuss how to lift the state space to handle non-homogeneity. In Section 3.2, we show that using the results of Section 3.1, the invariance of ellipsoids for a HAS can be formulated as a semidefinite program.

We end the paper with an application of the ellipsoidal controlled invariant sets to safety critical model predictive control. We show that precomputing such sets allows to guarantee safety of the model predictive controller and thus to alleviate expensive long-horizon computations thereby removing the need for long horizon.

2. CONTROLLED INVARIANT SET

In this section, we define HCS and HAS and give the invariance conditions for these two classes of hybrid systems. We detail the relation between controlled invariant sets of HCS and invariant sets of HAS.

2.1 Discrete-Time Affine Hybrid Control System

We will consider the following definition of Discrete-Time Affine Hybrid Control System.

Definition 1. A *Discrete-Time Affine Hybrid Control System (HCS)* is a system $S = (T, (A_\sigma, B_\sigma, c_\sigma)_{\sigma \in \Sigma}, (\mathcal{P}_q, U_q)_{q \in V})$

where $T = (V, \Sigma, \rightarrow)$, $\rightarrow \subseteq V \times \Sigma \times V$ and \mathcal{P}_q is the *safe set* corresponding to node q . A trajectory is a sequence $\{(x_k, u_k, \sigma_k)\}_{k \in \mathbb{N}}$ satisfying for all $k \in \mathbb{N}$:

$$\begin{aligned} x_{k+1} &= A_{\sigma_k} x_k + B_{\sigma_k} u_k + c_{\sigma_k}, \\ x_k &\in \mathcal{P}_{q_k}, u_k \in \mathcal{U}_{q_k}, q_k \xrightarrow{\sigma_k} q_{k+1}. \end{aligned}$$

Given a node $q \in V$, we denote the set of allowed switching signals as Σ_q , the state dimension as $n_{q,x}$ and the input dimension as $n_{q,u}$.

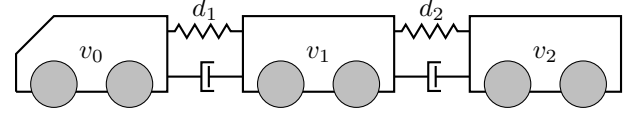


Figure 1. Illustration for Example 2 with two trailers.

We illustrate this definition with the cruise control example of Rungger et al. (2013).

Example 2. We consider a truck with M trailers as represented by Figure 1. There is a truck with speed v_0 followed by multiple trailers. The speed of the i th trailer is denoted v_i . There is a mass-spring-damper system between the truck and the first trailer (resp. the $(i-1)$ th trailer and the i th trailer). The scalar input u controls the speed v_0 of the truck. The spring elongation should always remain between -0.5 m and 0.5 m and the speeds of the truck and trailers should remain between 5 m s⁻¹ and 35 m s⁻¹. Moreover, there are three speed limits $\bar{v}_a = 15.6$ m s⁻¹, $\bar{v}_b = 24.5$ m s⁻¹, $\bar{v}_c = 29.5$ m s⁻¹ and whenever the truck is informed of a new speed limit, it has 0.8 s to decrease v_i ($0 \leq i \leq M$) below the speed limit.

We sample time with a period of 0.4 s and define an initial node q_{d0} and 6 nodes q_{ij} where $i \in \{a, b, c\}$ is the current speed limitation and $j \in \{0, 1\}$ is the number of sampling times left to satisfy the limit. The transitions are $q_{ij} \xrightarrow{\sigma} q_{\sigma 1}$ for each $i \in \{a, b, c, d\}$ and $\sigma \in \{a, b, c, d\} \setminus \{i\}$. The symbol a (resp. b , c) represents that the truck sees a new speed limitation \bar{v}_a (resp. \bar{v}_b , \bar{v}_c) and d represents that it does not see any new speed limitation. We suppose for simplicity that it is not possible to see a new speed limitation \bar{v}_σ from a node $q_{\sigma j}$. The possible transitions are represented in Figure 2.

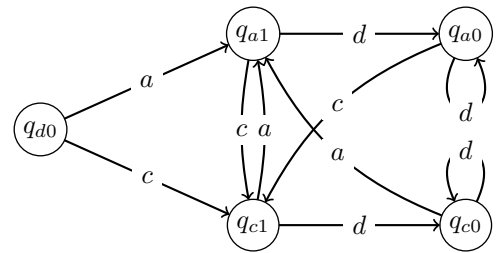


Figure 2. Transitions and switchings between the nodes for Example 2. Nodes q_{b1} and q_{b0} are not shown for clarity.

The reset maps $(A_\sigma, B_\sigma, c_\sigma)$ are simply the integration of the dynamical system over 0.4 s with a zero-order hold input extrapolation.

Let

$$\begin{aligned} P_0 &= \{ (d, v) \in \mathbb{R}^{2M+1} \mid -0.5 \leq d \leq 0.5, 5 \leq v \leq 35 \}, \\ P_i &= \{ (d, v) \in \mathbb{R}^{2M+1} \mid v \leq \bar{v}_i \}, \quad i = a, b, c, \end{aligned}$$

where $d = (d_1, \dots, d_M)$, $v = (v_0, \dots, v_M)$ and inequalities in the two equations above are entrywise. The safe sets are $\mathcal{P}_{q_{a0}} = P_0$ and for $i = a, b, c$, $\mathcal{P}_{q_{ij}} = P_0$ if $j > 0$ and $\mathcal{P}_{q_{i0}} = P_0 \cap P_i$. The input set is $\mathcal{U}_{ij} = \{u \in \mathbb{R} \mid -4 \leq u \leq 4\}$ for each node q_{ij} .

Definition 3. (Controlled invariant sets for a HCS). Consider a HCS S . We say that sets $\mathcal{C} = (\mathcal{C}_q)_{q \in V}$ are *controlled invariant* for S if $\mathcal{C}_q \subseteq \mathcal{P}_q$ for each $q \in V$ and $\forall q \rightarrow_\sigma q', x \in \mathcal{C}_q, \exists u \in \mathcal{U}_q$ such that

$$A_\sigma x + B_\sigma u + c_\sigma \in \mathcal{C}_{q'}.$$

Remark 4. It is important to distinguish two types of switching: *autonomous switching* and *controlled switching*; see details in (Liberzon, 2012, Section 1.1.3). Definition 3 is the definition of controlled invariance for autonomous systems and in this paper we only consider systems that switch autonomously. With controlled switching, “ $\forall q \rightarrow_\sigma q'$ ” is replaced by “ $\exists q \rightarrow_\sigma q'$ ” in Definition 3.

2.2 Handling controller constraints

We say that the input of a HCS is *unconstrained* if $\mathcal{U}_q = \mathbb{R}^{n_{q,u}}$ for all $q \in V$, otherwise we say that the input is *constrained*. The computation of controlled invariant sets for a HCS with constrained input can be reduced to the computation of invariant sets for a HCS with unconstrained input as shown by the following lemma.

Lemma 5. The sets $\mathcal{C} = (\mathcal{C}_q)_{q \in V}$ are *controlled invariant* for $S = (T, (A_\sigma, B_\sigma, c_\sigma)_{\sigma \in \Sigma}, (\mathcal{P}_q, \mathcal{U}_q)_{q \in V})$ if and only if there exist controlled invariant sets $\mathcal{C}' = (\mathcal{C}'_q)_{q \in V'}$ such that $\mathcal{C}'_q = \mathcal{C}_q \forall q \in V$ for the system $S' = (T', (A_\sigma, B_\sigma, c_\sigma)_{\sigma \in \Sigma'}, (\mathcal{P}'_q, \mathcal{U}'_q)_{q \in V'})$ where the new transitions $T' = (V', \Sigma', \rightarrow')$ are obtained as follows: For each transition $q \rightarrow_\sigma w$ in T , we create a node q^σ and the transitions $q \rightarrow'_{q^0} q^\sigma$ and $q^\sigma \rightarrow'_{\sigma'} w$ in T' . The set of nodes V' is the union of V with the set of nodes q^σ created.

The new safe and input sets are

$$\begin{aligned} \mathcal{P}'_q &= \mathcal{P}_q & \mathcal{U}'_q &= \mathbb{R}^{n_{q,u}} \\ \mathcal{P}'_{q^\sigma} &= \mathcal{P}_q \times \mathcal{U}_q & \mathcal{U}'_{q^\sigma} &= \mathbb{R}^0 \end{aligned}$$

and the new reset maps are

$$\begin{aligned} A_{q^0} &= \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} & B_{q^0} &= \begin{bmatrix} 0 \\ I \end{bmatrix} & c_{q^0} &= 0 \\ A_{\sigma'} &= [A_\sigma \ B_\sigma] & c_{\sigma'} &= c_\sigma \end{aligned}$$

and $B_{\sigma'}$ is the unique map sending $0 \in \mathbb{R}^0$ to $0 \in \mathbb{R}^{n_w}$.

Remark 6. If for a given q , Σ_q is a singleton $\{\sigma\}$, we can merge q and q^σ into one state hence have $\mathcal{P}'_q = \mathcal{P}_q \times \mathcal{U}_q$. In that case, \mathcal{C}_q will be the projection of \mathcal{C}'_q in its state space. Even if Σ_q is not a singleton, we can pick a single $\sigma \in \Sigma_q$ and merge q and q^σ into one state and use the reset map

$$A_{q^0} = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \quad B_{q^0} = \begin{bmatrix} 0 \\ I \end{bmatrix} \quad c_{q^0} = 0$$

so that switchings $\sigma' \in \Sigma_q \setminus \{\sigma\}$ ignore the part of the state of q that corresponds to the input to be used for σ .

Example 7. We represent on Figure 3 the application of the transformation described in Lemma 5 to the system of Example 2. We can use Remark 6 to avoid creating q^d for each q . Moreover, since $(A_\sigma, B_\sigma, c_\sigma)$ does not depend on σ , we can merge all the nodes q^a (resp. q^b , q^c) together into a common state that we name q_{a2} (resp. q_{b2} , q_{c2}).

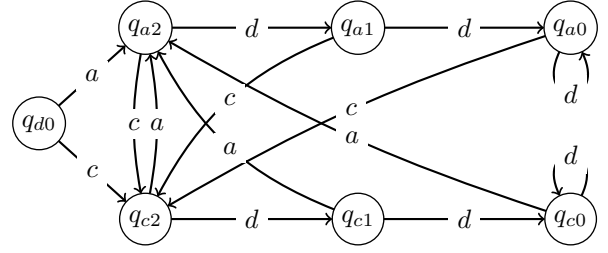


Figure 3. Transitions and switchings between the nodes for Example 7. Nodes q_{b2} , q_{b1} and q_{b0} are not shown for clarity.

2.3 Discrete-Time Affine Hybrid Algebraic System

Definition 8. A *Discrete-Time Affine Hybrid Algebraic System (HAS)* is a system $S = (T, (A_\sigma, E_\sigma, c_\sigma)_{\sigma \in \Sigma}, (\mathcal{P}_q)_{q \in V})$ where $T = (V, \Sigma, \rightarrow)$ and $\rightarrow \subseteq V \times \Sigma \times V$. A trajectory is a sequence $\{(x_k, \sigma_k)\}_{k \in \mathbb{N}}$ satisfying for all $k \in \mathbb{N}$:

$$\begin{aligned} E_{\sigma_k} x_{k+1} &= A_{\sigma_k} x_k + c_{\sigma_k}, \\ x_k &\in \mathcal{P}_{q_k}, q_k \rightarrow_{\sigma_k} q_{k+1}. \end{aligned}$$

Definition 9. (Invariant sets for a HAS). Consider a HAS S . We say that sets $\mathcal{C} = (\mathcal{C}_q)_{q \in V}$ are *invariant* for S if $\mathcal{C}_q \subseteq \mathcal{P}_q$ for each $q \in V$ and for all $q \rightarrow_\sigma q'$,

$$A_\sigma \mathcal{C}_q + c_\sigma \subseteq E_\sigma \mathcal{C}_{q'}. \quad (1)$$

Remark 10. Definition 9 can be interpreted as stating that \mathcal{C} is invariant if for each transition $q \rightarrow_\sigma q'$ and $x \in \mathcal{C}_q$,

$$\text{there exists } y \in \mathcal{C}_{q'} \text{ such that } A_\sigma x + c_\sigma = E_\sigma y.$$

A similar definition exists where this last part is replaced by

$$\text{for each } y \text{ such that } A_\sigma x + c_\sigma = E_\sigma y, y \text{ must belong to } \mathcal{C}_{q'}.$$

This is not equivalent to Definition 9 if A_σ and E_σ are not full rank. Moreover, computing ellipsoidal invariant sets according to this definition is much easier: it simply amounts to finding positive definite matrices Q_q such that $A_\sigma^\top Q_q A_\sigma \preceq E_\sigma^\top Q_{q'} E_\sigma$; see Owens and Debeljkovic (1985).

We now show that the computation of controlled invariant sets of a HCS can be reduced to the computation of invariant sets of a HAS.

Lemma 11. The sets $\mathcal{C} = (\mathcal{C}_q)_{q \in V}$ are *controlled invariant* for the HCS $S = (T, (A_\sigma, B_\sigma, c_\sigma)_{\sigma \in \Sigma}, (\mathcal{P}_q, \mathbb{R}^{n_{q,u}})_{q \in V})$ if and only if they are invariant sets for the HAS $S' = (T, (E_\sigma A_\sigma, E_\sigma, E_\sigma c_\sigma)_{\sigma \in \Sigma}, (\mathcal{P}_q)_{q \in V})$ where E_σ is a projection on $\text{Im}(B_\sigma)^\perp$.

3. COMPUTING CONTROLLED INVARIANT SETS

3.1 Duality correspondence for the invariance condition

Given a set \mathcal{C} and a linear map A , we define the following notations:

$$\begin{aligned} A\mathcal{C} &= \{Ax \mid x \in \mathcal{C}\} \\ A^{-1}\mathcal{C} &= \{x \mid Ax \in \mathcal{C}\} \\ A^{-\top}\mathcal{C} &= \{x \mid A^\top x \in \mathcal{C}\}. \end{aligned} \quad (2)$$

Note that A does not need to be invertible in these definitions.

Invariant sets can be computed numerically as *sublevel sets*¹ of polynomial functions using Sum-of-Squares. One property of sublevel sets that is usually used can be formulated as follows: If \mathcal{C} is the ℓ -sublevel set of a function f then for any function g , $g^{-1}(\mathcal{C})$ is the ℓ -sublevel set of the function $f \circ g$. Thanks to this property, computing a set \mathcal{C} satisfying $AC \subseteq \mathcal{C}$ for some linear map A can be for example achieved by searching for a set \mathcal{C} being the 1-sublevel set of a polynomial $p(x)$. Indeed, the invariance constraint is equivalent to $\mathcal{C} \subseteq A^{-1}\mathcal{C}$ which is equivalent to the following implication : for all x , $p(x) \leq 1 \Rightarrow p(Ax) \leq 1$. The latter proposition can be translated to a constraint of nonnegativity of a polynomial using the Sum-of-Squares formulation and the S-procedure.

For HAS, we have in (1) an invariance constraint of the form $AC \subseteq EC$ and we would like to find an equivalent form with a pre-image as we had with $\mathcal{C} \subseteq A^{-1}\mathcal{C}$. This can be achieved using the polar of the set \mathcal{C} thanks to the following lemma.

Lemma 12. ((Rockafellar, 2015, Corollary 16.3.2)). For any convex set \mathcal{C} (resp. convex cone \mathcal{K}) and linear map A ,

$$(AC)^\circ = A^{-\top}\mathcal{C}^\circ$$

$$(AK)^* = A^{-\top}\mathcal{K}^*$$

where \mathcal{C}° denotes the polar of the set \mathcal{C} and \mathcal{K}^* denotes the dual of the cone \mathcal{K} .

Lemma 12 shows that $AC \subseteq EC$ is equivalent to $A^{-\top}\mathcal{C}^\circ \supseteq E^{-\top}\mathcal{C}^\circ$. Since the invariant sets of the HAS may not have the origin in their interior, the polar transformation cannot be readily applied. We handle this non-homogeneity by taking the conic hull of the lifted sets $\mathcal{C} \times \{1\}$. More precisely, we define

$$\tau(\mathcal{C}) = \{(\lambda x, \lambda) \mid \lambda \geq 0, x \in \mathcal{C}\} \quad (3)$$

$$r(A, c) = \begin{bmatrix} A & c \\ 0 & 1 \end{bmatrix}. \quad (4)$$

It can be verified that for any set \mathcal{C} , vector c and linear map A ,

$$\tau(AC + c) = r(A, c)\tau(\mathcal{C}). \quad (5)$$

Moreover, for any half-space $a^\top x \leq \beta$,

$$a^\top x \leq \beta, \forall x \in \mathcal{C} \Leftrightarrow (-a, \beta) \in \tau(\mathcal{C})^*. \quad (6)$$

Theorem 13. Consider a HAS S . The closed convex sets $\mathcal{C} = (\mathcal{C}_q)_{q \in V}$ are *invariant* for S if and only if $\mathcal{C}_q \subseteq \mathcal{P}_q$ for each $q \in V$ and for all $q \rightarrow_\sigma q'$,

$$r(A_\sigma, c_\sigma)^{-\top} \tau(\mathcal{C}_q)^* \supseteq r(E_\sigma, 0)^{-\top} \tau(\mathcal{C}_{q'})^*. \quad (7)$$

3.2 Computation using ellipsoids

While Theorem 13 holds for any convex sets $(\mathcal{C}_q)_{q \in V}$, restricting our attention to ellipsoidal sets renders the invariance condition amenable to semidefinite programming. This section details the semidefinite program needed to find these ellipsoidal invariant sets and shows its exactness in Theorem 17.

We define the following notations for ellipsoids

$$\mathcal{E}_{Q,c} = \{x \mid (x - c)^\top Q(x - c) \leq 1\}$$

$$\mathcal{E}_{D,d,\delta} = \{x \mid x^\top D x + 2d^\top x + \delta \leq 0\}.$$

¹ The ℓ -sublevel set of a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the set $\{x \in \mathbb{R}^n \mid f(x) \leq \ell\}$.

We denote the set of symmetric matrices of \mathbb{R}^n as \mathcal{S}^n .

Lemma 14. Let $Q, D \in \mathcal{S}^n$, $c, d \in \mathbb{R}^n$, $\delta \in \mathbb{R}$ with $Q \succ 0$. We have $\mathcal{E}_{Q,c} = \mathcal{E}_{D,d,\delta}$ if and only if $D \succ 0$ and there exists $\lambda > 0$ such that

$$\lambda = d^\top D^{-1} d - \delta \quad (8)$$

$$c = -D^{-1} d \quad (9)$$

$$Q = D/\lambda. \quad (10)$$

We use the following corollary to represent the cones $\tau(\mathcal{C}_q)^*$ as the 0-sublevel set of quadratic forms $p(y) = p(x, z) = x^\top D_q x + 2d_q^\top x z + \delta_q z^2$.

Corollary 15. Let $\mathcal{K} = \{(x, z) \mid x^\top D x + 2d^\top x z + \delta z^2 \leq 0, z \geq 0\}$ be a cone that has a nonempty interior and no intersection with the hyperplane $\{(x, 0) \mid x \in \mathbb{R}^n\}$ except the origin. The cone \mathcal{K} is convex if and only if $D \succ 0$.

In Corollary 15, we require the cone to have no intersection with a hyperplane (except the origin). However, the cone $\tau(\mathcal{C}_q)^*$ has no intersection with the hyperplane $\{(x, 0) \mid x \in \mathbb{R}^n\}$ if and only if the origin is contained in \mathcal{C}_q which may not be the case. In order to alleviate this, the approach we suggest is to suppose that we know one point h_q in the interior of each \mathcal{C}_q and we use Corollary 15 in a transformed space where h_q is mapped to the z -axis $(0, 1)$. For this transformation we use the *Householder reflection* H_h (Golub and Van Loan, 2012, Section 5.1.2) which is symmetric and orthogonal.

The optimization problem to solve is represented in Program 16. The transformation of this program to a semidefinite program can be done automatically using the standard Sum-of-Square procedure; see Blekherman et al. (2012).

Program 16.

$$\begin{aligned} \max_{\substack{D_q \in \mathcal{S}^n, d_q \in \mathbb{R}^n, \\ \delta_q \in \mathbb{R}, \lambda_{q \rightarrow \sigma q'} \geq 0}} \sum_{q \in V} \log \det D_q \\ \begin{bmatrix} D_q & d_q \\ d_q^\top & \delta_q + 1 \end{bmatrix} \succ 0 \end{aligned} \quad (11)$$

$$p_q(y) = y^\top H_{h_q} \begin{bmatrix} D_q & d_q \\ d_q^\top & \delta_q \end{bmatrix} H_{h_q} y \quad (12)$$

$$p_q(r(A_\sigma, c_\sigma)^\top y) \leq \lambda_{q \rightarrow \sigma q'} p_{q'}(r(E_\sigma, 0)^\top y), \quad (13)$$

$$\forall q \in V, \forall q \rightarrow_\sigma q', \forall y \in \mathbb{R}^{n_q, x+1}$$

$$p_q(-a, \beta) \leq 0 \quad \forall q \in V, \forall a^\top x \leq \beta \text{ supporting } \mathcal{P}_q \quad (14)$$

$$p_q(0, 1) < 0 \quad \forall q \in V. \quad (15)$$

The constraint (11) ensures both convexity of $\tau(\mathcal{C}_q)^*$ and the fact that $\det D_q$ does not overestimate the volume of the ellipsoid transformed by the Householder reflection. The constraint (13) is the S-procedure applied to the condition (7). The constraint (14) uses (6) to ensure that \mathcal{C}_q is contained in \mathcal{P}_q . The constraint (15) ensures that $\tau(\mathcal{C}_q)^*$ has non-empty interior. Note that if \mathcal{P}_q has no unbounded subspace, (15) is not necessary since the non-empty interior condition will already be ensured by (14).

Theorem 17. Consider a HAS S and points $(h_q \in \mathcal{P}_q)_{q \in V}$. The polynomial $p_q(x, z)$ is feasible for Program 16 if and only if there exists invariant convex sets $\mathcal{C} = (\mathcal{C}_q)_{q \in V}$ such that $h_q \in \mathcal{C}_q$ for each $q \in V$ and $\tau(\mathcal{C}_q)^*$ is the 0-sublevel set of $p_q(x, z)$. Moreover, the optimal solution of Program 16

is the solution that minimizes the sum of the logarithm of the volume of the intersection of the each cone $\tau(\mathcal{C}_q)^*$ with the hyperplane $\{x \mid \langle h_q, x \rangle = 1\}$.

Example 18. We apply Program 16 to Example 7 with the same values for the parameters as the ones used in Rungger et al. (2013), that is, $m_0 = 500$ kg, $m = 1000$ kg, $k_d = 4600$ N s m⁻¹ and $k_s = 4500$ N kg⁻¹. The values used for h_q are the same for each node $q \in V$: $u = d_i = 0$ and $v_0 = v_i = (5 + v_a)/2$ for $i = 1, \dots, M$.

We vary the number of trailers M from 1 to 10. Figure 4 represents the controlled invariant set at node q_{a0} . As we can see, the constraints on the trailers are propagated to the truck and, as the number M increases, the truck speed and acceleration become more constrained.

The time taken by Mosek 8.1.0.34 (ApS (2017)) to solve the problem is given by Figure 5².

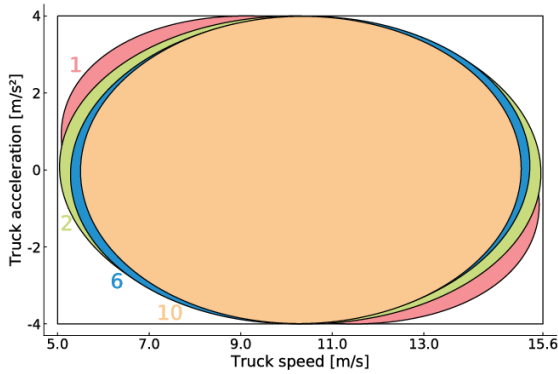


Figure 4. Projection onto the state v_0 and input u of the optimal solution of Program 16 for Example 18 at node q_{a0} for various numbers of trailers.

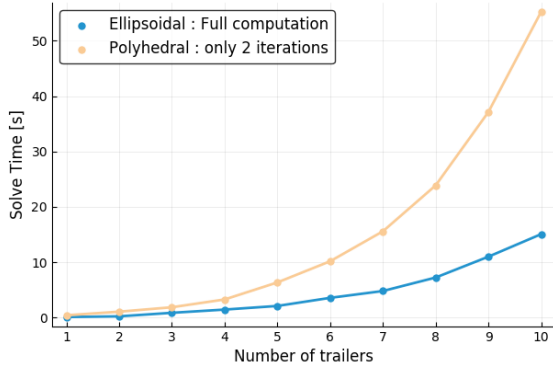


Figure 5. Computation time with Mosek 8.1.0.34 for Example 18 with various numbers of trailers compared to two iterations of the polyhedral approach (see e.g., the procedure p. 201 in Blanchini and Miani (2015)) implemented with the CDD library Fukuda (1999). Note that after two iterations, the polyhedral sets obtained are not controlled invariant. One needs to wait for the convergence of the algorithm to obtain a controlled invariant set. Moreover, iterations are usually increasingly slower as the number of facets of the polyhedral sets increases with the iterations.

² We set $\lambda_{q \rightarrow \sigma q'}$ to 1 for each transition $q \rightarrow_{\sigma} q'$ to make the problem convex.

4. APPLICATION TO MODEL PREDICTIVE CONTROL

As mentioned in the introduction, the controlled invariant sets can be used to derive a feedback control law. We illustrate this with a Model Predictive Control (MPC) numerical experiment. We consider a truck with one trailer ($M = 1$) as in Example 18. The truck starts with speeds $v_0 = v_1 = 10$ m s⁻¹ and spring displacement $d = 0$ m and has as objective to maximize the distance covered in 60 s. The maximal speed is initially 35 m s⁻¹ but after 30 s, it drops to $v_a = 15.6$ m s⁻¹.

In a classical MPC controller, the truck acceleration u is controlled by solving a constrained optimal control problem up to horizon H . We observe that if $H \leq 9.2$ s, the controller is at some point unable to find values of u satisfying input constraints such that the state remains in the safe set.

For safety-critical applications, this lack of guarantee is not acceptable as it is necessary to be certain that the system can remain in the safe set. Moreover, in a real-time context, the need to pick a large horizon is problematic as it increases the cost of online computations. In our setting, we constrain the state to remain in the controlled invariant sets computed in Example 18³ and thereby solve both issues; safety is guaranteed for arbitrarily long simulations and the length of the horizon does not influence safety so smaller length can be used. Note that the controlled invariant sets can be computed offline so if it allows to reduce the horizon length, it enables online computational cost to be moved offline. Besides, constraining the state variables to belong to the ellipsoidal controlled invariant sets is straightforward⁴. The results of the experiment can be found in Figure 6 and Figure 7.

5. CONCLUSION

We have developed a methodology for computing controlled invariant sets of Discrete-Time Affine Hybrid Control System (HCS) and Discrete-Time Affine Hybrid Algebraic System (HAS) with *autonomous switching* (see Remark 4). This method can be combined with semidefinite programming in order to compute ellipsoidal controlled invariant sets. We have shown that our technique can be used as a building block in a model predictive control scheme. This allows, among other things, to reduce the online computational cost by precomputing controlled invariant sets.

We feel that we have only scratched the surface of the potential of the duality correspondence of Section 3.1. Many extensions of this work are possible such as hybrid systems with controlled switching, or the use of Sum-Of-Squares techniques in order to enrich the geometry of the possible invariant sets.

³ Example 18 corresponds to an MPC controller of horizon 0.8 s. An MPC controller of different horizon computes different controlled invariant sets by updating the hybrid system accordingly.

⁴ The membership to $\mathcal{E}_{Q,c}$ is second order cone representable. Indeed consider a Cholesky factorization $Q = L^T L$, the inequality $(x - c)^T Q (x - c) \leq 1$ can be rewritten as $\|L(x - c)\|_2 \leq 1$ where $\|\cdot\|_2$ is the Euclidean norm.

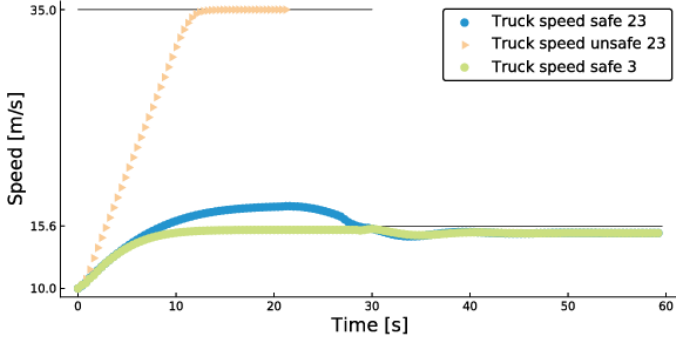


Figure 6. Evolution with time of the speed of the truck for various MPC strategies. In the legend, *safe* designates our MPC strategy using our computed invariant sets, while *unsafe* designates a classical MPC approach. The piecewise horizontal line represents the speed limitation at time t . One can see that the MPC approach with invariant sets allows to remain in the safe set even with an horizon of 3 time steps. Moreover, the unsafe controller can fail to find feasible values, as shown in Figure 7.

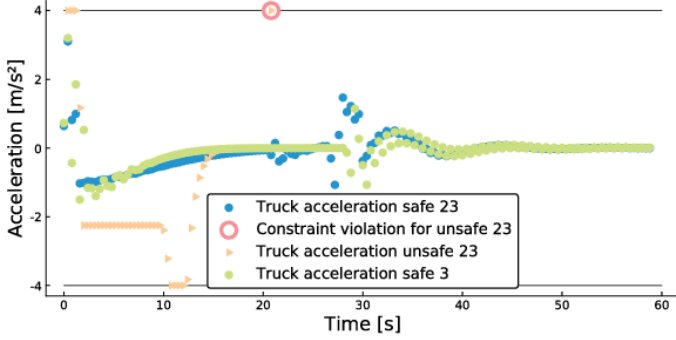


Figure 7. Acceleration of the truck in safe and unsafe mode. See Figure 6 for the legend syntax. We see (just after $t = 20$ s) that the unsafe controller requires a value $u > 4 \text{ m s}^{-2}$ in order to remain in the safe set. Moreover, we can see that the control is smoother in safe mode. Note that using a longer horizon H renders the control even smoother, see e.g., between 30 s and 40 s.

The reformulation of the computation of controlled invariant sets of hybrid control system to the computation of invariant sets of hybrid algebraic system with Lemma 5 and Lemma 11 allows to have a more behavioral invariance relation. In the future, we would like to put our result in the framework of behavioral theory in order to investigate how to further generalize them; see Willems and Polderman (2013).

REFERENCES

- ApS, M. (2017). Mosek optimization suite release 8.1.0.43. URL: <http://docs.mosek.com/8.1/intro.pdf>.
- Athanasopoulos, N. and Jungers, R.M. (2016). Computing the domain of attraction of switching systems subject to non-convex constraints. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control, HSCC '16*, 41–50. ACM, New York, NY, USA.
- Avis, D., Bremner, D., and Seidel, R. (1995). How good are convex hull algorithms? In *Proceedings of the eleventh annual symposium on Computational geometry*, 20–28. ACM.
- Blanchini, F. (1999). Set invariance in control. *Automatica*, 35(11), 1747–1767.
- Blanchini, F. and Miani, S. (2015). *Set-theoretic methods in control*. Springer, second edition.
- Blekherman, G., Parrilo, P., and Thomas, R. (2012). *Semidefinite Optimization and Convex Algebraic Geometry*. Society for Industrial and Applied Mathematics, Philadelphia, PA.
- Fukuda, K. (1999). cdd/cdd+ Reference Manual. *Institute for Operations Research, ETH-Zentrum*.
- Golub, G.H. and Van Loan, C.F. (2012). *Matrix computations*, volume 3. JHU Press.
- Korda, M., Henrion, D., and Jones, C.N. (2014). Convex computation of the maximum controlled invariant set for polynomial control systems. *SIAM Journal on Control and Optimization*, 52(5), 2944–2969.
- Kurzanski, A.B. and Varaiya, P. (2005). On verification of controlled hybrid dynamics through ellipsoidal techniques. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, 4682–4687. IEEE.
- Liberzon, D. (2012). *Switching in systems and control*. Springer Science & Business Media.
- Owens, D.H. and Debeljkovic, D.L. (1985). Consistency and liapunov stability of linear descriptor systems: A geometric analysis. *IMA Journal of Mathematical Control and Information*, 2(2), 139–151.
- Pólik, I. and Terlaky, T. (2007). A survey of the s-lemma. *SIAM review*, 49(3), 371–418.
- Rakovic, S.V. and Baric, M. (2010). Parameterized robust control invariant sets for linear systems: Theoretical advances and computational remarks. *IEEE Transactions on Automatic Control*, 55(7), 1599–1614.
- Rockafellar, R.T. (2015). *Convex analysis*. Princeton university press.
- Rungger, M., Mazo Jr, M., and Tabuada, P. (2013). Specification-guided controller synthesis for linear systems and safe linear-time temporal logic. In *Proceedings of the 16th international conference on Hybrid systems: computation and control*, 333–342. ACM.
- Rungger, M. and Tabuada, P. (2017). Computing robust controlled invariant sets of linear systems. *IEEE Transactions on Automatic Control*.
- Smith, S.W., Nilsson, P., and Ozay, N. (2016). Interdependence quantification for compositional control synthesis with an application in vehicle safety systems. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, 5700–5707. IEEE.
- Tomlin, C., Pappas, G.J., and Sastry, S. (1998). Conflict resolution for air traffic management: A study in multi-agent hybrid systems. *IEEE Transactions on automatic control*, 43(4), 509–521.
- Willems, J.C. and Polderman, J.W. (2013). *Introduction to mathematical systems theory: a behavioral approach*, volume 26. Springer Science & Business Media.