Towards a Framework for Realizable Safety Critical Control through Active Set Invariance

Thomas Gurriet[1], Andrew Singletary[2], Jacob Reher[1], Laurent Ciarletta[3], Eric Feron[2], and Aaron Ames[1]

***Abstract*—This paper presents initial results towards a realizable framework for the safety critical controlled invariance of cyber-physical systems. The main contribution of this paper is the development of a control barrier function based methodology which can be used to enforce set invariance on systems in the presence of non-linear disturbances and uncertainty. The first part of this work is a review of the current methods available for finding viable sets and how they are linked to practical choices regarding safety. Their limitations and directions towards improvements when it comes to handling model uncertainty are also highlighted. The second part of this work is the formulation of a condition which can guarantee set invariance in the presence of generic uncertain in the dynamics. An associated optimization problem to enforce that condition is proposed and a method to convexify the problem and make it solvable in real-time is formally presented. The effectiveness of the proposed framework is illustrated experimentally on a two-wheeled inverted pendulum.**

I. Introduction

The last couple of decades have seen the rapid emergence of many enabling technologies in the field of robotics, thanks to the increasing computational power density of processors, the development of small and cost-effective sensors, and advances in power electronics. As a result, extraordinary robot behaviors are now common in the lab, but these have yet to be translated to our everyday life. This is in no small part due to a lack of the ability to guarantee the safety of these systems. Safety is a widely used concept in every day life, but one that can be difficult to translate rigorously. Safety is centered around the idea of the constrained behavior of a system. If a system behavior can be constrained, safety in its most common meaning simply becomes a matter of constraining it to a desired behavior deemed “safe”. We will therefore adopt the notion of set invariance for dynamical systems as the formal translation of the concept of safety.

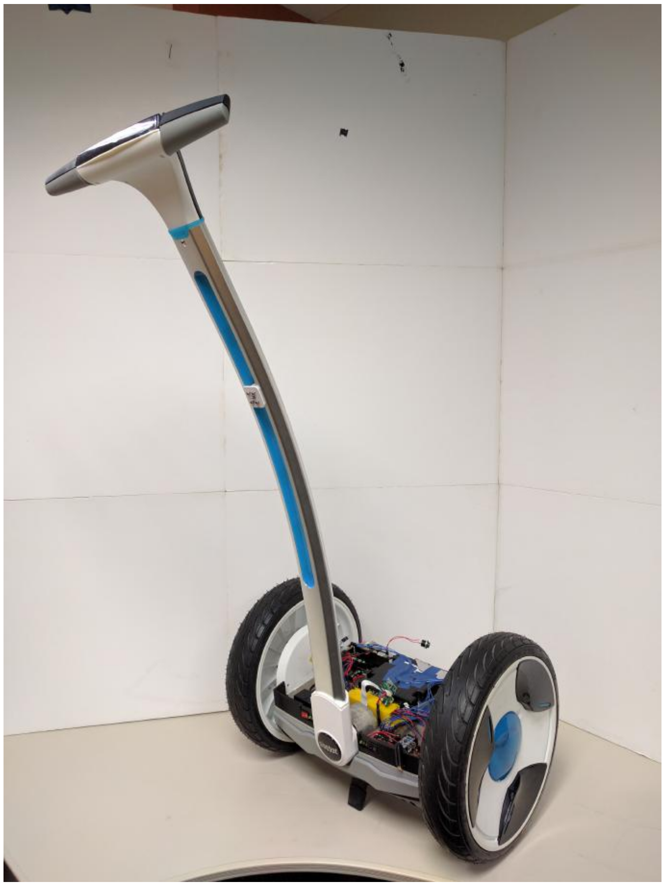

Fig. 1: Experimental platform: a two-wheeled inverted pendulum.

In this paper, we will use the terminology of a “safety set” of a system to denote the set of allowed states for this system. The task of controlling the system in order to guarantee the invariance of this safety set during all of the evolution of the system will be the main focus of this work. It is important to note that all guarantees provided will be in terms of invariance of the safety set, which does not necessarily guarantee the safety of the real system in the practical and legal sense that is required for any real-life application. The guarantees in terms of system safety will always come, first and foremost, from a proper definition of the system requirements and their proper translation into a safety set, as well as from a proper implementation of the required algorithms.

Generally, controlled invariance of an arbitrary safety set cannot be guaranteed, especially when the control input is constrained, as it is for any real system. Therefore, a subset of the safe set must be found such that it can be rendered invariant, therefore making the safety set invariant as well. If it exists, such a subset is said to be viable [1]. The largest of these subsets is known as the viability kernel. Knowing

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

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

[3] Laurent Ciarletta is with the Madynes team Inria, LORIA UMR 7503, Vandoeuvre-lès-Nancy, 54506, France.

the viability kernel would in theory allow for maximum operational space while ensuring that the system remains in its safety set for all times. However, the choice of a viable set to render invariant has to be made in light of many practical considerations as we will discuss, and the viability kernel may not necessarily be the ideal choice of viable subset of the safety set. The chosen viable subset of the safety set we will be referred to as the "**safety kernel**" in this paper.

Finding these viability sets and kernels for different class of systems has been the focus of multiple research over the years [2]. There exist many different methods to find viable sets for linear systems [3], [4], [5], but general methods applicable to more realistic non-linear systems [6], [7], [8] suffer from the usual curse of dimensionality. Exactly solving this problem of viability is in general as hard as solving HJB-like equations [8], hence making solving it online on embedded hardware currently infeasible. Therefore, the usual solution is to solve this global problem offline and extract from it just the necessary information needed to then online actively maintain the system in the safety set.

Several approaches to what we will call "**active set invariance**" control approach (ASI for short) have been proposed [9], [10], [11], [12] with varying degrees of success in implementation. All these algorithms undoubtedly exhibited safety enhancing behavior but they still did not provided strict safety guarantees for the real system. All these algorithms are based on rigorous proofs of set-invariance, but they currently fail to address our inability to characterize the world with arbitrary precision and accuracy. Including disturbances and uncertainty in the mathematical framework of ASI is still relatively new [13], [14] and will be one of the main focuses of this paper. Our approach is in essence similar to [14] but tries to overcome the limitations in the type of uncertainty that can be addressed in [14] by proposing a method to "convexify" the space of constraints making the problem tractable.

This paper presents an generic approach to active set invariance for dynamical systems in the presence of disturbances and uncertainty. In Section 2, the choice of a safety kernel to render invariant will be discussed in light of practical and technical considerations. In Section 3, control barrier functions [11] will be introduced and the subsequent formulation of an optimization based input filter capable of handling disturbances and uncertainty will be detailed and formally justified via the main result of the paper. Finally, section 4 will present the implementation of the framework presented above to a two-wheeled inverted pendulum and discuss some preliminary experimental results.

II. Finding control invariant sets

A. Viability approach

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

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

with f and g being locally Lipschitz continuous functions defined on an open set $O \subseteq \mathbb{R}^n$ and $u \in U$ a compact set of $\mathbb{R}^m$. Note that the variations of the input u must guarantee the uniqueness of solutions to (1) for most of the discussed concept to make sense. The Lipschitz continuity of the input provides for example a sufficient condition in that regard.

As discussed in the introduction, we will tackle the issue of safety from a set-invariance perspective. In practice, there are several "passive" ways to make sure a system does not exit its safety set, but for highly dynamic systems the problem becomes more complex, especially if these systems are unstable. The first question to naturally arise when faced to this problem is : "Is it possible to control the system to as to render the safety set invariant?". Viability Theory [1] focuses on addressing this question at an abstract level and we will borrow some of the vocabulary and ideas introduced by this theory.

Definition 1. For the dynamical system (1), a closed set $S \subset O$ is said to be a **viable set** if for all initial conditions $x(t_0) \in S$, there exist an input signal $u(t) \in U$ such that $\forall t \geq t_0,\ x(t) \in S$.

As arbitrary safety sets are in general not viable, one must first find a subset of the safety set that is viable to allow for most active set invariance methods to work. We will call these subsets of the safety set safety kernels.

Definition 2. For the dynamical system (1) and a closed safety set $S \subset O$, a **safety kernel** $\tilde{S}$ is a viable subset of S.

The biggest viable subset of S is called the viability kernel and can be considered as the "optimal safety kernel", hence making it one of the centers of attention of the viability research effort. It has been shown that the viability kernel can be expressed as a solution to an Hamilton-Jacobi partial differential equation [8], which makes the search for this set fundamentally hard.

Let's briefly review the idea behind this approach. Considering the safety set S as the target set of a backwards reachability problem, which can be described as the zero sub-level set of $g(x)$,

$$S = \mathcal{G}_0 = \{x \in \mathbb{R}^n \mid g(x) \leq 0\} \tag{2}$$

The backwards reachable set can be calculated by solving for the viscosity solution of a terminal value Hamilton-Jacobi-Isaacs partial differential equation.

$$\frac{\partial \Phi}{\partial t} + \min\left[0, H(x, \nabla\Phi(x,t))\right] = 0 \tag{3}$$

where $H(x, \nabla\Phi(x,t))$ is the Hamiltonian of the system. The terminal condition for the problem is

$$\Phi(x, 0) = g(x) \tag{4}$$

Thus, the backwards reachable set for some finite time $t = \tau < 0$ is

$$\mathcal{G}(\tau) = \{x \in \mathbb{R}^n \mid \Phi(x,\tau) \leq 0\} \tag{5}$$

By letting τ approach $-\infty$, the backwards reachable set can be computed for a near-infinite time horizon, which gives an approximation of the viability kernel. A level set toolbox for MATLAB® was designed for finding viscosity solutions to Hamilton-Jacobi PDE's hence allowing to approximate viability kernels for generic non-linear systems [15]. The main drawback of this method however is its exponential scaling with the dimension of the system, due to the required gridding of the state-space. More recent algorithms based on parametric set representations, shown in [2], have been developed to handle higher-dimensional systems but have yet to be shown to work for non-linear systems.

B. Lyapunov approach

Even though the viability kernel is of fundamental interest, it is not the only choice of safety kernel. Sometimes, we don't only want the system to stay within a given set, but we also want to be able to reach a given safe state in case of emergency. This is an issue that is critical in many practical situations. UAV's, for example, may need to stay within reach of an unpopulated area at all time. Hence, there is another set that is important to consider beside the safety set : the "emergency set". The emergency set represent a point or a set in the state space that a controller must stabilize to in case of emergency. Therefore the safety kernel in this context must be a basin of attraction of the emergency controller that will be used to stabilize the system to the emergency set.

Definition 3. Let $\Phi(t, x)$ be the solution to (1). A **basin of attraction** B (or region of attraction) is the set

$$B \triangleq \left\{ x \in \mathbb{R}^n \mid \begin{array}{c} \Phi(t,x) \text{ is defined } \forall t \geq 0, \\ \lim_{t\to\infty} \Phi(t,x) = x^* \end{array} \right\} \tag{6}$$

where x^* is the equilibrium point of interest.

This idea is not new but to the extend of our knowledge, all the methods implementing this idea [12], [9] switch to the emergency controller when "getting close" to unreachability of the emergency set. This strategy is very punishing, especially in the context of human operated systems. One way to overcome these issues can be to choose as the safety kernel a basin of attraction that is contained in the safety set. Then, using the ASI filtering method discussed later, we can guarantee the invariance of the safe set as well as the reachability of the emergency set in a minimally invasive way, while reserving the switching to the emergency controller to actual emergency situations.

Finding basins of attraction is closely related to finding Lyapunov functions, which is arguably an easier task than finding viability kernels. While finding a Lyapunov function for a system is nontrivial, a significant step forward has been made with the development of sums of squares (SOS) techniques, making it to some extent easier than finding a viability kernel in terms of computational complexity. An SOS optimization approach to finding polynomial Lyapunov functions has become quite popular.

Let's briefly review the ideas behind SOS optimization for finding regions of attraction. Interested readers can find more details in [16], [17], [18].

Definition 4. A polynomial p is a **sum of squares (SOS)** if there exist polynomials $\{q_i\}_{i=1}^N$ such that $p = \Sigma_{i=1}^N q_i^2$

Using the Lyapunov asymptotic stability theorem, we can characterize a region of attraction by:

$$B \triangleq \{x \in \mathbb{R}^n \mid V(x) \leq \gamma, \text{ and } \nabla V(x) f(x) \leq 0\} \tag{7}$$

with $V(x)$ a positive definite function. Therefore, given polynomial Lyapunov function and dynamics, one can formulate an SOS program that checks if

$$\forall x \in \{x \mid V(x) \leq \gamma\}, \ \nabla V(x) f(x) \leq 0 \tag{8}$$

for a given $\gamma > 0$. The positivstellensatz characterization of polynomials (or S-procedure) tells us that if there exists a polynomial s that is SOS such that

$$-\left(\nabla V(x) \cdot f + s(\gamma - V(x))\right) \text{ is SOS,}$$

then the contractiveness condition (8) is satisfied. Checking that a polynomial is SOS being equivalent to solving an SDP program, it is therefore a convex problem. MATLAB® toolboxes such as SOSTOOLS and SPOT exist for these methods, which convert SOS or modified SOS constraints into an SDP problem. However, SDP solvers are still relatively slow and better-scaling method of checking SOS tbased on diagonally-dominant sum of squares optimization are presented in [19].

Similarly to viability algorithms, it is interesting to try and find the largest ROA possible. In that case, the problem become non-linear, but iterative approaches exist to try and overcome this increase in complexity by sequentially searching for a Lyapunov function and a region of attraction [18].

C. Addressing model uncertainty

Until now, we have only ideal system like (1). In practice however, uncertainty is a integral part of engineering and controller design. Let us therefore expand the dynamics (1) to include parameter uncertainty and disturbances:

$$\dot{x} = f(x, p, w) + g(x, p, w)u \tag{9}$$

where f and g are locally Lipschitz continuous functions with respect to x, continuous in w, and defined on a proper open set $O \subset \mathbb{R}^n \times \mathbb{R}^{n_p} \times \mathbb{R}^{n_w}$ with $u \in U \subset \mathbb{R}^m$ a compact set. Here, p corresponds the parameters of the model assumed fixed in time, and the disturbance term w is assumed to be locally Lipschitz continuous. Given (9), it is in general impossible to render a set invariant for any values of p and w. Bounds must therefore be chosen for these terms in light of practical and operational consideration in order to be able to construct an appropriate safety kernel and provide guaranties of its invariance. It is important to realize that there is a fundamental trade-off between the tolerated amount of uncertainty and the performance of the system quantified by the size of the safety kernel.

Let's define the notion of "robust viability" in this particular context.

Definition 5. For the dynamical system 9, a closed set $S \subset O$ is said to be a **robust viable set** (or discriminating kernel) if for all $p \in P$, $w \in W$ and $x(t_0) \in S$, there exist an input signal $u(t) \in U$ such that $\forall t \geq t_0,\ x(t) \in S$.

Few of the methods discussed so far are readily able to handle model uncertainty, which remains to be a significant area of improvement. The reachability method in [8] can handle the addition of a disturbance terms included in the Hamiltonian, however handling parametric uncertainty is much more challenging. An addition of a disturbance term was implemented to a system in [18] for the sum of squares optimization method, which required a sacrifice to optimality and made the challenge of finding an initial, feasible polynomial more difficult. Other approached like [20] are this point but but have yet to be

One last point to consider when working with practical cyber-physical systems is the model of time used. Until now we had considered that everything works in continuous-time, but in actuality, the sensing data and controller work in discrete time. Treating the system as a sampled data system, as defined in [2], involves treating the plant dynamics as continuous and the controller as discrete. Lyapunov-like sufficient conditions to ensure stability for sampled-data systems are introduced in [21], and attempting to apply them to the optimization methods mentioned previously is a current area of research for the team. When it comes to viability set search algorithms, to the best of our knowledge only linear dynamics can currently be handled [10] in the context of sampled data systems.

III. ASI FILTERING STRUCTURE

In the previous section, we have reviewed the different methods available to find safety kernel that guarantee the feasibility of the active set invariance task. What we discuss in this section is a general approach to impose this set invariance in a modular and minimally intrusive way. This proposed approach inserts a filter between the controller and the system. This way, the ASIF (active set invariance filter) actively filters the inputs sent by the controller (cf. figure 2) to ensure that they do not make the system leave the chosen safety kernel $\tilde{S}$.

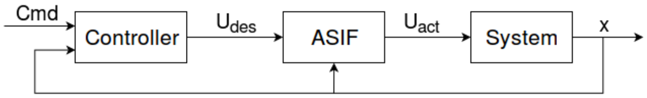

Fig. 2: ASI control structure

In general, there exists for most states in $\tilde{S}$ a range of allowable inputs that one must select from. That choice should not impact the enforced set invariance property of the ASIF. This is why an ASIF based on constrained optimization is a natural choice for this task. Note, however, that without more information than the output of the controller, only "blind" point-wise optimality can be achieved. An ASIF structure where the cost function is provided by the controller along with the desired input is an interesting solution to that problem and will be the subject of future research.

A. Barrier functions

Let us now shift the attention to a control law that can guarantee the forward invariance of the safety kernel $\tilde{S}$ (which has been constructed as viable subset of S). The core concept behind the proposed ASIF comes from Nagumo's theorem [22]. Nagumo's theorem states that the forward invariance of $\tilde{S}$ is equivalent to the "sub-tangentiality condition" (10):

$$f(x) + g(x)u\left(x,t\right) \in \mathcal{T}(x), \tag{10}$$

for all $x \in \tilde{S}$ and $t \geq t_0$, where $\mathcal{T}(x)$ is the tangent cone [1], [22] to $\tilde{S}$ at x. Note that the variations of the input $u(x,t)$ must guarantee uniqueness of solutions to (1) for this statement to hold. The challenge now is to find a representation of $\tilde{S}$ for which the expression of $\mathcal{T}(x)$ is tractable. The approach proposed by [11] represents $\tilde{S}$ as the zero superlevel set of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$. However, such functions can only describe smooth sets. In this section, we will extend this representation to a particular class of non-smooth sets called "practical sets" [22]. To describe such sets, one only needs to consider r continuously differentiable functions $h_i : \mathbb{R}^n \to \mathbb{R}$ such that:

$$\begin{aligned} \tilde{S} &= \left\{x \in \mathbb{R}^n \mid h_i\left(x\right) \geq 0,\ i \in \llbracket 1, r \rrbracket\right\} \\ \partial\tilde{S} &= \left\{x \in \mathbb{R}^n \mid h_i\left(x\right) = 0,\ i \in \llbracket 1, r \rrbracket\right\}. \end{aligned} \tag{11}$$

where $\llbracket 1, r \rrbracket = \{1...r\} \subset \mathbb{N}$. For such a set, the tangent cone to $\tilde{S}$ at x can be expressed very simply by[1]:

$$\mathcal{T}(x) = \left\{z \in \mathbb{R}^n \mid \forall i \in Act(x),\ \nabla h_i(x).z \geq 0\right\}, \tag{12}$$

where:

$$Act(x) \triangleq \left\{i \in \llbracket 1, r \rrbracket \mid h_i(x) = 0\right\}. \tag{13}$$

The sub-tangentiality condition (10) can therefore be elegantly written as:

$$L_f h_i(x) + L_g h_i(x) u\left(x,t\right) \geq 0, \tag{14}$$

for all $x \in \partial\tilde{S}$, $t \geq t_0$, and $i \in Act(x)$. This formulation, however, is not very practical as it only constraints the input when the system is on the boundary of the safety kernel, which is impossible to determine with absolute accuracy in practice. The idea proposed in [11] is to introduce a "strengthening term" in (14) and to impose this new "barrier condition" for all states in $\tilde{S}$:

$$L_f h_i(x) + L_g h_i(x) u(x,t) + \alpha(h_i(x)) \geq 0, \tag{15}$$

for all $x \in \tilde{S}$, $t \geq t_0$ and $i \in \llbracket 1, r \rrbracket$, with $\alpha : \mathbb{R} \to \mathbb{R}$ an extended class $\mathcal{K}$ function as defined in [11]. This barrier condition and the uniqueness of solutions to (1) being satisfied on $\tilde{S}$ implies by definition of α that the "sub-tangentiality condition" (10) is also satisfied, hence guaranteeing its forward invariance as proven in [11].

Because $\tilde{S}$ is viable by construction, the existence of an input $u\left(x,t\right)$ satisfying (15) is guaranteed for all $x \in \partial\tilde{S}$. It is easy to show that given a viable set the feasibility of this barrier condition can be guaranteed on all of $\tilde{S}$ with a proper choice of strengthening function α.

[1] If $Act(x) = \emptyset$, $\mathcal{T}(x) = \mathbb{R}^n$

Proposition 6. *Consider the dynamical system* (1) *and set* $\tilde{S}$ *characterized by* (11) *with a continuously differentiable functions* h_i, $i \in [\![1,r]\!]$. *If* $\tilde{S}$ *is viable, and for all* $i \in [\![1,r]\!]$, $L_f h_i(x)$ *and* $L_g h_i(x)$ *are bounded on* $\tilde{S}$, *then there exists an extended class* $\mathcal{K}$ *function* $\tilde{\alpha} : \mathbb{R} \longrightarrow \mathbb{R}$ *such that for all* $x \in \tilde{S}$ *there exists an input* $u \in U$ *for which condition* (15) *is satisfied.*

Proof: Consider the set

$$A(r) \triangleq \left\{ x \in \tilde{S} \mid \exists i \in [\![1,r]\!],\ 0 \leq h_i(x) \leq r \right\} \subseteq \tilde{S},$$

and let's define function $\alpha : \mathbb{R} \longrightarrow \mathbb{R}$ as:

$$\alpha(r) \triangleq -\min_{\substack{x \in A(r) \\ u \in U}} \left(\min_{i \in [\![1,r]\!]} \left(L_f h_i(x) + L_g h_i(x) u \right) \right).$$

Because U is compact, $L_f h_i(x)$ and $L_g h_i(x)$ are bounded on $\tilde{S}$ and $L_f h_i(x) + L_g h_i(x) u$ is continuous with respect to both x and u, α is a well defined non-decreasing function on $\mathbb{R}$ for which for all $x \in \tilde{S}$ and $i \in [\![1,r]\!]$ there exist $u^*(x)$ such that $-\alpha(h(x)) \leq L_f h_i(x) + L_g h_i(x) u^*(x)$. Because $\tilde{S}$ is viable, then for all $x \in \partial\tilde{S}$ there exist an input $u^{**} \in U$ such that $L_f h_i(x) + L_g h_i(x) u^{**} \geq 0$, for all $i \in Act(x)$. Let's notice that $A(0) = \partial\tilde{S}$, and hence that $\alpha(0) \leq 0$. Because it is always possible to find an extended class $\mathcal{K}$ function $\tilde{\alpha}$ such that for all $\lambda \in \mathbb{R}$, $\tilde{\alpha}(\lambda) \geq \alpha(\lambda)$, then for all $x \in \tilde{S}$ and $i \in [\![1,r]\!]$ there exist $u^*(x)$ such that $-\tilde{\alpha}(h_i(x)) \leq L_f h_i(x) + L_g h_i(x) u^*(x)$. ■

Remark 7. Note that the choice of functions α and h_i is arbitrary. It would be interesting to investigate whether or not it is possible to guarantee the forward invariance of a sampled-data system [3], [23] with a proper choice of α and h_i while imposing (15) only at the sampled points.

The question now is how to actually impose the set-invariance condition derived above. In this same paper [11], the authors use quadratic programming as a tool to unify Control-Lyapunov and Control Barrier Functions in one safe and performant controller. What will interest us in that approach is its flexibility as it can actually be used in conjunction with any controller similarly to the structure presented in figure 2. Imposing the barrier condition (15) is very natural in this optimization problem where minimizing the normed difference between the desired and actual inputs provide the best level of fidelity to the controller commands while guaranteeing invariance of the safety set. The QP based ASIF is given by:

Ideal-QP

$$\begin{aligned} u^*_{\text{act}}(x,t) = \underset{u_{\text{act}} \in U}{\text{argmin}} \quad & (u_{\text{des}}(t) - u_{\text{act}})^2 \\ \text{s.t.} \quad & BC_i(x, u_{\text{act}}) \geq 0,\ \forall i \in [\![1,r]\!] \end{aligned} \tag{16}$$

where:

$$BC_i(x,u) \triangleq L_f h_i(x) + L_g h_i(x) u + \alpha(h_i(x)).$$

Note that when $r > 1$ or U is a polytope, deriving an analytical solution is possible but tedious as the number of conditions increase exponentially as 2^{r+2m} when U is an hyper-cube of $\mathbb{R}^m$. Nevertheless, it can still be computationally interesting for small values of r, especially if parallel computing capabilities are available.

Furthermore, the Lipschitz continuity of such controllers has been studied in [24], but the necessary conditions proposed are too restrictive for a generic safety kernel and allowable input set. This issue is however non-existant in the case of sampled-data systems which are actually a more accurate representation of reality than purely continuous systems.

B. Robust ASIF formulation

Up to this point, only the safety of the ideal system (1) is guaranteed (under the assumption of continuous feedback of course). It is however very hard if not impossible to derive a model that describes with absolute exactness the behavior of a complex physical system. Therefore, it is fundamental to address the effects of disturbances and model uncertainty in our ASI framework. The first step in that process has already been discussed in the previous section as having a safety kernel that is robust is mandatory to guaranteeing the feasibility of the optimization problem solved by the ASIF. The second step is to provide a new set-invariance condition that will ensure forward invariance of the safety kernel even if the dynamics are not known exactly. Let's define the "robust barrier condition":

$$RBC_i(x,p,w,u) \geq 0, \tag{17}$$

where :

$$RBC_i(x,p,w,u) \triangleq L_f h_i(x,p,w) + L_g h_i(x,p,w) u + \alpha(h_i(x)).$$

Then, if (17) is satisfied for all $x \in \tilde{S}$, $p \in P$, $w \in W$, and $i \in [\![1,r]\!]$, it is satisfied in particular for the actual values of the parameters and disturbances. Provided that $\tilde{S}$ is a robust viable set, then (6) can be extended to provide a choice of α guaranteeing the non-emptiness of the set of inputs satisfying (17) in $\tilde{S}$. Therefore, a guaranteed feasible robust ASIF can be formulated as:

Robust-QP

$$\begin{aligned} u^*_{\text{act}}(x,t) = \underset{u_{\text{act}} \in U}{\text{argmin}} \quad & (u_{\text{des}}(t) - u_{\text{act}})^2 \\ \text{s.t.} \quad & RBC_i(x,p,w,\varepsilon,u_{\text{act}}) \geq 0, \\ & \forall p \in P,\ \forall w \in W,\ \forall i \in [\![1,r]\!] \end{aligned} \tag{18}$$

This optimization problem now belong to the class of robust optimization problems[25], [26]. In general, the dependency in the uncertainty terms is **not linear** making the problem hard to solve. However, it is relatively easy to find polytopic over-approximations of the safety constraint set as we will discuss later. If one is able to provide such an over-approximation, then the difficult optimization problem (18) can be replaced by a less optimal quadratic program but one that still guarantees set invariance. First let us prove a simple lemma.

Lemma 8. *Let $f : \mathbb{R}^m \longrightarrow \mathbb{R}$ and $g : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ be any two continuous functions. Let $A \in \mathbb{R}^n$ be a compact set. Let's define:*

$$\begin{aligned} B &\triangleq \{g(x) \in \mathbb{R}^m \mid x \in A\} \\ C &\triangleq \{f(y) \in \mathbb{R} \mid y \in B' \supseteq B\} \\ D &\triangleq \{f \circ g(x) \in \mathbb{R} \mid x \in A\}, \end{aligned}$$

where B' is compact, then:

$$\left[\max_{y \in B'} (f(y))\right] \leq 0 \implies \left[\max_{x \in A} (f \circ g(x))\right] \leq 0. \tag{19}$$

Proof: First let's note the problem is well posed since by continuity of f and g, B, C and D are compact. Hence, Weierstrass' theorem guarantees the existence of the maximums in (19). We also have:

$$\begin{aligned} \max_{x \in A} (f \circ g(x)) &= \max_{z \in D} (z) \\ \max_{y \in B'} (f(y)) &= \max_{z \in C} (z). \end{aligned} \tag{20}$$

Let $x \in A$, then $f \circ g(x) \in D$. But $g(x) \in B \subseteq B'$, hence $f(g(x)) \in C$. So $D \subseteq C$. Which means that:

$$\left[\max_{z \in D} (z)\right] \leq \left[\max_{z \in C} (z)\right],$$

hence (19). ■

Let's now prove the main theorem of this paper.

Theorem 9. *Consider the dynamical system* (9) *and a practical robust viable set $\tilde{S}$ characterized by continuously differentiable functions h_i, $i \in [\![1, r]\!]$. Then it is possible to reformulate* (18) *as a quadratic program that guarantees forward invariance of $\tilde{S}$.*

Proof: Let's start by rewriting the constraint set of (18) as an accessory optimization problem. Given $x \in \tilde{S}$ and $u_{act} \in U$, for all $p \in P$, $w \in W$, $\varepsilon \in \Upsilon$, and $i \in [\![1, r]\!]$, then:

$$RBC_i(x, p, w, u_{act}) \geq 0, \tag{21}$$

if and only if the optimal value of the accessory optimization problem:

$$\max_{\substack{p \in P \\ w \in W}} \left(a_i(x, p, w)^\top u_{act} + b_i(x, p, w)\right),$$

is negative for all $i \in [\![1, r]\!]$ with:

$$\begin{aligned} a_i(x, p, w) &\triangleq -L_g h_i(x, p, w) \\ b_i(x, p, w) &\triangleq -L_f h_i(x, p, w) - \alpha(h_i(x, p, w)). \end{aligned}$$

The core idea of this proof is to now transform this non-linear accessory optimization problem into a linear program with a lower objective value. Given $x \in \tilde{S}$ and $u_{\text{act}} \in U$, then from Lemma 8 we have:

$$\left[\max_{p \in P,\, w \in W} \left(\sum_{j=1}^{m} \left(a_{i,j}(x, p, w)\, u_{\text{act},j}\right) + b_i(x, p, w)\right)\right] \leq 0$$

$$\Uparrow$$

$$\left[\begin{array}{rl} \max\limits_{\tilde{a}_{i,j},\, \tilde{b}_i} & \left(\sum\limits_{j=1}^{m} \left(\tilde{a}_{i,j} u_{\text{act},j}\right) + \tilde{b}_i\right) \\ \text{s.t.} & \tilde{a}_{i,j} \in \{a_{i,j}(x, p, w) \mid p \in P,\ w \in W\} \\ & \tilde{b}_i \in \{b_i(x, p, w) \mid p \in P,\ w \in W\} \end{array}\right] \leq 0,$$

for all $i \in [\![1, r]\!]$, which can be rewritten in a more compact form as:

$$\begin{array}{rl} c^*(x) = \max\limits_{\widetilde{a_i}} & \left(\widetilde{a_i}^\top \widetilde{u}\right) \\ \text{s.t.} & D_i(x)\, \widetilde{a_i} \leq d_i(x) \end{array}, \tag{22}$$

for all $i \in [\![1, r]\!]$, where $\widetilde{a_i} \triangleq [[\tilde{a}_{i,1}, ..., \tilde{a}_{i,m}]\, |b_i]^\top$ and $\widetilde{u} \triangleq [u_{\text{act}}|1]^\top$. Here we have, in the context of Lemma 8, chosen the set B' as the smallest hyper-box containing B but the result still holds for any polyhedral enclosure of B. Hence, $d_i(x)$ and $D_i(x)$ encode the bounds on $\tilde{a}_{i,j}$ and $\tilde{b}_i$ and we leave it to the reader to derive the exact content of $d_i(x)$ and $D_i(x)$.

Because (22) is a linear program (guaranteed feasible by a proper choice of $\tilde{S}$), we can use strong duality to see that (22) is equivalent to:

$$\begin{array}{rl} c^*(x) = \min\limits_{\widetilde{\lambda_i}} & \left(\widetilde{\lambda_i}^\top d_i(x)\right) \\ \text{s.t.} & \widetilde{\lambda_i}^\top D_i(x) = \widetilde{u} \\ & \widetilde{\lambda_i} \geq 0 \end{array}, \tag{23}$$

for all $i \in [\![1, r]\!]$. So given $x \in \tilde{S}$ and $u_{\text{act}} \in U$ such that the optimal value of (23) is negative, condition (21) is also satisfied hence guaranteeing set invariance of $\tilde{S}$, and the resulting robust optimization problem is indeed a quadratic program since:

$$\left(\begin{array}{rl} \min\limits_{u_{\text{act}} \in U} & (u_{\text{des}}(t) - u_{\text{act}})^2 \\ \text{s.t.} & c^*(x) \leq 0 \end{array}\right) =$$

$$\left(\begin{array}{rl} \min\limits_{u_{\text{act}} \in U,\, \widetilde{\lambda_i}} & (u_{\text{des}}(t) - u_{\text{act}})^2 \\ \text{s.t.} & \widetilde{\lambda_i}^\top d_i(x) \leq 0 \\ & \widetilde{\lambda_i}^\top D_i(x) = \widetilde{u} \\ & \widetilde{\lambda_i} \geq 0. \end{array}\right) \tag{24}$$

■

Remark 10. We end up with a robust ASIF formulation (24) which is computationally tractable and still guarantees forward invariance of $\tilde{S}$. However, this simplification of the constraint space comes at the expense of added conservatism on the allowable input space. It would therefore be interesting to quantify the induced shrinking of the allowed input space.

The proposed formulation (24) is actually fairly generic. Even systems that are not affine in input can be handled at the expense of a loss in performance, as the allowed input space shrinks because of the conservatism added to the accessory optimization problem. It would be interesting to see how this framework adapts to set representations coming from the field of computer vision as brushed over in [27]. Some of our future work will focus on utilizing this framework to allow autonomous vehicles to explore unknown and potentially dynamic environments safely.

IV. Practical implementation on a two-wheeled inverted pendulum

A. The experimental platform

The system we chose to test our framework is a commercial two-wheeled inverted pendulum (cf. figure 1). To fit our experimental needs, all the electronics has been replaced and only the main frame, the motors and the battery remain unchanged. The power drive is entirely contained in the RoboteQ FBL-2360 motor controller. The motor controller also provides motor speed feedback and low-level information on the power train. A low-level safety board has also been designed specifically for this systems in order to allow for the vehicle to be remotely shut down in case emergency, even in case of a lockup of the rest of the system.

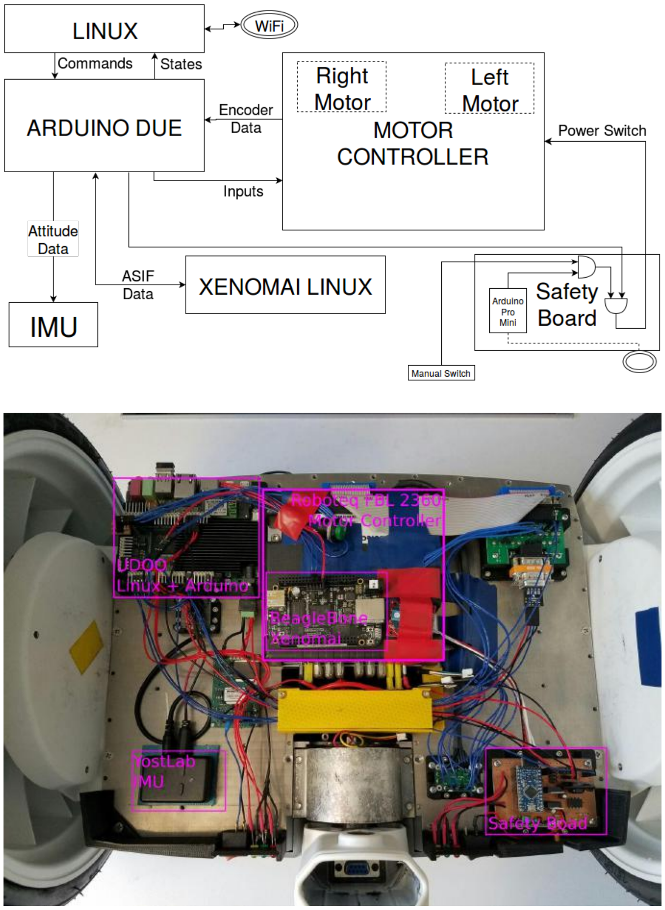

Fig. 3: Schematic of the system

The information processing and control is handled by three separate embedded computers (cf. figure 3). Two are regrouped on an UDOO-quad, an i.MX6 quad-core ARM processor running Ubuntu 14.04 and an Arduino Due. The Arduino Due runs all the low level system management, allowing for good real-time capabilities whereas the Linux side of the UDOO takes care of logging data and relaying communication between the Arduino and our ground station control via WiFi. This communication is based on a ROS network running on the Linux side of the UDOO. A GUI designed using the MATLAB® AppDesigner environment allows us to display the system states in real-time, send commands and log the data sent back by the on-board systems (cf. figure 4).

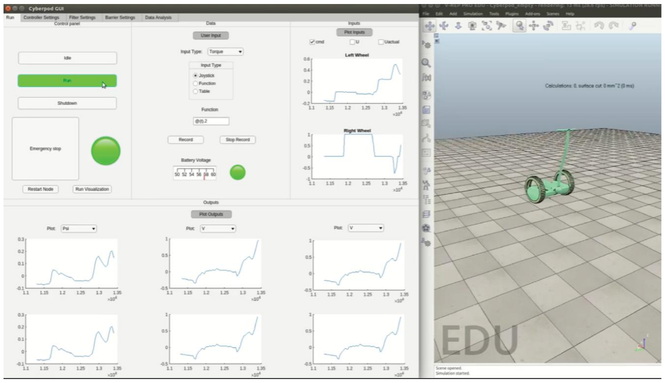

Fig. 4: GUI developped for the system

A BeagleBone Black running a Xenomai kernel is also connected to the system allowing us to run computationally intensive algorithms with real-time requirements. Finally, a Yost Labs' 3-Space IMU provides attitude information at 250Hz.

B. ASIF implementation

The system model was derived using a first order model of the motors, and assuming no sliding between the wheels and the ground and no friction in the joints. Only the 2 dimensional behavior of the system (no turning) is considered in this example. The equations of motion for the system, with parameter values entered, are shown below with v being the speed of the segway, ϕ the tilt angle, and u de commanded voltage to the motors:

$$\frac{d}{dt}\begin{pmatrix} v \\ \phi \\ \dot{\phi} \end{pmatrix} =$$

$$\begin{pmatrix} \frac{\cos(\phi)(-1.8u+11.5v+9.8\sin(\phi))-10.9u+68.4v-1.2\dot{\phi}^2\sin(\phi)}{\cos(\phi)-24.7} \\ \dot{\psi} \\ \frac{(9.3u-58.8v)\cos(\phi)+38.6u-243.5v-\sin(\phi)(208.3+\dot{\phi}^2\cos(\phi))}{\cos^2(\phi)-24.7} \end{pmatrix}$$

For demonstration purposes, the vehicle is controlled by a simple, hand tuned PD controller, that tracks desired velocity. The input bounds are $u \in [-15, 15]$V. The controller does a good job at stabilizing the system up-right, though it is easy to make the robot fall down with overly ambitious commands. For testing our ASIF, the desired safe set was a pitch angle $\phi \in \left[-\frac{\pi}{12}, \frac{\pi}{12}\right]$rad, $v \in [-5, 5]$ m/s, and $\dot{\phi} \in [-2\pi, 2\pi]$ rad/s.

Using this state-space constraints and the dynamics of the system, the non-robust viability kernel was approximated using the Hamilton-Jacobi method described above (cf. figure 5). First, the reachability analysis was performed on a 75x75x75 grid of the state space, with the edges of the grid at the state constraints of the system. After this result converged, polynomial regression was used to create an analytical representation of the set that can be used in the ASIF. While the parametric uncertainty was not included in the viability kernel approximation, the input was constrained to 30% of its actual bounds, which was found experimentally to be sufficient for the ASIF to always be feasible.

The QP solver used for the experiment was qpOASES [28], [29], as it is very efficient for solving sequential problems such as this one. In the implementation of the ASIF, it took the BeagleBone Black 0.4 ms on average to solve the robust QP of five decision variables and 4 constraints. It also takes 0.8 ms to transmit the data back and forth between the BeagleBone and Arduino Due. To make the the safety constraint space convex, as discussed in Theorem (9), the interval arithmetic library [30] was used. Of the 0.4 ms solve time, less that 0.1 ms is used to do the interval arithmetic computations.

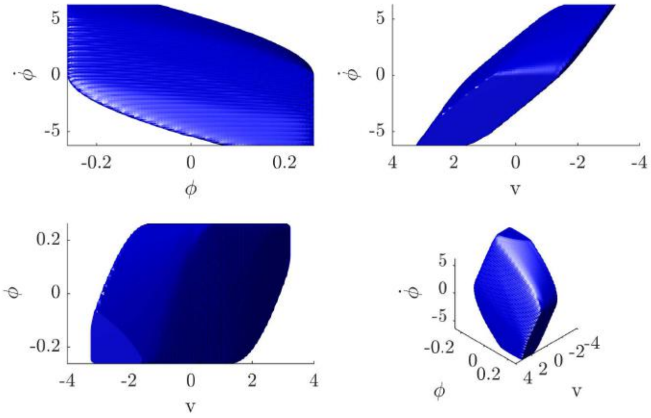

Fig. 5: Safety kernel of the system for $\phi \in \left[-\frac{\pi}{12}, \frac{\pi}{12}\right]$

C. The experiments

First, the vehicle is given a desired pseudo-sinusoidal input, which cause the pitch angle of the vehicle, ϕ, to exceed the imposed $\frac{\pi}{12}$ angle constraint when the ASIF is not active. Two runs were performed, one without the ASIF engaged, and one with. The results are shown in figure 6. It is clear that the input is being regulated in a manner that prevents the safety condition from being breached, but in a way that minimally alters the overall trajectory.

However, this only works because the system is known with sufficient accuracy. If mass is added on the system, the ideal ASIF becomes unable to maintain the vehicle pitch in the desired safety set. Figure 7 illustrates with fact when the controller is turned off and only the ASIF is active. To address this modeling uncertainty, we implemented the robust formulation derived in the previous section, and as expected the pitch angle remains between the prescribed bound with various values of disturbance load (cf. figure 7).

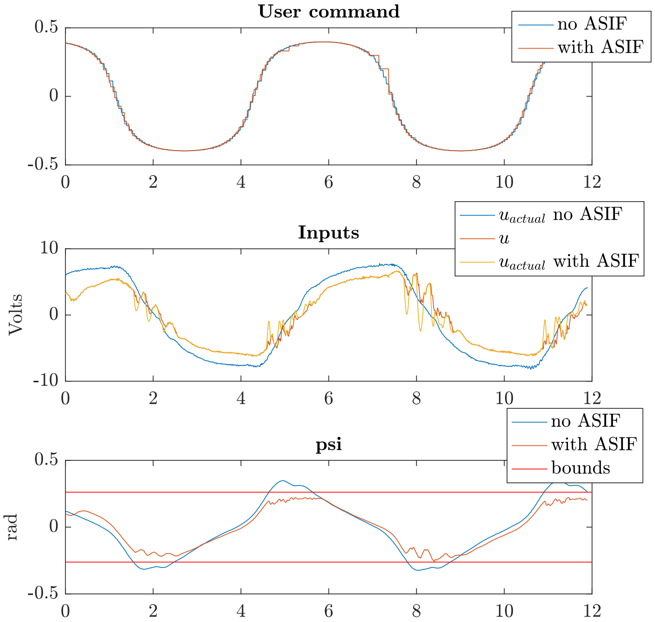

Fig. 6: LQR w/ and w/o ASIF

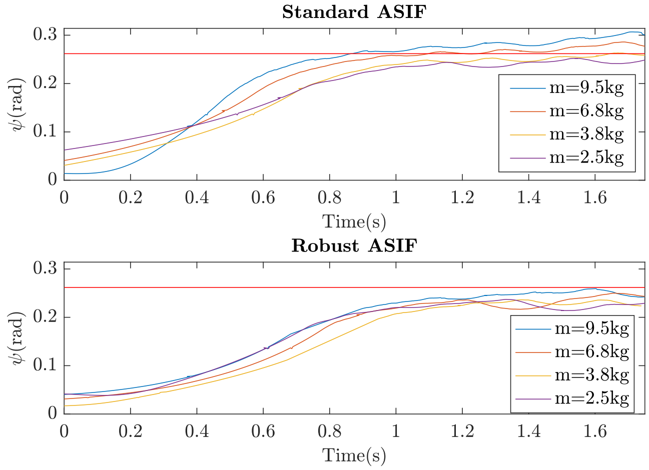

Fig. 7: Standard ASIF tuned for m = 2.5kg and Robust ASIF tuned for m $\in [2.5, 9.5]$ kg, with varying loads.

In this last experiment, the controller is turned of and the system is left to lilt over, until the ASIF kicks in and prevent it from going past the chosen angle limit.

D. Robustness to external disturbances

Furthermore, tests have been conducted for the non-robust ASIF when the system is perturbed by an impulse force. The PD controller is activated and a zero-velocity command is sent so as to keep the system up-right and immobile (cf. figure 8). This type of distrubanced essentially corresponds to an instantanuous change of the system state. As long as the system is inside the safety kernel when the disturbance vanishes, the non-ASIF is sufficient to ensure set-invariance. So as expected, if the ASIF isn't present, the PD controller reacts very violently to this perturbation which triggers a failsafe when the vehicle tilt goes $\frac{\pi}{4}$. When is ASIF is active however, the system is kept within the safety set and the disturbance is essentially damped out in a minimally invasive way. A video of this experiment is provided in [31].

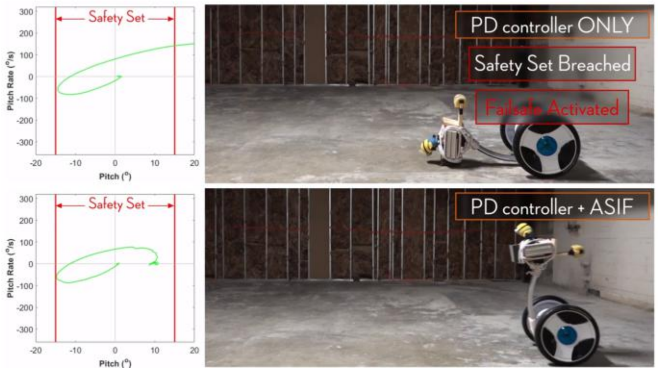

Fig. 8: ASIF with an impulse disturbance.

V. Conclusions and Future Work

In this work, we began to provide a framework for ensuring the safety of realistic cyber-physical systems, which are prone to disturbances and parametric uncertainty. The methods that currently exist for approximating viable sets and basins of attractions, while not refined for such systems, have been outlined and implemented. An active set invariance filter based on control barrier functions was created and successfully implemented on a two-wheeled inverted pendulum. This ASIF was shown to prevent the robot from leaving its defined safe set. Future work involves applying this framework to higher-dimensional systems, as well as streamlining the process of accounting for parametric uncertainty and disturbance terms. An alternative approach based on online optimization is also being examined, which would allow working with higher-dimensional systems without the need to globally solve for the safety kernel beforehand.

Acknowledgment

This work is supported by NSF CPS award #1724457.

References

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

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

[3] 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.

[4] Antoine Girard, Colas Le Guernic, and Oded Maler. Efficient computation of reachable sets of linear time-invariant systems with inputs. In *International Workshop on Hybrid Systems: Computation and Control*, pages 257–271. Springer, 2006.

[5] Shahab Kaynama, Ian M Mitchell, Meeko Oishi, and Guy A Dumont. Scalable safety-preserving robust control synthesis for continuous-time linear systems. *IEEE Transactions on Automatic Control*, 2015.

[6] Noël Bonneuil. Computing the viability kernel in large state dimension. *Journal of Mathematical Analysis and Applications*, 2006.

[7] John N Maidens, Shahab Kaynama, Ian M Mitchell, Meeko MK Oishi, and Guy A Dumont. Lagrangian methods for approximating the viability kernel in high-dimensional systems. *Automatica*, 2013.

[8] 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.

[9] 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.

[10] I. M. Mitchell, J. Yeh, F. J. Laine, and C. J. Tomlin. Ensuring safety for sampled data systems: An efficient algorithm for filtering potentially unsafe input signals. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 7431–7438, Dec 2016.

[11] 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.

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

[13] Anayo K Akametalu, Jaime F Fisac, Jeremy H Gillula, Shahab Kaynama, Melanie N Zeilinger, and Claire J Tomlin. Reachability-based safe learning with gaussian processes. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 1424–1431. IEEE, 2014.

[14] Quan Nguyen and Koushil Sreenath. Optimal robust safety-critical control for dynamic robotics. *International Journal of Robotics Research (IJRR)*, in review, 2016.

[15] Ian M Mitchell. The flexible, extensible and efficient toolbox of level set methods. *Journal of Scientific Computing*, 35(2):300–329, 2008.

[16] Weehong Tan and Andrew Packard. Searching for control lyapunov functions using sums of squares programming. In *42nd Annual Allerton Conference on Communications, Control and Computing*, 2004.

[17] Anirudha Majumdar, Amir Ali Ahmadi, and Russ Tedrake. Control design along trajectories with sums of squares programming. *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[18] Zachary Jarvis-Wloszek, Ryan Feeley, Weehong Tan, Kunpeng Sun, and Andrew Packard. Control applications of sum of squares programming. In *Positive Polynomials in Control*. Springer, 2005.

[19] Anirudha Majumdar, Amir Ali Ahmadi, and Russ Tedrake. Control and verification of high-dimensional systems with dsos and sdsos programming. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, page 394. IEEE, 2014.

[20] S. Azizi, L. A. B. Torres, and R. M. Palhares. Regional robust stabilisation and domain-of-attraction estimation for mimo uncertain nonlinear systems with input saturation. *International Journal of Control*, 91(1):215–229, 2018.

[21] Iasson Karafyllis and Costas Kravaris. Global stability results for systems under sampled-data control. *International Journal of Robust and Nonlinear Control*, 19(10):1105–1128, 2009.

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

[23] Ian M Mitchell, Shahab Kaynama, Mo Chen, and Meeko Oishi. Safety preserving control synthesis for sampled data systems. *Nonlinear Analysis: Hybrid Systems*, 10:63–82, 2013.

[24] Benjamin J Morris, Matthew J Powell, and Aaron D Ames. Continuity and smoothness properties of nonlinear optimization-based feedback controllers. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pages 151–158. IEEE, 2015.

[25] Dimitris Bertsimas, David B Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM review*, 53, 2011.

[26] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*. Princeton University Press, 2009.

[27] Thomas Gurriet and Laurent Ciarletta. Towards a generic and modular geofencing strategy for civilian uavs. In *International Conference on Unmanned Aircraft Systems*, pages 540–549. IEEE, 2016.

[28] Anil Aswani, Patrick Bouffard, Xiaojing Zhang, and Claire Tomlin. Practical comparison of optimization algorithms for learning-based mpc with linear models, 2014.

[29] H.J. Ferreau, C. Kirches, A. Potschka, H.G. Bock, and M. Diehl. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363, 2014.

[30] Michael Lerch, German Tischler, Jürgen Wolff Von Gudenberg, Werner Hofschuster, and Walter Krämer. Filib++, a fast interval library supporting containment computations. *ACM Transactions on Mathematical Software (TOMS)*, 32(2):299–324, 2006.

[31] Experimental tests on segway. http://youtu.be/c2D1IVlE7XU.