An Online Approach to Active Set Invariance

Thomas Gurriet[1], Mark Mote[2], Aaron D. Ames[1], and Eric Feron[2]

Abstract—**This paper presents an online approach to safety critical control. The common approach for enforcing safety of a system requires the offline computation of a viable set, which is either hard and time consuming or very restrictive in terms of operational freedom for the system. The first part of this work shows how one can constrain a system to stay within reach of an appropriately chosen backup set in a minimally invasive way by performing online sensitivity analysis around a backup trajectory. For linear systems, we show how to use an optimal backup strategy in the form of a Model Predictive Controller (MPC) to maximize the operational freedom of the system. The second part of this work shows how to leverage this capability and factor in state constraints to enforce set invariance only based on online computations of sensitivities. For linear systems, the optimal strategy is again considered and we show how one can perform the sensitivity analysis based on a measure of feasibility of a state constrained MPC. This approach is illustrated in simulation on a linear inverted pendulum.**

I. Introduction

Thanks to the ever-increasing performance of embedded computers and electromechanical systems, developing Cyber-Physical-Systems (CPS) is now more accessible than ever. Many people recognize the potential of CPS, especially in the context of smart cities, transportation, and manufacturing. Nonetheless, there has been a lack of translation of results form the CPS domain to real-world settings. This is in part due to their lack of reliability and safety [1]. Safety is a notoriously difficult problem for which solutions that are both rigorous and practical have yet to be realized. As a result, people in practice are either left having to isolate the systems from anything they could injure, or having to use heuristic algorithms whose effectiveness is hard if not impossible to evaluate. Rigorous and efficient solutions to this challenge will therefore most likely be the key in allowing autonomous CPS to accompany us in our everyday lives.

The concept of safety is fundamentally linked to the idea of constrained behavior. Given a chosen *safety set* (set of states of the system), a system being safe is commonly defined as this system never leaving the safety set. Safety is therefore usually associated with the topic of set invariance [2]. When a system is simple or enjoys some particular structure as in [3], analytical control strategies ensuring the system does not leave the safety set can be derived. But in general, it is very difficult to directly find such a control strategy. The main difficulty with staying inside an arbitrary safety set comes from the fact that in general, some subsets of the safety set cannot be visited by the system without eventually and inexorably leaving the safety set. However, if one is able to find a viable subset of the safety set [4], that is, a set such that for any initial condition inside this viable set it is possible to remain in this viable set—finding a *safe* control law becomes much simpler.

Provided one is able to compute a viable set, continuous filtering of the control input can then be performed so as to enforce set invariance in a minimally invasive way as proposed in [5]. The largest viable subset of the safety set is called the *viability kernel*. Finding the viability kernel grants maximum operational freedom to the system while ensuring the it can remain in the safety set. As a result, computing viability kernels has been the focus of of a wide variety of research over the years [6]. Approaches that have been proposed include: discretized solutions of Hamilton-Jacobi equations [7], SOS optimization [8], sampling [9] and many others [6]. Unfortunately, all these algorithms take a substantial amount of time to run and can only handle high dimensional systems at the expense of a very conservative results, leading to small operational regions and poor performances for the system.

The main contribution of this paper is a method to alleviate the need to compute a viable set *offline* by computing the local information needed to stay inside the safety set *online*. This method requires having a *backup strategy* as in [10], [11], but contrary to the aforementioned work, the backup strategy is never followed. Instead, a set of all the states *safely within reach* of an appropriate *backup set* is implicitly defined. This set is showed to be *larger* than the backup set and viable, which permits the use of Active Set Invariance Filter (ASIF) [12]. Because this set is only defined implicitly, the information necessary to the ASIF is numerically evaluated online using a forward sensitivity method [13]. This approach is extended, for linear systems, to the case where the backup strategy is *optimal* which ensures safety while providing maximal operational freedom for the system.

The rest of the paper is laid out as follows. In Sect. II, we present the mathematical background of the ASIF. In Sect. III, we present a method to use the ASIF of Sect. II to stay *within reach* of a backup set. In Sect. IV and V we show how this method can be implemented for linear and nonlinear systems respectively. In Sect. VI, we present a method to use the conditions derived in Sect. III and guarantee feasibility of the ASIF when used to enforce set invariance of an arbitrary safety set. In Sect. VI, we show how this can be implemented for linear systems. Conclusions are provided in Sect. VIII.

[1] Thomas Gurriet and Aaron Ames are with the Department of Mechanical Engineering, California Institute of Technology, Pasadena, CA, 91125 USA.

[2] Mark Mote and Eric Feron are with the Department of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA.

II. Background on Active Set Invariance

In this paper, we start by considering ideal continuous-time affine control system of the form:

$$\dot{x} = f(x) + g(x)u, \tag{1}$$

with f and g continuous functions defined on $\mathbb{R}^n$, and with $u \in U$ a compact set of $\mathbb{R}^m$. Note that the existence and uniqueness of solutions to (1) is required for most of the discussed results to hold. The Lipschitz continuity of f, g and u provides for example a sufficient condition in that regard.

As discussed in the introduction, the notion of safety is formalized into a set invariance requirement.

Definition 1. A closed set S is **forward invariant** for system (1) if $x(t_0) \in S \implies \forall t \geq t_0,\ x(t) \in S$.

The main tool at our disposal for set invariance is Nagumo's theorem [14]. Nagumo's theorem states that the forward invariance of S for system (1) is equivalent to the *sub-tangentiality condition*:

$$f(x) + g(x)u(x) \in \mathcal{T}_S(x), \tag{2}$$

being satisfied for all $x \in S$, where $\mathcal{T}_S(x)$ is the contingent cone to S at x [4], [14].

Depending on the type of set considered, there are different ways of expressing the contingent cone. Practical sets [14] are suitable for most realistic cases and makes it convenient to express the contingent cone. To describe such sets, one only[1] needs to consider r continuously differentiable functions $h_i : \mathbb{R}^n \to \mathbb{R}$ such that

$$\begin{aligned} S &= \{x \in \mathbb{R}^n \mid h_i(x) \geq 0,\ i \in \{1, \ldots, r\}\} \\ \partial S &= \{x \in \mathbb{R}^n \mid h_i(x) = 0,\ i \in \{1, \ldots, r\}\}. \end{aligned} \tag{3}$$

For such sets, the contingent cone can be expressed as

$$\mathcal{T}_S(x) = \{z \in \mathbb{R}^n \mid \forall i \in Act(x),\ \nabla h_i(x).z \geq 0\}, \tag{4}$$

with $Act(x) \triangleq \{i \in \{1, \ldots, r\} \mid h_i(x) = 0\}$. In that case, the sub-tangentiality condition (2) can be written as

$$TC_i(x, u) \triangleq L_f h_i(x) + L_g h_i(x) u(x) \geq 0, \tag{5}$$

for all $x \in \partial S$, and $i \in Act(x)$. Here, $L_f h$ and $L_g h$ denote the Lie derivatives of h along f and g respectively. Therefore, condition (5) defines for any $x \in S$ a set $U_S(x)$ of admissible inputs that guarantee forward invariance of S:

$$U_S(x) \triangleq \begin{cases} \{u \in \mathbb{R}^m \mid \forall i \in \{1, \ldots, r\}, TC_i(x, u) \geq 0\}, & \text{if } x \in \partial S \\ \mathbb{R}^m, & \text{otherwise} \end{cases}$$

The sub-tangentiality condition is however not very usable in practice as it only defines an non trivial set of admissible inputs when the system is on the boundary of the safety set, which is a surface in the state space, i.e. has no volume. The idea introduced in [5] is to consider a strengthening term in (5) and to impose this new *barrier condition*:

$$BC_i(x, u) \triangleq L_f h_i(x) + L_g h_i(x) u(x) + \alpha_i(h_i(x)) \geq 0, \tag{6}$$

for all $x \in S$, $i \in \{1, \ldots, r\}$ and with the *strengthening* extended class $\mathcal{K}$ functions $\alpha_i : \mathbb{R} \to \mathbb{R}$. This barrier condition defines a set $\widetilde{U_S}(x)$ of admissible inputs:

$$\widetilde{U_S}(x) \triangleq \{u \in \mathbb{R}^m \mid \forall i \in \{1, \ldots, r\}, BC_i(x, u) \geq 0\} \tag{7}$$

and because for all $x \in S$, $\widetilde{U_S}(x) \subseteq U_S(x)$, this new condition also implies forward invariance of S.

In [15], the authors propose a method to supplement any existing controller that would not be capable of ensuring set invariance on its own by continuously filtering this controller's inputs. This is done by solving a quadratic program in real-time (cf. Fig. 1 and (8)), minimizing the norm of the difference between the desired and actual inputs, therefore providing an *optimal* level of fidelity to any controller's desired inputs while guaranteeing safety of the system.

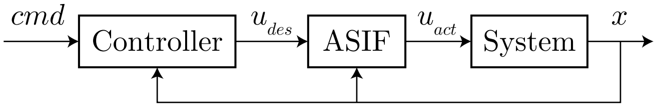

Fig. 1: Active Set Invariance control structure.

ASIF-QP

$$\begin{aligned} u_{\text{act}}(x) = \underset{u \in U}{\operatorname{argmin}} \quad & \|u_{\text{des}}(x) - u\|^2 \\ \text{s.t.} \quad & u \in \widetilde{U_S}(x) \end{aligned} \tag{8}$$

This is however only half of the story as nothing guarantees that $\widetilde{U_S}(x) \cap U$ is non empty for all $x \in S$. So there are no guarantees that (8) is always feasible. This is because for a given control system, arbitrary sets cannot *a priori* be rendered forward invariant. A set that can be rendered forward invariant is commonly referred to as **viable** set [4].

Definition 2. A closed set S is **viable** for system (1) if for all $x(t_0) \in S$, there exists a control law $u : \mathbb{R}^n \to U$ such that $\forall t \geq t_0$, $x(t) \in S$ under that policy.

Ideally, one would want to find the largest viable subset of S (the *viability kernel*) to maximize the operational freedom of the system. This is however notoriously hard—just as hard as finding an optimal control law [7]. But as in optimal control, there is a dual approach: continuously solving for the optimal control action at the current state. Unfortunately, solving viability this way requires finding a trajectory over an *infinite* time horizon, which is not possible in practice. In the rest of this paper, we show how it is possible to take a small viable subset of S (which is easy to compute) and, by considering a *finite time* trajectory, allow the system to evolve in a larger viable subset of S.

Note that because in most cases $\widetilde{U_S}(x) \subset U_S(x)$, finding a viable set is not necessarily sufficient to ensure that (8) is always feasible. One has to be careful and choose the strengthening functions α_i such that for all $x \in S$, $\widetilde{U_S}(x) \cap U \neq \emptyset$ as shown in [12].

[1] See [14] for conditions under which S is practical.

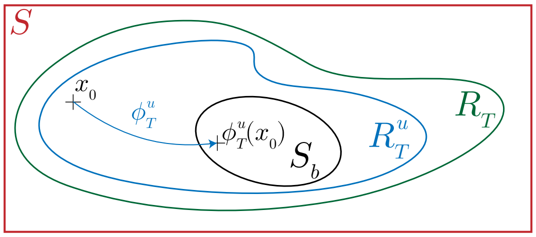

Fig. 2: Safety Set in red, Backup Set in black, and Region of Attraction in blue and Backward Reachable set in Green.

III. Online Reachability Theory

A. Arbitrary Backup Strategy

For the rest of the paper, we will call *control policy* inputs that parameterized by time and *control law* inputs that are parameterized by the state of the system. Let $\mathcal{U}$ be the set of all Lipschitz continuous *control laws* $u : \mathbb{R}^n \to U$. We will also denote by $\mathcal{U}_T$ the set of all piecewise continuous *control policies* $u : [0, T] \to U$. Let's assume that f and g are locally Lipschitz continuous over $\mathbb{R}^n$ and that for all control laws $u \in \mathcal{U}$ there exists a solution to (1) that is unique and defined for all time. Therefore, one can define $\phi^u : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ to be the global *flow* of (1) under the control law u. Let S be a closed *safety set* system (1) is required to stay in and $S_b \subset S$ be compact *backup set* as depicted in Fig. 2. We furthermore assume that S_b is practical and can be represented as the super level set of a smooth function $h_b : \mathbb{R}^n \to \mathbb{R}$. Finally, we define

$$R^u_T \triangleq \{x \in \mathbb{R}^n \mid \phi^u_T(x) \in S_b\}, \tag{9}$$

with $T \geq 0$ and $u \in \mathcal{U}$. Under all these assumptions, the map $\phi^u_t : \mathbb{R}^n \to \mathbb{R}^n$ defined by $\phi^u_t(x) \triangleq \phi^u(t, x)$ is an homeomorphism of $\mathbb{R}^n$ [16], hence $R^u_T = (\phi^u_T)^{-1}(S_b)$ and $\phi^u_T(R^u_T) = S_b$. The definition of forward invariance can therefore be reformulated in terms of the flow as $x(t_0) \in S \Rightarrow \forall t \geq t_0,\ x(t) \in S$ is equivalent to $\forall t \geq 0,\ \phi^u_t(S) \subseteq S$.

Our approach is motivated by the following observations.

Proposition 1. *Given $u \in \mathcal{U}$, if S_b is forward invariant under that control law, then for all $T \geq 0$, R^u_T is forward invariant.*

Proof. Let's reason by contraction and assume that R^u_T is not forward invariant. That means there exist $x^* \in R^u_T$ and $t^* \geq 0$ such that $\phi^u_{t^*}(x^*) \notin R^u_T$. Let $\tilde{x} \triangleq \phi^u_{t^*}(x^*)$. By property of the flow, $\phi^u_T(\tilde{x}) = \phi^u_{t^*}\left(\phi^u_{T-t^*}(\tilde{x})\right)$. But $\phi^u_{T-t^*}(\tilde{x}) = \phi^u_T(x^*) \triangleq x_b \in S_b$. So $\phi^u_T(\tilde{x}) = \phi^u_{t^*}(x_b) \in S_b$ since S_b is forward invariant. Which implies that $\tilde{x} \in R^u_T$, hence the contradiction that proves the proposition. □

Proposition 2. *Given $u \in \mathcal{U}$, then S_b being forward invariant under that control law is equivalent to $S_b \subseteq R^u_T$ for all $T \geq 0$.*

Proof. Let's first assume that for all $T \geq 0$, $S_b \subseteq R^u_T$. By definition of R^u_T and because ϕ^u_T is an homeomorphism, $\forall T \geq 0,\ \phi^u_T(R^u_T) = S_b$. So for all $T \geq 0$, $\phi^u_T(S_b) \subseteq S_b$ which proves the necessity of forward invariance. The sufficiency of forward invariance follows directly from Prop. 1, as for all $T \geq 0$, $\phi^u_T(R^u_T) = S_b$ and $\phi^u_T(R^u_T) \subseteq R^u_T$ so $S_b \subseteq R^u_T$. □

Proposition 3. *Given $u \in \mathcal{U}$ and a forward invariant set $S_b = \{x \in \mathbb{R}^n \mid h_b(x) \geq 0\}$ with $h_b : \mathbb{R}^n \to \mathbb{R}$ smooth, then $R^u_T = \{x \in \mathbb{R}^n \mid h_b \circ \phi^u_T(x) \geq 0\}$.*

Proof. Consider $x \in R^u_T$, then $\phi^u_T(x) \in S_b$. So $h_b(\phi^u_T(x)) \geq 0$, hence $x \in \{x \in \mathbb{R}^n \mid h_b(\phi^u_T(x)) \geq 0\}$. Let's now consider $x \in \{x \in \mathbb{R}^n \mid h_b(\phi^u_T(x)) \geq 0\}$, then $\phi^u_T(x) \in S_b$, hence $x \in R^u_T$. □

So given a *small* forward invariant set, proposition (3) gives us access to *larger* forward invariant sets via the flow of the system. Because sets that are forward invariant for a given control law are obviously viable, it is now possible to use barrier condition (6) to guarantee that system (1) remains in R^u_T. Using Prop. 3, (6) evaluated at a current state x_0 becomes

$$\nabla(h_b \circ \phi^u_T)(x_0)\,\tilde{f}(x_0, u) + \alpha\left((h_b \circ \phi^u_T)(x_0)\right) \geq 0,$$

i.e.

$$\nabla h_b(x_T)\, D\phi^u_T(x_0)\, \tilde{f}(x_0, u) + \alpha(h_b(x_T)) \geq 0, \tag{10}$$

where $\tilde{f}(x_0, u) \triangleq f(x_0) + g(x_0)u(x_0)$ and $x_T \triangleq \phi^u_T(x_0)$.

What is notable in this last equation is that in order to evaluate it, one only needs to compute $\phi^u_T(x_0)$ and $D\phi^u_T(x_0)$, which is the key idea in this paper. Instead of computing large viable sets, one can just pick a smooth control law that makes a small subset of the safety set forward invariant and enforce forward invariance with respect to a larger viable subset of the safety set. This only requires one to evaluate the flow and the gradient of the flow locally at the current state. By enforcing (10) through (8), the system is constrained to stay inside R^u_T, which means that the system is guaranteed to only evolve in a region where the chosen *backup strategy* u can bring the system back to the backup set S_b. This way, *reachability* is enforced in a minimally invasive way, and the deployment of the backup strategy can be reserved for actual emergencies.

An important detail we have not touched on yet is that the flow needs to be differentiable in order to be able to compute $D\phi^u_T(x_0)$. Smoothness of the dynamics and the control law is sufficient in that regard, but is also restrictive. In practice, dynamics are very often smooth and control laws at least piecewise smooth. So when $D\phi^u_T(x_0)$ is not defined, the backup strategy which is effective at rendering R^u_T forward invariant can be used instead of the ASIF until the discontinuity is *crossed*.

B. Optimal Backup Strategy

The natural question that arises is "what is the *best* backup control law to choose?". That is, a control law that maximizes the size of R^u_T.

To address this question, we define the *closed-loop time-limited backward reachable set* of S_b as

$$\begin{aligned} &R_T \triangleq \\ &\{x \in \mathbb{R}^n \mid \exists u \in \mathcal{U},\ \exists t^* \in [0, T] :\ \phi^u_{t^*}(x) \in S_b\}, \end{aligned} \tag{11}$$

Then it immediately follows that for all $u \in \mathcal{U}$, $R_T^u \subseteq R_T$. So the *optimal backup control law*, if it exists, is the one that yields R_T.

Let's therefore consider the control law u^* given by

$$u^*(x_0) = u^*_{x_0}(0) \tag{12}$$

with $u^*_{x_0}$ being the control policy solution to the following optimal control problem:

MPC

$$\begin{aligned} u^*_{x_0} \triangleq \underset{u\in\mathcal{U}_T}{\text{argmax}} \quad & h_b(x(T)) \\ \text{s.t.} \quad & \dot{x} = f(x) + g(x)u(t) \\ & x(0) = x_0 \\ & u(t) \in U, \ \forall t \in [0,T] \end{aligned} \tag{13}$$

Remark 1. Note that the time here has been shifted such that $x_0 = x(t_0) = x(0)$ for convenience as only time-invariant control systems are considered.

Proposition 4. *Given u^* as defined in* (12) *and assuming $u^* \in \mathcal{U}$, $R_T^{u^*} = R_T$.*

Proof. The inclusion $R_T^{u^*} \subseteq R_T$ follows trivially from the definition (11). For the other inclusion, let's reason by contradiction and assume that $x_0 \in R_T$ but $x_0 \notin R_T^{u^*}$. This means that $\phi_T^{u^*}(x_0) \notin S_b$. But $\phi_T^{u^*}(x_0) = x(T)$, so the optimal value h_b^* of (13) under the control policy $u^* \in \mathcal{U}$ is such that $h_b^*(x(T)) < 0$. But $x_0 \in R_T$, so $\exists \tilde{u} \in \mathcal{U}$ such that $\phi_T^{\tilde{u}}(x_0) \in S_b$ since S_b is forward invariant under u, i.e. $h_b^*(x(T)) \geq 0$, which contradicts the fact that u^* is optimal for (13). Hence the contradiction proving that $R_T^{u^*} \supseteq R_T$. □

Remark 2. Note that S_b does not need to be forward invariant for Prop. 4 to hold. Our approach however requires S_b to be forward invariant under u^*. For that, h_b being a Control Lyapunov Function (CLF) for system (1) is sufficient as explained in [17]. This also implies that u^* yields a stabilizing controller and is therefore a *good* backup strategy [17].

We can now answer our initial question and assert that the *best* backup control law is u^* given by (11). Remember that in order to utilize this control law, one has to be able to compute $\nabla\left(h_b \circ \phi_T^{u^*}\right)(x_0)$ and $\phi_T^{u^*}(x_0)$. By solving (13), we actually get both of them. Indeed, $\phi_T^{u^*}(x_0) = x^*(T)$ where x^* is the optimal trajectory solution to (13), and

$$\nabla\left(h_b \circ \phi_T^{u^*}\right)(x_0) = \nabla h^*_{b,u^*}(x_0), \tag{14}$$

with h_b^* the optimal cost of (13). One will immediately recognize in (14) the co-states associated with (13) at time $t = 0$ [18]. So by solving (13) in a way that also solves for the co-states, all the information necessary to running the ASIF for $R_T^{u^*}$ can be recovered. Because (13) is not state constrained, the co-states are exactly the sensitivity of h_b to perturbation in the state [19], except when at a state x_0 where $u^*(x_0)$ isn't unique. One must also be careful when using u^* as in general it is not continuous. Particular care must be taken to verify that (13) yield a continuous u^* in order for the proposed method to work. Often, the cost of (13) can be only *slightly* modified to guarantee continuity with minimal loss of optimality.

IV. Online Reachability for Linear Systems

A. Implementation

Linear systems provide an ideal class of systems in which to illustrate the discussion of the previous section as MPC is *easy* for these systems (solvable in polynomial time). Let's consider linear control systems of the form

$$\dot{x} = Ax + Bu, \tag{15}$$

with $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$. For such systems, it is possible to exactly discretize the continuous dynamics, and the resulting MPC can be reduced to a finite dimensional convex optimization problem. Provided the control input is piecewise constant over intervals of duration $dt = T/N$, we can define

$$A_{dt} = e^{Adt}, \text{ and } B_{dt} = \left(\int_0^{dt} e^{A(dt-\tau)} d\tau\right) B, \tag{16}$$

and (15) *aligns exactly* with the discrete dynamics given by

$$x_{i+1} = A_{dt}x_i + B_{dt}u_i, \ i = 1..N. \tag{17}$$

The backup set can systematically be chosen to be an ellipsoidal set described by

$$h_b(x) = v - x^\top P x, \tag{18}$$

with $v \in \mathbb{R}^{+*}$ and $P \in \mathbb{R}^{n\times n}$ such that h_b is a CLF of (15).

The MPC (13) can be approximated by the following quadratic program:

MPC-Linear

$$\begin{aligned} \underset{\substack{u_1 \dots u_N \\ x_1 \dots x_{N+1}}}{\text{maximize}} \quad & \sum_{i=1}^{N+1} \beta_i h_b(x_i) \\ \text{s.t.} \quad & x_{i+1} = A_{dt}x_i + B_{dt}u_i, \ i = 1..N \\ & x_1 = x(t_0) \\ & u_i \in U, \ i = 1..N \end{aligned} \tag{19}$$

with β_{N+1} chosen to be *much larger* than the other β_i so that (19) yields a control law *close* to u^* but continuous. In this exact discretized version of (13), the co-states of (13) are equal to the Lagrange multipliers in (19). Let $\lambda^* \in \mathbb{R}^n$ be the vector of optimal Lagrange dual variables associated with the constraint $x_1 = x(t_0)$ in (19). Then the barrier condition (10) can be written as

$$\lambda^{*\top} A_{dt} x_0 + \lambda^{*\top} B_{dt} u(x_0) + \alpha(h^*) \geq 0. \tag{20}$$

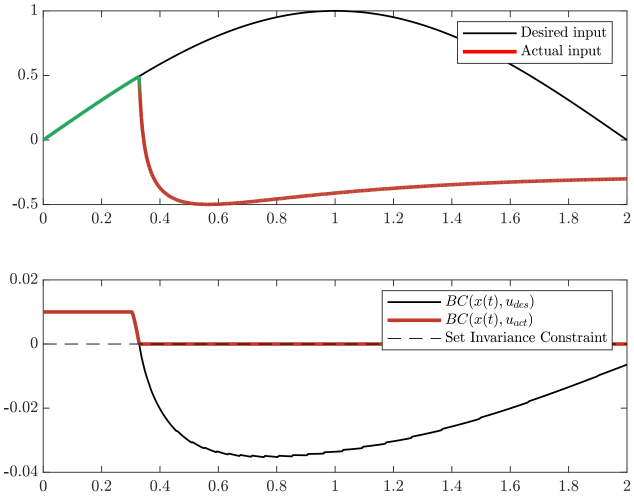

Fig. 3: Desired input (black line) and actual input (red and green line). The actual input is green when the reachability barrier condition is *not active* and red when *active*.

B. Numerical Example

Let's consider the linear inverted pendulum

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \tag{21}$$

with $U = [-1, 1]$. The target set is described as in (18) by

$$h_b(x) = 0.001 - x^\top \begin{bmatrix} 0.5858 & 0.2 \\ 0.2 & 0.4142 \end{bmatrix} x. \tag{22}$$

The time horizon is chosen to be $T = 1$, with $N = 200$. The extended class $\mathcal{K}$ function α in (20) is given by $\alpha(x) = 10x$. The system starts at $x(0) = [0, 0.2]^\top$ and is simulated over the time interval $[0, 2]$ using the ASIF (8) and a nominal input $u_{\text{des}}(t) = \sin\left(\frac{\pi}{2}t\right)$ (cf. Fig. 1).

As we can see in Fig. 3, the nominal input (dashed black line) is being followed (green part of the solid curve), until the barrier condition becomes active (red part of the solid curve). This filtering happens when the *end-point* of the MPC trajectory (in blue in Fig. 4) approaches the boundary of the backup set (black ellipsoid). The filtering stays active for the rest of the simulation and the system stays within reach of the target set, as shown by the fact that all MPC trajectories end in the backup set.

V. Online Reachability for Nonlinear Systems

A. Implementation

The MPC formulation (13) provides a control strategy that has the benefit of being maximally effective at bringing the trajectory into the backup set, and consequently leads to the *least intrusive* ASIF. However, when the system is not linear, MPC is NP-hard to implement in *real-time* which makes the use of an optimal control law almost impossible. In that case, one has to rely on finding an analytical control law as *close* as possible to the optimal one. This can for example be done using some form of reinforcement learning. In that case, provided that the dynamics and the control law are smooth, it is actually easy to compute $\phi_T^u(x_0)$ and $D\phi_T^u(x_0)$.

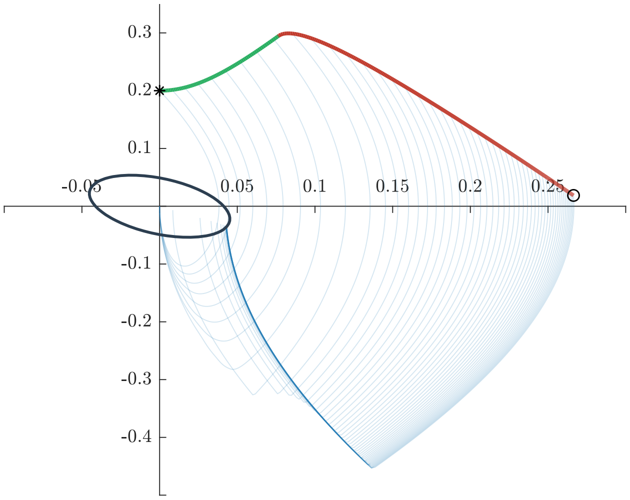

Fig. 4: State space trajectory of the system, green when the reachability barrier condition is *not active* and red when *active*. In blue are the backup trajectories with the MPC at certain simulation steps. In black is the backup set S_b.

By integrating (1) forward with that control law in the interval $[0, T]$, $\phi_T^u(x_0)$ can be computed, up to the numerical accuracy of the integration method of course. For $D\phi_T^u(x_0)$, one only need to integrate along with (1) a *sensitivity* matrix Q. As explained in [13], the dynamics of Q is given by

$$\frac{dQ(t)}{dt} = Df_{cl}(\phi_t^u(x_0))\, Q(t), \tag{23}$$

with $Q(0) = I$ and where $f_{cl}(x) \triangleq f(x) + g(x)u(x)$. In that case, the matrix $Q(t)$ represent exactly the Jacobian of ϕ_t^u at x_0:

$$Q(t) = D\phi_t^u(x_0). \tag{24}$$

Note that it is important that the control law used respects that $\forall t \in [0, T],\ u(t) \in U$. This is often overlooked as $u(t) \in U$ is a limitation imposed by hardware in the form of a *hard* clamping. In our case however, the closed loop dynamics has to be smooth. Smooth clamping can be achieved for an arbitrary control law fairly easily though, as finding a sequence of smooth functions that converge uniformly to the saturation function over $\mathbb{R}$ is quite trivial.

B. Numerical Example

We again consider the setup of Sec. IV-B. The backup control law u is chosen to be a saturated LQR regulator given by the nonlinear control law

$$u(x) = \sigma([5.58, 5.58]x), \tag{25}$$

where the smooth saturation function $\sigma : \mathbb{R} \mapsto [-1, 1]$ is given by

$$\sigma(x) = \frac{2}{1 + \exp(-2x)} - 1, \tag{26}$$

smoothly clamping the LQR to$[-1, 1]$.

The simulated behaviors for the input and trajectory are shown in Fig. 5, and Fig. 6 respectively. Here we see that

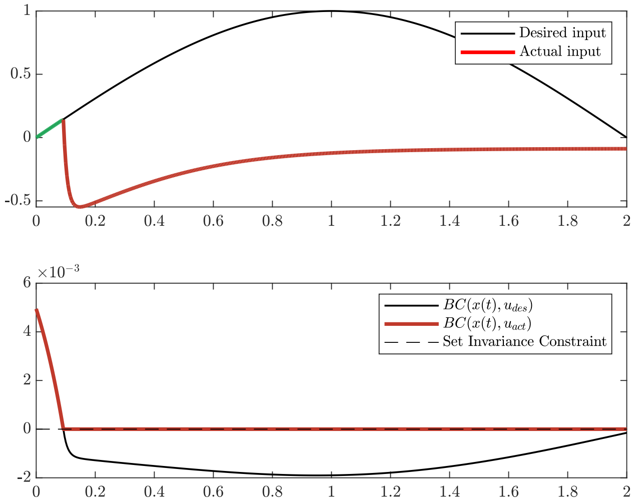

Fig. 5: Desired input (black line), and actual input (green and red line) at the top. The nominal input is green when the barrier condition is *not active* and red when *active*.

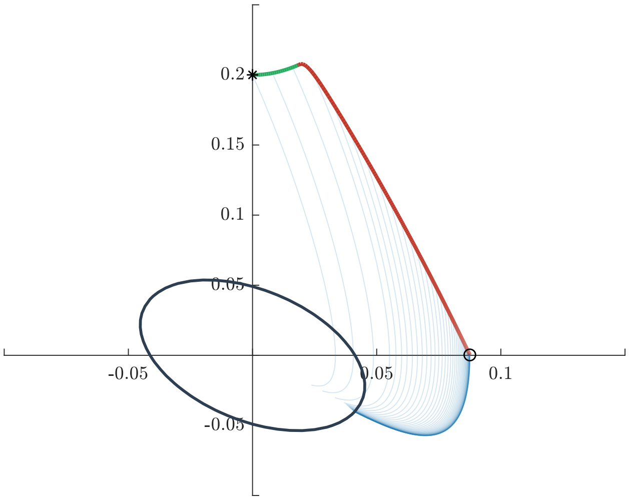

Fig. 6: Plot of the trajectory in the state space. The trajectory is green when the barrier condition is *not active* and red when active. In blue are the backup trajectories with the LQR at certain simulation steps. In black is the backup set S_b.

the endpoint of the trajectory approaches the boundary of S_b much sooner than with the MPC, resulting in the desired input being altered by the ASIF much earlier.

VI. Online Set Invariance Theory

In the previous sections, we showed how to keep a system within reach of a backup set given an arbitrary control law. The idea now is to use this capability to derive conditions to be enforced by the ASIF guaranteeing the system always remain in S. In this context, let's consider the set

$$\varOmega_T \triangleq \{x \in S \mid \forall t \in [0,T],\ \phi^u_t(x) \in S\}. \tag{27}$$

We can now state the following theorem.

Theorem 1. *Given $u \in \mathcal{U}$, if $S_b \subseteq S$ is forward invariant under u then for all $T \in \mathbb{R}^+$, $\tilde{S}_T \triangleq \varOmega_T \cap R^u_T$ is a subset of S that is forward invariant under that control law. In addition, for S described as in* (3)*, then*

$$\varOmega_T = \bigcap_{t\in[0,T]} \{x \in \mathbb{R}^n \mid h_i \circ \phi^u_t(x) \geq 0,\ i \in \{1..r\}\}. \tag{28}$$

Proof. The fact that $\tilde{S}_T \subseteq S$ follows trivially from the definition of $\varOmega_T$. Let's reason by contradiction and assume that $\tilde{S}_T$ is not forward invariant. This means that there exist $x^* \in \tilde{S}_T$ and $t^* \geq 0$ such that $\phi^u_{t^*}(x^*) \notin \tilde{S}_T$. But from (1) we know that R^u_T is forward invariant so $\phi^u_{t^*}(x^*) \in R^u_T$ and $\phi^u_{t^*}(x^*) \notin \varOmega_T$. Which implies that there exists $t^\# > T$ such that $\phi^u_{t^\#}(\phi^u_{t^*}(x^*)) \notin S$, or equivalently that there exists $t' > 0$ such that $\phi^u_{t'}(\phi^u_T(x^*)) \notin S$. But $x^* \in R^u_T$ so $\phi^u_T(x^*) \in S_b$ and because S_b is forward invariant, $\phi^u_{t'}(\phi^u_T(x^*)) \in S_b$, which contradicts $S_b \subseteq S$. Equation 28 follows trivially from the definitions of $\varOmega_T$ and S. □

So one can therefore use $\tilde{S}_T$ instead of S and be sure that the ASIF will be feasible and (1) will remain in S. Equation (28) can therefore be used along Prop. 3 to define an new set of admissible inputs for the ASIF-QP as barrier condition (6) evaluated at a current state x_0 for the set $\varOmega_T$ becomes

$$\nabla h_i(x_t)\, D\phi^u_t(x_0)\, \tilde{f}(x_0, u) + \alpha_i(h_i(x_t)) \geq 0, \tag{29}$$

for all $t \in [0,T]$ and $i \in \{1, \ldots, r\}$, with $x_t \triangleq \phi^u_t(x_0)$ and $\tilde{f}(\underbrace{x_0, u}) \triangleq f(x_0) + g(x_0)u(x_0)$. The set of admissible inputs $U_{\tilde{S}_T}(x)$ is therefore the set of $u \in \mathbb{R}^m$ such that

$$\left\{ \begin{array}{l} \nabla h_b(x_T)\, D\phi^u_T(x_0)\, \tilde{f}(x_0, u) + \alpha(h_b(x_b)) \geq 0 \\ \nabla h_i(x_t)\, D\phi^u_t(x_0)\, \tilde{f}(x_0, u) + \alpha_i(h_i(x_t)) \geq 0 \end{array} \right., \tag{30}$$

for all $t \in [0,T]$ and $i \in \{1, \ldots, r\}$, and this set is guaranteed to be non-empty for all $x \in \tilde{S}_T$ as $\tilde{S}_T$ is viable (assuming the α_i have been properly chosen of course).

One caveat though is that (29) is an *uncountable* set of linear constraints, which elevates the ASIF-QP into the class or *robust optimization problems*. However, when using the technique presented in Sec. V to evaluate x_T and $D\phi^u_T(x_0)$, the by-product of the integration is that x_t and $D\phi^u_t(x_0)$ are also available on a *countable* set of times $t_j \in \underbrace{[0,T]}$. So it is actually possible to *approximate* the set $\underbrace{U_{\tilde{S}_T}}(x)$ by considering a countable set of constraints (29) at these different t_j without extra computations than the ones required to compute (10). Interior point solver are good at handling a large number of constraints which makes this type of approximation actually tractable.

Imposing this *reachability* constraint can seem restrictive compared to pure *viability*, but in practice it is often an actual requirement. For commercial aviation and transoceanic flights for example, being able to land on airfield within a set amount of time is the safety criterion used by the Federal Aviation Administration (cf. ETOPS). So having the ability to enforce *reachability* constraints on top of set invariance can be beneficial, but if reachability is not a concern, one just needs to take T as large as possible to maximize the size of $\tilde{S}_T$. Even though we don't demonstrate it here, one can easily convince themselves that the closer to *optimal* the backup policy is, the larger $\tilde{S}_T$ is.

VII. Online Set Invariance for Linear Systems

In this section, we show how $\varOmega_T$ can be represented when the dynamics of the system are linear and the control policy chosen is a constrained version of (19).

A. Implementation

Consider system (15), and assume that S and U are polytopes. In this case, (19) is still a quadratic program:

MPC-Linear-Constrained

$$\begin{aligned} h^* = \max_{\substack{u_1 \dots u_N \\ x_1 \dots x_{N+1}}} \quad & \sum_{i=1}^{N+1} \beta_i h_b\left(x_i\right) \\ \text{s.t.} \quad & x_{i+1} = A_{dt} x_i + B_{dt} u_i,\ i = 1..N \\ & x_1 = x\left(t_0\right) \\ & x_i \in S,\ i = 1..N \\ & u_i \in U,\ i = 1..N \end{aligned} \tag{31}$$

The idea is to use a measure of feasibility of (31) to describe $\varOmega_T$. For that, we consider the radius of the largest sphere inscribed in F. Indeed, the radius of the inscribed sphere is monotonically increasing with the volume of the constraint set of (31) and is null only when (31) is infeasible, i.e. when $x\left(t_0\right) \in \partial\varOmega_T$. Denote by F the constraints set of (31) written here in condensed form:

$$F = \left\{ \left[\begin{array}{c} \bar{u} \\ \hline \bar{x} \end{array}\right] \in \mathbb{R}^{mN \times n\overline{N}} \;\middle|\; \begin{array}{l} A_e \left[\begin{array}{c} \bar{u} \\ \hline \bar{x} \end{array}\right] = b_e \\ A_i \left[\begin{array}{c} \bar{u} \\ \hline \bar{x} \end{array}\right] \leq b_i \end{array} \right\}, \tag{32}$$

with $\overline{N} = N + 1$.

Because F is not full dimensional, we need to consider a sphere inscribed in the projection $\tilde{F}$ of F on the dynamics plane $A_e x = b_e$. Let $N_c = mN + n\overline{N}$, $K \in \mathbb{R}^{N_c \times mN}$ be a basis of the null-space of A_e, and $x_e \in \mathbb{R}^{N_c}$ be a particular solution of $A_e x = b_e$. We choose the particular solution to be $x_e = A_e^+ b_e$, with Moore-Penrose pseudo inverse being given by $A_e^+ = A_e^\top \left(A_e A_e^\top\right)^{-1}$. Therefore

$$\left\{x \in \mathbb{R}^{N_c} \mid A_e x = b_e\right\} = \left\{Kz + x_e \mid z \in \mathbb{R}^{mN}\right\}. \tag{33}$$

So $\tilde{F}$ is the full dimensional polytope given by

$$\tilde{F} = \left\{z \in \mathbb{R}^{mN} \mid \tilde{A} z \leq \tilde{b}\right\}, \tag{34}$$

with $\tilde{A} = A_i K$ and $\tilde{b} = b_i - A_i x_e$, as we have $F = K\tilde{F} + x_e$.

Finding the radius of the sphere inscribed in $\tilde{F}$ can be achieved by solving the Linear Program [20]:

LP-Feasibility

$$\begin{aligned} r^* = \max_{r \in \mathbb{R},\ z \in \mathbb{R}^{mN}} \quad & r \\ \text{s.t.} \quad & \bar{A} \left[\begin{array}{c} r \\ \hline z \end{array}\right] \leq \tilde{b} \end{aligned} \tag{35}$$

where:

$$\bar{A} = \left[V \mid \tilde{A}\right] \text{ with } V = \sqrt{\operatorname{diag}\left(\tilde{A}\tilde{A}^\top\right)}$$

The following barrier condition is considered to enforce the feasibility of (31):

$$\frac{\partial r^*}{\partial x_1}\left(A_{dt} x_1 + B_{dt} u\right) + \alpha\left(r*\right) \geq 0. \tag{36}$$

The value of $\frac{\partial r^*}{\partial x_1}$ can again be extracted from the optimal dual variables λ^* in (35). We know that $\lambda^* = \frac{\partial r^*}{\partial \tilde{b}}$, so:

$$\frac{\partial r^*}{\partial b_e} = \frac{\partial r^*}{\partial \tilde{b}} \frac{\partial \tilde{b}}{\partial b_e} = \lambda^{*\top} \frac{\partial \tilde{b}}{\partial b_e}, \tag{37}$$

and

$$\frac{\partial \tilde{b}}{\partial b_e} = \frac{\partial\left(b_i - A_i x_e\right)}{\partial b_e} = -A_i A_e^+. \tag{38}$$

So $\frac{\partial r^*}{\partial x_1}$ are the rows of $\frac{\partial r^*}{\partial b_e}$ corresponding to the constraint $x_1 = x\left(t_0\right)$ in (35).

Note that $\tilde{A}$ contains lines equal to $0^\top$. This is because the state constraints on x_1 are *orthogonal* to the dynamics surface. Therefore r^* is missing some of the feasibility information about (35). To retrieve that information, we use the expression of $\varOmega_T$ given in (28) but only at time 0 as the rest of the constraints in (28) are completely contained in (36). So by enforcing (6) for S, (36) and (20) with appropriate choices of strengthening terms α, the MPC (31) is guaranteed to always be feasible, which translates to the ASIF also always being feasible. Note also that one must be careful when using this method as pathological cases can occur, for example when the set of constrains $\tilde{F}$ looses a dimension. Handling such case is outside the scope of this paper but definitely deserves closer attention before any kind of practical implementation can be attempted.

B. Numerical Example

Consider the same setup as in Sec. (IV-B) is considered, but with a time horizon $T = 0.7$, and a simulation over the time interval $[0, 1]$. The state bounds are chosen to be

$$x_1 \in [-0.1, 0.15] \text{ and } x_2 \in [-0.3, 0.25]. \tag{39}$$

As we can see in Fig. 7, the nominal input (dashed black line) is being followed (green part of the solid curve), until either of the 3 barrier conditions becomes active (red, orange or yellow part of the solid curve). As previously, the filtering happens when the *end-point* of the *projected trajectories* (in blue in Fig. 8) approaches the boundary of the backup set (black ellipsoid), but also when a point on the projected trajectories approaches the boundary of the safety set. The filtering stays active for the rest of the simulation and the system stays within reach of the target set and the system remains inside S (cf. Fig. 8).

Note that the *feasibility* is constrained to be greater than 0.4 to avoid numerical instabilities when the size of the constraint set F becomes too small. Not also that the choice of α is not made *a priori* here but is done as part of the optimization of the ASIF-QP. For that, the function α is chosen to be $\alpha\left(x\right) = \alpha_s x$ and α_s is considered as an extra decision variable in the ASIF-QP with appropriate lower bound and associated cost.

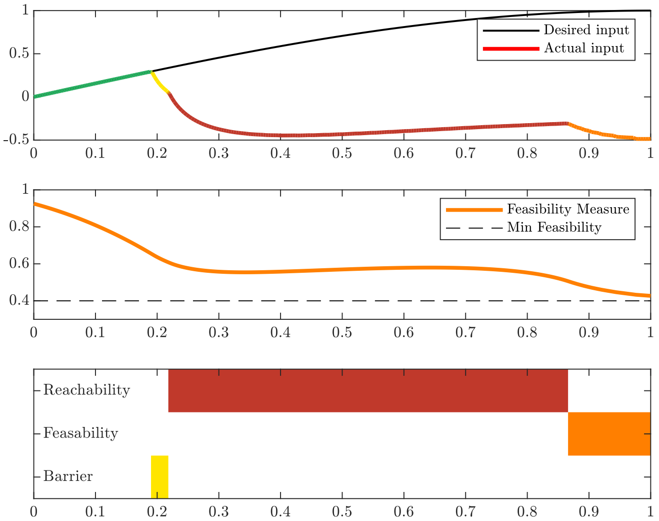

Fig. 7: Desired input (black line) and actual input (multicolored line). The actual input is green when no barrier condition is active and red, orange or yellow when one of the barrier condition is active.

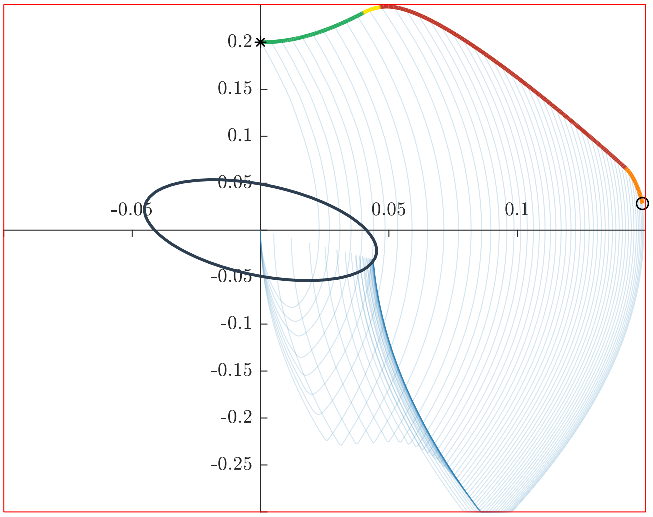

Fig. 8: State space trajectory of the system, green when no barrier condition is active and red, orange or yellow when one of the barrier condition is active. In blue, the trajectories corresponding to the backup strategy at each simulation step. In black is the backup set S and in red is the safety set S.

VIII. Conclusions and Future Work

This paper presented a method to compute *online* sensitivity information around a backup trajectory to replace the *offline* complex computations of viable sets necessary for the active set invariance task. We discussed how the operational freedom of the system is directly correlated to the optimality of the backup policy, and how sensitivity information can be extracted from an MPC controller in the case of linear systems. The effectiveness of the approach was illustrated in simulation on an unstable linear system.

Solving an MPC online is not trivial though, especially when the dynamics are nonlinear. An alternative implementation based on numerical integration has therefore been presented which allows for any control law to be used. The authors are currently exploring how machine learning techniques can be leveraged to approximate the MPC with an analytical expression. The authors are also exploring how *verified integration* techniques can be used to eliminate the approximations arising from numerical integration and discretization for the constraint space of the ASIF.

Acknowledgment

This work is supported by NSF awards #1724464, #1724457, #1446758 and #1555332.

References

[1] Edward A Lee. Cyber physical systems: Design challenges. In *Object oriented real-time distributed computing (isorc), 2008 11th ieee international symposium on*, pages 363–369. IEEE, 2008.

[2] Franco Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.

[3] Reza Ghaemi and Domitilla Del Vecchio. Control for safety specifications of systems with imperfect information on a partial order. *IEEE Transactions on Automatic Control*, 59(4):982–995, 2014.

[4] Jean-Pierre Aubin. *Viability theory*. Springer Science, 2009.

[5] Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 2016.

[6] Ian Mitchell. A summary of recent progress on efficient parametric approximations of viability and discriminating kernels. In *SNR@ CAV*, pages 23–31, 2015.

[7] Ian M Mitchell, Alexandre M Bayen, and Claire J Tomlin. A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on automatic control*, 2005.

[8] Xiangru Xu, Jessy W Grizzle, Paulo Tabuada, and Aaron D Ames. Correctness guarantees for the composition of lane keeping and adaptive cruise control. *IEEE Transactions on Automation Science and Engineering*, 2017.

[9] Jeremy H Gillula, Shahab Kaynama, and Claire J Tomlin. Sampling-based approximation of the viability kernel for high-dimensional linear sampled-data systems. In *Proceedings of the 17th international conference on Hybrid systems*, pages 173–182. ACM, 2014.

[10] Tom Schouwenaars. *Safe trajectory planning of autonomous vehicles*. PhD thesis, Massachusetts Institute of Technology, 2005.

[11] Stanley Bak, Deepti K Chivukula, Olugbemiga Adekunle, Mu Sun, Marco Caccamo, and Lui Sha. The system-level simplex architecture for improved real-time embedded system safety. In *Real-Time and Embedded Technology and Applications Symposium, 2009. RTAS 2009. 15th IEEE*, pages 99–107. IEEE, 2009.

[12] Thomas Gurriet, Andrew Singletary, Jake Reher, Laurent Ciarletta, Eric Feron, and Aaron Ames. Towards a framework for realizable safety critical control through active set invariance. In *Proceedings of the 9th International Conference on Cyber-Physical Systems*. IEEE, 2018.

[13] Hans Seywald and Renjith R Kumar. Desensitized optimal trajectories. *Advances in the Astronautical Sciences*, 93(1):103–116, 1996.

[14] Franco Blanchini and Stefano Miani. *Set-theoretic methods in control*. Springer, 2008.

[15] Urs Borrmann, Li Wang, Aaron D Ames, and Magnus Egerstedt. Control barrier certificates for safe swarm behavior. *IFAC-PapersOnLine*, 48(27):68–73, 2015.

[16] John M Lee. Smooth manifolds. In *Introduction to Smooth Manifolds*, pages 1–29. Springer, 2003.

[17] Ali Jadbabaie, Jie Yu, and John Hauser. Unconstrained receding-horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 46(5):776–783, 2001.

[18] Arthur Earl Bryson. *Applied optimal control: optimization, estimation and control*. CRC Press, 1975.

[19] Hans Seywald and Renjith R Kumar. Finite difference scheme for automatic costate calculation. *Journal of Guidance, Control, and Dynamics*, 19(1):231–239, 1996.

[20] Jianzhe Zhen and Dick Den Hertog. Computing the maximum volume inscribed ellipsoid of a polytopic projection. *INFORMS Journal on Computing*, 30(1):31–42, 2017.