Robotics: Science and Systems 2020
Corvalis, Oregon, USA, July 12-16, 2020

Eyes-Closed Safety Kernels: Safety of Autonomous Systems Under Loss of Observability

Forrest Laine, Chih-Yuan Chiu, and Claire Tomlin

***Abstract*—A framework is presented for handling a potential loss of observability of a dynamical system in a provably safe way. Inspired by the fragility of data-driven perception systems used by autonomous vehicles, we formulate the problem that arises when a sensing modality fails or is found to be untrustworthy during autonomous operation. We cast this problem as a differential game played between the dynamical system being controlled and the external system factor(s) for which observations are lost. The game is a zero-sum Stackelberg game in which the controlled system (leader) is trying to find a trajectory which maximizes a function representing the safety of the system, and the unobserved factor (follower) is trying to minimize the same function. The set of winning initial configurations of this game for the controlled system represents the set of all states in which safety can be maintained with respect to the external factor, even if observability of that factor is lost. This is the set we refer to as the Eyes-Closed Safety Kernel. In practical use, the policy defined by the winning strategy of the controlled system is only needed to be executed whenever observability of the external system is lost or the system deviates from the Eyes-Closed Safety Kernel due to other, non-safety oriented control schemes. We present a means for solving this game offline, such that the resulting winning strategy can be used for computationally efficient, provably-safe, online control when needed. The solution approach presented is based on representing the game using the solutions of two Hamilton-Jacobi partial differential equations. We illustrate the applicability of our framework by working through a realistic example in which an autonomous car must avoid a dynamic obstacle despite potentially losing observability.**

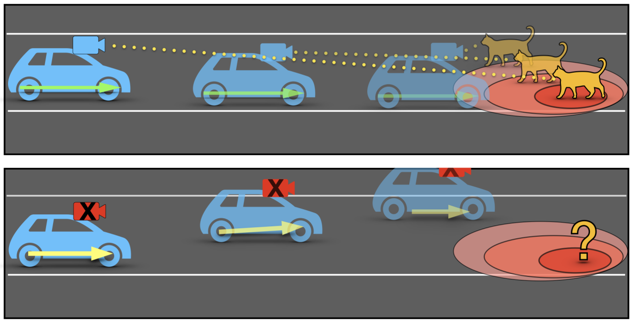

Fig. 1. Graphic Depiction of the Loss of Observability Situation. When planning avoidance maneuvers, the vehicle must assume what type of observations it will receive of the cat in the future. If it assumes persistent observation of the cat , it can plan a maneuver which relies on the ability to sense and avoid the cat. This situation is depicted in the top pane of the graphic. When the vehicle reaches the location of the cat, it avoids the cat through a region that the cat *could have been in*, but was detected to not be in. Alternatively, if the vehicle assumes it will not receive further observations of the cat, it must avoid the region in which the cat might be even though it cannot observe the cat to determine its exact location. We refer to this region as the Eyes-Closed Safety Kernel. The trajectory in the top pane might be preferred, but may not be safe if observations are lost during the maneuver. The difference between these two models is subtle but important. The design of controllers that are safe with respect to this latter paradigm is the topic of this article.

I. Introduction

Before deploying autonomous systems into uncontrolled environments, a degree of confidence in the ability of those systems to operate safely and proficiently is obviously needed. This need is exacerbated for safety-critical systems that can become dangerous in their failure modes. One way to generate confidence in a system is to prove safety properties for a system model. Although this method is imperfect—system model assumptions may not hold in practice—the space over which the assumptions do hold becomes a validated operating region in which the performance of the system is guaranteed.

In this article, we present a method for generating a control scheme for which the safety of the modeled system can be proven, without assuming the *persistence of observability* of the system. System analysis often involves the assumption of an observation model, common variants of which include full observations (all system state dimensions are observed), partial observations (only some state dimensions are observed), and noisy observations (observations are corrupted by noise). Typically, these observation models are assumed to be constant throughout the mission of the autonomous system. We relax that assumption to allow for cases in which the observability of the system (or part of the system) might suddenly be lost. We refer to the corresponding set of states for which safety is guaranteed, despite loss of perception of the external factor, as the Eyes-Closed Safety Kernel.

Losses of system observability have always been an important consideration, as sensors can fail for various reasons, but accounting for these losses has become increasingly necessary as the perception modules used in modern autonomous vehicles have transitioned to data-driven components. It is very hard to verify that these components will always function as intended, especially if the input data encountered at run-time is dissimilar to the data for which those systems were trained on. This poses a significant issue for safety-critical systems. We would like to be able to leverage the advances and strengths of data-driven techniques, yet we desire that the autonomous systems that we develop are still provably safe.

One approach to developing such systems might derive from *anomaly detection*, which aims to predict whether the input data is anomalous with respect a previously-verified data set, and therefore has a high risk of causing the sensing modality to fail. This is potentially an easier problem than guaranteeing correctness over geometric regions of the data space. If it is accepted that the sensing modality can fail unexpectedly during operation, care must be taken so that the autonomous system is able to maintain safe and proficient operation despite such failures potentially happening during dynamic situations.

The purpose and contribution of this article is the design of a control scheme which can guarantee safe handling of a detected loss of observability of the autonomous system or factors it interacts with. We design such a controller by formulating and solving a differential game representing the situation of interest. We do *not* address the challenge of correctly detecting anomalous inputs to the data-driven algorithms (or other predictions of failure), and instead point to the growing literature of methods interested in addressing that problem, such as [1], [2], [3], [4], [5], [6] to list just a few. The framework presented here is complementary to those methods, offering a mathematically sound means of handling such detected anomalies in the context of autonomous systems.

A graphic overview of the problem formulation is depicted in Figure 1. We use this graphic to reiterate the key difference between the method presented here and that of the standard safe obstacle-avoidance paradigms. Typical formulations of obstacle-avoidance can similarly be cast as a differential game such as in [7], although those and other formulations assume (to the best of our knowledge) a persistent observation model. **Under the standard assumption, avoidance trajectories are planned assuming that the system can sense and react to obstacles in the future. This is in contrast to the framework we present, where avoidance trajectories are generated assuming that *observations will potentially be lost at any instant*.** To the best of our knowledge, this is the first work to account for such situations in a provably safe way, without introducing any additional conservatism beyond what this observation model necessitates.

Before presenting this framework in detail, we summarize the contributions we make:

- We develop a mathematical formulation of the loss of observability scenario as a zero-sum, differential Stackelberg game.
- We develop a novel means of computing a global solution to the game by showing how it can be equivalently expressed through two related partial differential equations. We derive this alternative representation and demonstrate how the global solution can be solved numerically.
- We work through an example demonstrating the practical applicability of our approach to robotic systems. Particularly, we indicate how an autonomous vehicle can (in real time) avoid collision with a dynamic obstacle when an unreliable perception module is responsible for tracking it.

II. Formulation

We begin presentation of our framework by first introducing a model of the types of autonomous systems we wish to consider. In this section and the remainder of this article, we refer to this model as the system of consideration itself, although it should be kept in mind that all properties stated are either modeling assumptions or properties derived from the model.

To represent the problem of interest, we assume that the full state of the autonomous system can be decomposed into two independent sub-systems, which we refer to as the internal and external systems. We assume that the internal and external systems have independent controls and state evolutions, which depend only on their respective states and controls. We assume that both systems are able to observe their own state at all times, that the external system can observe the internal system at all times, and that the internal system can observe the external system in an impersistent manner, meaning observations can be lost and regained at arbitrary times. We take the perspective of the internal system, and aim to control it in a manner that maintains a safety specification involving both systems, robust to potentially impersistent observations of, and adversarial control by the external system.

Figure 1 depicts an example of a system which can be decomposed as such. There, the vehicle is the internal system and the obstacle (cat) is the external system. The steering rate and acceleration of the vehicle might be the internal controls, and the directional velocity of the cat might be the external control.

Assume without loss of generality that time begins at some $t_0 \geq 0$, at which point the state of both systems are known to both systems, within some bounded error tolerance. Define $x_{int} \in \mathbb{R}^{n_{int}}$ and $x_{ext} \in \mathbb{R}^{n_{ext}}$ as the internal and external states, respectively, evolving according to the dynamics:

$$\begin{aligned} \dot{x}_{int}(t) &= f_{int}(x_{int}(t), u_{int}(t)), & t &\in [t_0, t_f] \\ \dot{x}_{ext}(t) &= f_{ext}(x_{ext}(t), u_{ext}(t)), & t &\in [t_0, t_f] \end{aligned} \tag{1}$$

with control signals drawn from the set of measurable functions:

$$\begin{aligned} u_{int}(\cdot) &\in \mathbb{U}_{int} = \{\phi_{int} : [t_0, t_f] \to \mathscr{U}_{int}\}, \\ u_{ext}(\cdot) &\in \mathbb{U}_{ext} = \{\phi_{ext} : [t_0, t_f] \to \mathscr{U}_{ext}\}. \end{aligned} \tag{2}$$

where $\mathscr{U}_{int} \subset \mathbb{R}^{m_{int}}$ and $\mathscr{U}_{ext} \subset \mathbb{R}^{m_{ext}}$ are compact subsets of the input spaces $\mathbb{R}^{m_{int}}, \mathbb{R}^{m_{ext}}$.

We assume that the function representing the dynamics of the internal system, $f_{int} : \mathbb{R}^{n_{int}} \times \mathscr{U}_{int} \to \mathbb{R}^{n_{int}}$, is uniformly continuous, bounded, and Lipschitz continuous in its first argument. As such, for every initial external system state $x_{int0} \in \mathbb{R}^{n_{int}}$ and internal control signal $u_{int}(\cdot) \in \mathbb{U}_{int}$, then as stated in [8], there exists a unique trajectory $\xi_{int}(t; t_0, x_{int0}, u_{int}(\cdot))$ satisfying:

$$\begin{aligned} \frac{d}{dt}\xi_{int}(t; t_0, x_{int0}, u_{int}(\cdot)) = \\ f_{int}(\xi_{int}(t; t_0, x_{int0}, u_{int}(\cdot)), u_{int}(t)), \end{aligned} \tag{3}$$

$$\xi_{int}(t_0; t_0, x_{int0}, u_{int}(\cdot)) = x_{int0}. \tag{4}$$

Similarly, define the inverse trajectory of the internal system as follows. Let ξ_{int}^{-1} be the trajectory which propagates states forward in time starting from some state x_{int0}^{-1} at t_0 up until time t, using the *reverse-time dynamics* of the internal system:

$$\begin{aligned}\frac{d}{ds}\xi_{int}^{-1}(t;t_0,x_{int0}^{-1},u_{int}(\cdot)) = \\ -f_{int}(\xi_{int}^{-1}(t;t_0,x_{int0}^{-1},u_{int}(\cdot)),u_{int}(t)),\end{aligned} \tag{5}$$

$$\xi_{int}(t_0;t_0,x_{int0}^{-1},u_{int}(\cdot)) = x_{int0}^{-1}. \tag{6}$$

With a slight abuse of notation, x_{int0}^{-1} is used to indicate the initial state of the reverse-time trajectory mapping ξ_{int}^{-1}, and is not related to the variable x_{int0} in any direct way.

The same assumptions are made on the dynamics of the external system, such that state trajectories of the internal system are defined analogously of those of the internal system, and given by $\xi_{ext}(t;t_0,x_{ext0},u_{ext}(\cdot))$ and $\xi_{ext}^{-1}(t;t_0,x_{ext0}^{-1},u_{ext}(\cdot))$.

To formally define the impersistent observation model on the external system, we define Σ to be a σ-algebra on $[0,\infty)$, and let $\mathbf{T} \in \Sigma$ be a set of potentially open and infinite intervals on $[0,\infty)$, **unknown to the internal system**, which denote periods of time in which the internal system's observations of the external system are lost. Define

$$y_{int}(t) \in \mathbb{R}^{n_{int}} \times \{\mathbb{R}^{n_{ext}} \cup \varnothing\} \tag{7}$$

to be the observation of the internal system at time t, with $\varnothing$ representing the n_{ext}-dimensional empty value. The observation model is then given by

$$y_{int}(t) = \begin{cases} \begin{bmatrix} x_{int}^{\mathsf{T}}(t), & x_{ext}^{\mathsf{T}}(t) \end{bmatrix}^{\mathsf{T}} & t \notin \mathbf{T} \\ \begin{bmatrix} x_{int}^{\mathsf{T}}(t), & \varnothing^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} & t \in \mathbf{T} \end{cases} \tag{8}$$

Conversely, the observation model for the external system is simply given by

$$y_{ext}(t) = \begin{bmatrix} x_{int}^{\mathsf{T}}(t), & x_{ext}^{\mathsf{T}}(t) \end{bmatrix}^{\mathsf{T}}, \; \forall t \in [t_0,t_f]. \tag{9}$$

These definitions formalize the notion that for times $t \in \mathbf{T}$, the internal system can only observe its own state, and not the external state.

Finally, we define a quantitative notion of safety of the internal system, which is given by the function $d : \mathbb{R}^{n_{int}} \times \mathbb{R}^{n_{ext}} \to \mathbb{R}$. The joint internal and external state of the system at time t is defined to be safe (for the internal system) if

$$d(x_{int}(t),x_{ext}(t)) \geq 0. \tag{10}$$

Therefore, we can think of the function d as some signed distance to the set of unsafe system configurations.

The objective of the internal system is to remain safe for all $t \in [t_0,t_f]$ and all possible $\mathbf{T}$. We can, without loss of generality, assume that $\mathbf{T} = (t_0,t_f]$, since any control policy that is safe without observations of the external system will also be safe with those observations, whereas the converse is in general not true. Under this assumption, $\mathbf{T}$ does not include the point t_0, since the initial state of each systems is assumed to be known (within a known, bounded error tolerance) to both systems at t_0.

The assumption that $\mathbf{T} = (t_0,t_f]$ may appear overly-conservative, since in practice the internal system is at least intended to receive observations of the external system. However, if we wish the internal system to truly be robust to the possibility of lack of observations at any and all times, we must assume such is true in the analysis and design of our system. Therefore, this choice does not introduce additional conservatism into the formulation. During the actual execution of the system, any received observations should and will be used to allow proficient behavior of the system, which we discuss in Section II-A.

Given the assumptions on $\mathbf{T}$, the function indicating whether the initial internal and external system configurations will be safe for all time $[t_0,t_f]$ can be encoded in the following value function:

$$\begin{aligned} &V_{int}(x_{int0},x_{ext0},t_0,t_f) = \\ &\qquad \sup_{u_{int}(\cdot)\in\mathbb{U}_{int}} \; \inf_{u_{ext}(\cdot)\in\mathbb{U}_{ext}} \; \min_{t\in[t_0,t_f]} d(x_{int}(t),x_{ext}(t)) \\ &\qquad \text{s.t. } x_{int}(t) = \xi_{int}(t;t_0,x_{int0},u_{int}(\cdot)) \\ &\qquad\qquad x_{ext}(t) = \xi_{ext}(t;t_0,x_{ext0},u_{ext}(\cdot)) \end{aligned} \tag{11}$$

This function computes the smallest distance encountered to unsafe configurations along the worst-case trajectory of the unobserved external system, and best-effort trajectory of the internal system. A non-negative value function indicates that the internal system can remain safe, despite failure to observe the external system and its adversarial attempts to render the internal system unsafe. A negative value, meanwhile, indicates that there exists control actions by the external system such that no control signal of the internal system can maintain safety for all time, at least without observations of the external system. We wish to compute the super-zero-level set of (11), or equivalently compute the exact set of all safe initial configurations of the system.

Note that in (11), the optimization for both systems' control signals are *open-loop*, i.e. neither is an explicit function of the state of either system (besides the initial states, x_{int0} and x_{ext0}). This is because the internal system is assumed to be unable to observe the state of the external system for all times $t > t_0$, and therefore unable to react to its position during times of unobservability. Since the optimization over the external system control is chosen inside the optimization over internal control signals, this information structure does allow for the external control signal to be chosen knowing the position of the internal system. In this situation, there is no difference between an open-loop and closed-loop control signal for the external system. We note that in game-theoretic terms, this game can be described as an open-loop Stackelberg game, in which the internal system is the lead player.

Looking at the structure of (11), it is not obvious how to evaluate the value function for arbitrary inputs, let alone compute the super-zero-level set. This is mainly because it requires solving difficult nested optimizations for each evaluation. This difficulty exists even in offline settings in which real-time evaluation is not necessary. Unlike the case with *closed-loop* differential games, the game value function in this form cannot even be pre-computed using offline dynamic programming, due to the open-loop nature of the information structure. This is because, while the state of the external system is known to the internal system at time t_0, for all $t > t_0$, the state of external system is assumed to be unknown.

That being said, we will see that because the internal and external system have independent dynamics, (11) can be reformulated in a way that does allow for a dynamic programming solution, allowing its offline computation to leverage standard methods. This alternative formulation will rely on first computing the forward reachable set (FRS) of the external system (set of all states the external system can reach from its initial position), and then computing the backwards avoid tube for the internal system (set of all initial states of the internal system that can avoid the FRS of the external system). Using this method to compute the game value (11) offline, the value and a resultant control policy can be used in online, real-time execution of the system.

We first formally define the forward reach set of the external system, denoted $\mathbf{R}_{ext}(t,x_{ext0};t_0)$ for time $t \in [t_0,t_f]$. This is the set of all states x at time t that the external system can reach with admissible control signals, starting from x_{ext0} at time t_0. Equivalently, this is the set of all states x at time t_0 from which the inverse trajectory of the external system can reach x_{ext0} at time t_f.

$$\begin{aligned}&\mathbf{R}_{ext}(t,x_{ext0};t_0)\\ &\quad := \{x \in \mathbb{R}^{n_{ext}} : \exists\ u_{ext}(\cdot) \in \mathbb{U}_{ext}\\ &\qquad\qquad \text{s.t. } \xi_{ext}(t;t_0,x_{ext0},u_{ext}(\cdot)) = x\}\\ &\quad = \{x \in \mathbb{R}^{n_{ext}} : \exists\ u_{ext}(\cdot) \in \mathbb{U}_{ext}\\ &\qquad\qquad \text{s.t. } \xi_{ext}^{-1}(t;t_0,x,u_{ext}(\cdot)) = x_{ext0}\}\end{aligned} \tag{12}$$

Using the second definition, $\mathbf{R}_{ext}(t,x_{ext0};t_0)$ can easily be expressed as the sub-zero-level set of the following value function:

$$\begin{aligned}V_{\mathbf{R}_{ext}}(x,t,x_{ext0};t_0) := \inf_{u_{ext}(\cdot)\in\mathbb{U}_{ext}} \|z(t) - x_{ext0}\|\\ \text{s.t. } z(t) = \xi_{ext}^{-1}(t;t_0,x,u_{ext}(\cdot))\end{aligned} \tag{13}$$

Following immediately from the definition,

$$V_{\mathbf{R}_{ext}}(x,t_0,x_{ext0};t_0) = \|x - x_{ext0}\|. \tag{14}$$

The following theorem, stated and proved in [9], shows that $V_{\mathbf{R}_{ext}}(x,t,x_{ext0};t_0)$, and consequently $\mathbf{R}_{ext}(t,x_{ext0};t_0)$, can be numerically solved for all $t \in [t_0,t_f]$ using a variational Hamilton-Jacobi partial differential equation.

Theorem 1: Suppose $f_{ext} : \mathbb{R}^{n_{ext}} \times \mathscr{U}_{ext} \to \mathbb{R}^{n_{ext}}$, is uniformly continuous, bounded, and Lipschitz continuous in its first argument. Then the value function $V_{\mathbf{R}_{ext}}(x,t,x_{ext0};t_0)$ defined in (13) is the unique viscosity solution of the variational Hamilton-Jacobi PDE:

$$\begin{aligned}\frac{\partial V_{\mathbf{R}_{ext}}}{\partial t} + H = 0,\\ \forall t \in [t_0,t_f], x \in \mathbb{R}^{n_{ext}},\end{aligned} \tag{15}$$

with initial condition:

$$V_{\mathbf{R}_{ext}}(x,t_0,x_{ext0};t_0) = \|x - x_{ext0}\|, \tag{16}$$

where the Hamiltonian, H, is defined by:

$$H = \min_{u\in\mathscr{U}_{ext}} \frac{\partial \mathbf{V}_{R_{ext}}}{\partial x}(x,t,x_{ext0};t_0)\cdot(-f_{ext}(x,u)). \tag{17}$$

Proof: (see [9], Section II). ■

Therefore, this PDE can be solved numerically to obtain the set $\mathbf{R}_{ext}(t,x_{ext0};t_0)$, which can then be used to define a new, *pessimistic* safety function, which eliminates the need to know the exact state $x_{ext}(t)$.

$$\hat{d}(x_{int},t,x_{ext0};t_0) := \min_{x_{ext}\in\mathbf{R}_{ext}(t,x_{ext0};t_0)} d(x_{int},x_{ext}). \tag{18}$$

Observe that the safety of the internal system at time t can equivalently be expressed through the condition $\hat{d}(x_{int}(t),t,x_{ext0};t_0) \geq 0$. We note that this function in general may also be difficult to evaluate, however, as it too requires solving a potentially difficult optimization problem for each evaluation. By defining the set $\mathbf{D}$ to be the sub-zero-level set of the function $\hat{d}$, we can construct an alternative pessimistic safety function that will equivalently represent our safety criterion. First define the set

$$\begin{aligned}&\tilde{\mathbf{D}}(t,x_{ext0};t_0) \subset \mathbb{R}^{n_{int}} :=\\ &\qquad proj_{int}(\mathbf{D} \cap (\mathbf{R}_{ext}(t,x_{ext0};t_0) \times \mathbb{R}^{n_{int}})).\end{aligned} \tag{19}$$

Here $proj_{int}$ is the projection operator, projecting arguments into the state space of the internal system. In words, for a given time t, and initial state x_{ext0} at initial time t_0, the set $\tilde{\mathbf{D}}(t,x_{ext0};t_0)$ is the set of all states x_{int} such that there exists a x_{ext} reachable by the external system at time t, such that the pair (x_{int},x_{ext}) is unsafe. Now we define the alternate pessimistic safety function as the signed distance to this set:

$$\tilde{d}(x_{int},x_{ext0},t;t_0) := \begin{cases} \min_{x\in\tilde{\mathbf{D}}(t,x_{ext0};t_0)} \|x - x_{int}\| & x_{int} \notin \tilde{\mathbf{D}}\\ -\min_{x\notin\tilde{\mathbf{D}}(t,x_{ext0};t_0)} \|x - x_{int}\| & x_{int} \in \tilde{\mathbf{D}}\end{cases} \tag{20}$$

Using this alternate pessimistic safety function, we define a new value function,

$$\begin{aligned}&\tilde{V}_{int}(x_{int0},x_{ext0},t,t_0,t_f) =\\ &\qquad \sup_{u_{int}(\cdot)\in\mathbb{U}_{int}} \min_{s\in[t,t_f]} \tilde{d}(x_{int}(s),s,x_{ext0};t_0)\\ &\qquad \text{s.t. } x_{int}(s) = \xi_{int}(s;t,x_{int0},u_{int}(\cdot))\end{aligned} \tag{21}$$

Note that

$$\tilde{V}_{int}(x,x_{ext0},t_f,t_0,t_f) = \tilde{d}(x,t_f,x_{ext0};t_0). \tag{22}$$

Here, an extra argument on time is present that does not appear in the function V_{int} in (11). When evaluated at $t = t_0$, however, the value function $\tilde{V}_{int}$ has an equivalent zero-level set as V_{int}. Since we are ultimately interested in the super-zero-level set of V_{int}, we can equivalently work with $\tilde{V}_{int}$, which we will show can also be computed with known methods. Given the definition of this new value function, it becomes evident that the super-zero-level set, when evaluated at a particular x_{ext0}, is the set of states x_{int0} for which the internal system can avoid the set $\tilde{\mathbf{D}}(t, x_{ext0}; t_0)$, for all $t \in [t_0, t_f]$. This is precisely the definition of backwards avoid-tube of the set $\tilde{\mathbf{D}}(t, x_{ext0}; t_0)$, which we denote by $\mathbf{A}_{int}$:

$$\begin{aligned}&\mathbf{A}_{int}(t, \tilde{D}(\cdot,\cdot;\cdot), x_{ext0}; t_0, t_f) := \\ &\quad \{x \in \mathbb{R}^{n_{int}} : \exists u_{int}(\cdot) \in \mathbb{U}_{ext}, \\ &\qquad \text{s.t. } \xi_{int}(s; t, x, u_{int}(\cdot)) \notin \tilde{\mathbf{D}}(s, x_{ext0}; t_0),\ \forall s \in [t, t_f]\}.\end{aligned} \tag{23}$$

In this definition, the argument t is the time corresponding to the beginning of the internal system trajectory, where as the set $\tilde{\mathbf{D}}$ is always computed with respect to initial time t_0.

Similar to $V_{\mathbf{R}_{ext}}(x, t, x_{ext0}; t_0)$, the value function $\tilde{V}_{int}(x, x_{ext}, t, t_0, t_f)$ (for a particular x_{ext}, t_0, and t_f), and hence the avoid tube $\mathbf{A}_{int}(t, \tilde{\mathbf{D}}, x_{ext0}; t_0, t_f)$, can be solved using a HJ PDE, as demonstrated in [7]:

Theorem 2: Suppose $f_{int} : \mathbb{R}^{n_{int}} \times \mathscr{U}_{int} \to \mathbb{R}^{n_{int}}$, is uniformly continuous, bounded, and Lipschitz continuous in its first argument. Then the value function $\tilde{V}_{int}(x, x_{ext}, t, t_0, t_f)$ is the unique viscosity solution of the variational Hamilton-Jacobi PDE:

$$\begin{aligned}0 = \max\left\{\frac{\partial \tilde{V}_{int}}{\partial t} + H, \tilde{d}(x, t, x_{ext0}; t_0) - \tilde{V}_{int}(x, x_{ext}, t, t_0, t_f)\right\} \\ \forall t \in [t_0, t_f], x \in \mathbb{R}^{n_{int}},\end{aligned} \tag{24}$$

with terminal condition:

$$\tilde{V}_{int}(x, x_{ext0}, t_f, t_0, t_f) = \tilde{d}(x, t_f, x_{ext0}; t_0), \tag{25}$$

where the Hamiltonian is defined by:

$$H = \min_{u \in \mathscr{U}_{int}} \frac{\partial \tilde{V}_{int}}{\partial x}(x, x_{ext}, t, t_0, t_f) \cdot f_{int}(x, u). \tag{26}$$

Proof: (see [7], Theorem 1). ■

Therefore, by first solving the partial differential equation given by (15), (16), (17), and then the partial differential equation corresponding to (24), (25), (26), we can obtain a value function for which the super-zero-level set denotes the set of all safe states for the internal system, with the initial configuration of the external system parameterized by x_{ext0} and despite un-observability of the external system. We denote this set the *Eyes-Closed Safety Kernel* of the internal system:

$$\mathbf{E}(t, t_0, t_f) = \{(x_{int}, x_{ext}) : \tilde{V}_{int}(x_{int}, x_{ext}, t, t_0, t_f) \geq 0\} \tag{27}$$

Here we make an important note. In order to compute $\mathbf{E}(t, t_0, t_f)$, both of the PDEs must be solved for *all* x_{ext0} of interest. However, both PDEs can in general be solved for an entire set of x_{ext0} simultaneously. This formulation would simply result in PDEs which have state dimension equal to $2n_{ext}$ and $(n_{int} + n_{ext})$ for the first and second PDEs, respectively. The equations given in (15), (16), (17), (24), (25), and (26) are still valid in this case, as the states x_{ext0} do not evolve through time. We made explicit the computation of both PDEs as functions of a particular x_{ext0} since in practice, symmetries in the problem can be exploited to avoid computing the full PDEs, which can be expensive or even intractable to compute. We demonstrate this technique in the example in Section III.

A. System Execution

We stated that the result of the preceding calculations, the eyes-closed safety kernel, could be used for safe autonomous execution when the external system is unobserved as well as when it is observed. Recall that $\mathbf{T}$ (the intervals of time for which the external system observations are lost) is unknown to the internal system. However, we assume that the internal system is endowed with the ability to know at some $t \in [t_0, t_f]$ if $t \in \mathbf{T}$. In other words, the internal system is able to instantaneously detect that it has lost observations of the external system, through use of an anomaly detection module or by some other means.

Note that since the dynamics of both the internal and external systems do not depend on time, the value functions computed are temporal-shift invariant. This means that $\mathbf{E}(x_{int}, x_{ext}, 0, 0, t_f)$ provides a safe operating region in which the internal system can operate, guaranteeing the existence of a safe control signal for t_f seconds.

Over those t_f seconds, the optimal control for the internal system at state x_{int} at time t with last-known observation of the external system at x_{ext0} is given by

$$\begin{aligned}&u^*_{int}(x_{int}, t; x_{ext0}, t_f) = \\ &\qquad \arg\min_{u \in \mathscr{U}_{int}} \frac{d\tilde{V}_{int}}{dx_{int}}(x_{int}, x_{ext0}, t, 0, t_f) \cdot f_{int}(x_{int}, u)\end{aligned} \tag{28}$$

If at any time before the t_f seconds are expired, the internal system detects that $t \notin \mathbf{T}$ and observations of the external system are regained, then the control of the internal system can resume nominal control designed to satisfy some other objective, subject to remaining in the set $\mathbf{E}(0, 0, t_f)$. We note that by definition, there always exist control actions that keep the system within $\mathbf{E}$, regardless of the control of the external system. A practical way to ensure that the system never leaves the set is to monitor the value $\tilde{V}_{int}(x_{int}, x_{ext}, 0, 0, t_f)$ and switch to the control defined in (28) if the value drops within some tolerance above 0.

We further note that the guarantees provided on safety are only valid for t_f seconds after observations are lost. However, by taking t_f very large in the computation of (27), we can obtain guarantees for arbitrarily long intervals of time. In practice, although the set $\tilde{\mathbf{D}}(t_f, x_{ext0}; t_0)$ may grow unbounded with t_f, we find that $\mathbf{A}_{int}(t_0, \tilde{\mathbf{D}}, x_{ext0}; t_0, t_f)$ often converges to a fixed set. This will intuitively happen, for example, whenever the internal system is "faster" than the external system: if collision can be avoided early in the game, it can be avoided forever.

B. Note on Sub-Optimal Approximations

There may exist systems for which the decomposition of the internal and external systems is not possible, yet consideration of the loss of observation of the external states is still desired. In these cases, the method presented here does not directly apply, although modifications can be made to compute sub-optimal yet conservative approximations of the sets defined.

One such approximation is to fix a control policy for the internal system, so that the state trajectory of the internal system can be evaluated and incorporated into the objective of the external system. Then, computing the set of initial relative configurations of the internal and external system such that the external system can reach unsafe relative states, an *over-approximation of the unsafe relative initial configurations* is defined. By taking the complement of this set as the safe operating region for the internal system, the same guarantees on safety that are made in the exact method hold, although this approximation to the safe operating region may be significantly smaller or even empty compared to the true safe operating region.

III. Example

To demonstrate the method presented, we work through an example situation that is common to autonomous ground vehicles. The situation we wish to model is the interaction between a vehicle (internal system) and a dynamic obstacle (external system). As is typical for autonomous vehicles, the state of the external dynamic obstacle is estimated by the internal system using sensors. We assume that the internal system receives impersistent observations of the external system, which are exact within some bounded tolerance. This resembles a situation in which the internal system is using a data-driven algorithm to detect and track obstacles, but may encounter and detect out-of-distribution inputs, at which points in the perception module can not be trusted.

We assume for the sake of visualization that both the vehicle and dynamic obstacle evolve according to simple unicycle models, described in (30) and (31). Therefore, their state vectors are both 3-dimensional, making the sets defined in Section II easy to visualize. The dimension of the two systems need not be the same, and indeed, higher dimensional systems can be considered to represent more realistic systems (barring some computational limits as discussed in Section III-A).

The internal system is considered safe if and only if the two vehicles do not collide. For the sake of simplicity and readability, we assume that each car is circular with radii equal to r. Then the set of safe states is defined to be the super-zero-level set of the function

$$d(x_{int}, x_{ext}) := (x_1 - x_4)^2 + (x_2 - x_6)^2 - (2r)^2, \qquad (29)$$

where the dynamics of the internal system are given by

$$\begin{aligned} \dot{x}_1(t) &= u_1(t) \cdot cos(x_3(t)) \\ \dot{x}_2(t) &= u_1(t) \cdot sin(x_3(t)) \\ \dot{x}_3(t) &= u_2(t), \end{aligned} \qquad (30)$$

and the dynamics of the dynamic obstacle (external system) are given by

$$\begin{aligned} \dot{x}_4(t) &= u_3(t) \cdot cos(x_6(t)) \\ \dot{x}_5(t) &= u_3(t) \cdot sin(x_6(t)) \\ \dot{x}_6(t) &= u_4(t). \end{aligned} \qquad (31)$$

We assume that both agents are limited by the control bounds,

$$\begin{aligned} u_1 &\in [0,4], \quad u_2 \in [-1.0, 1.0] \\ u_3 &\in [0,3], \quad u_4 \in [-0.75, 0.75]. \end{aligned} \qquad (32)$$

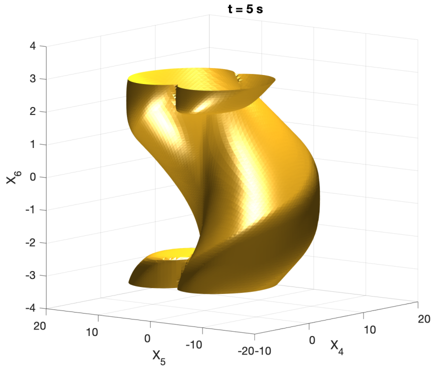

Fig. 2. Visualization of the set $\mathbf{R}_{ext}(5.0, \mathbf{0}; 0)$. The states inside the yellow volume are all the possible states the external system can reach in 5.0 seconds.

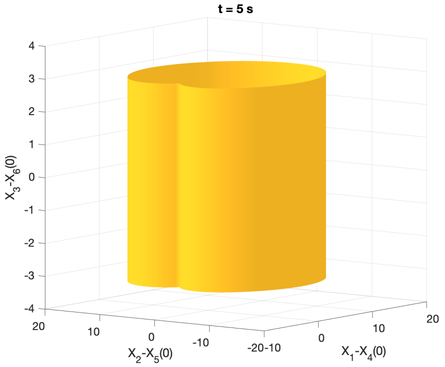

Fig. 3. Visualization of the set $\tilde{\mathbf{D}}(5.0, \mathbf{0}; 0)$. The states inside the yellow volume are all of the states of the internal system, relative to the last known position of the external system (x_{ext0}), such that collision at the current time is possible. Note the invariance of the set with respect to the relative heading dimension ($x_3 - x_6(0)$). This is because collision should be avoided, irrespective of the heading of the two sub-systems.

We solve for the sets $\mathbf{R}_{ext}(t, x_{ext0}; 0)$, $\tilde{\mathbf{D}}(t, x_{ext0}; 0)$, and $\mathbf{A}_{int}(t, \tilde{\mathbf{D}}, x_{ext0}; t_0, t_f)$, defined in Section II. We assume without loss of generality that $t_0 = 0$, and solve for $t_f = 5.0$. We find that this time horizon is sufficient for the set $\mathbf{A}_{int}$ to converge to within an acceptably small tolerance.

To solve for the set $\mathbf{R}_{ext}(t, x_{ext0}; 0)$ we use the Level Set Toolbox [10], a Matlab [11] software package for solving partial differential equations using the level set method [12]. This set is visualized in Figure 2 for $t = 5.0$, and $x_{ext0} = \mathbf{0}$, where $\mathbf{0}$ is the zero vector. To compute the set $\tilde{\mathbf{D}}$, we first observe that the forward reachable set of the external system is invariant with respect to the initial condition x_{ext0}, up to a rigid transformation (rotation and translation). Therefore, we can solve for $\mathbf{R}_{ext}(t, \mathbf{0}; 0)$, and then compute the sets $\tilde{\mathbf{D}}(t, x_{ext0}; 0)$ and $\mathbf{A}_{int}(t, \tilde{\mathbf{D}}, x_{ext0}; 0, 5.0)$ in relative coordinates $x_1 - x_4(0)$, $x_2 - x_5(0)$, and $x_3 - x_6(0)$. Here, we have avoided having to compute the sets for all possible values of x_{ext0}, as was discussed in Section II.

To see how this is done, first consider $\tilde{\mathbf{D}}(t, \mathbf{0}; 0)$. We project the set $\mathbf{R}_{ext}(t, \mathbf{0}; 0)$ on to the (x_4, x_5) plane, relabel the

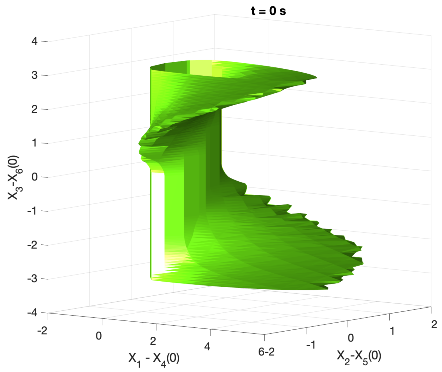

Fig. 4. Visualization of the set $\mathbf{A}_{int}(0,\tilde{\mathbf{D}},\mathbf{0};0,5.0)$. The states inside the green volume are the initial states of the internal system, relative to the initial state of the external system, that must be avoided in order to guarantee collision is avoided for t_f seconds, even in the event that observations of the external system are lost at any time.

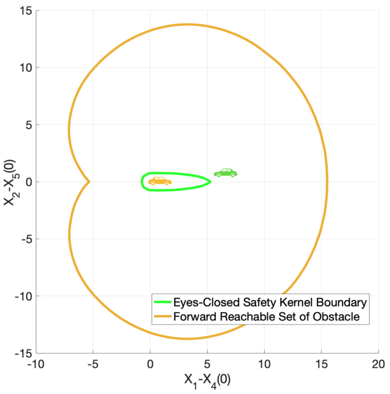

Fig. 5. Visualization of the Eyes-Closed Safety Kernel, $\mathbf{A}_{int}(0,\tilde{\mathbf{D}},\mathbf{0};0,5.0)$, and the Forward Reachable Set of the obstacle, $\tilde{\mathbf{D}}(5.0,\mathbf{0};0)$, for the slice $x_3 - x_6(0) = \pi$. Here we see that despite the green vehicle (internal system) starting within the FRS of the orange vehicle (external system), the green vehicle lies within the Eyes-Closed Safety Kernel. This means that the dynamics of the combined system are such that there exists a maneuver by the green vehicle which can evade the orange vehicle, despite losing observability.

coordinates from $x_4 \to (x_1 - x_4(0))$, $x_5 \to (x_2 - x_5(0))$, and $x_6 \to (x_3 - x_6(0))$, take the Minkowski sum of the projection with a circle of radius $2r$ (to account for the geometry of both systems), and then extrude the set in the $(x_3 - x_6(0))$ dimension. The result of these operations are visualized in Figure 3 for $t = 5.0$. Finally, we compute the time-varying backwards avoid-tube of $\tilde{\mathbf{D}}(t,\mathbf{0};0)$, denoted $\mathbf{A}_{int}(t,\tilde{\mathbf{D}},\mathbf{0};0,5.0)$. This set is computed again using the Level Set Toolbox, and is displayed in Figure 4 at $t = 0$.

The set $\mathbf{A}_{int}(0,\tilde{\mathbf{D}},\mathbf{0};0,5.0)$ is intuitive to interpret. The super-zero-level set (free space outside the green volume in Figure 4) is the set of all initial configurations of the internal and external systems, such that the internal system can escape collision despite adversarial control by the external system and without observations of the external system. The regions that are unable to avoid collision are those in which the initial configuration of the two systems are either already in collision, or those in which the internal system is initially headed towards the external system, and cannot turn away in time due to the limited control authority on u_2 (the set wraps at $(x_3 - x_6(0)) = \pm\pi$ due to the periodicity of the relative heading). Because the bounds on u_1 are greater than those on u_3, the internal system can escape collision from all other initial configurations.

While the result of the computations thus far in this example are sufficient to synthesize a controller which is able to avoid obstacles after observations are lost, it has been assumed that the obstacle to be avoided was observed before observations were lost. However, in dynamic environments, there are often situations in which no obstacles might be detected, such as when an obstacle is beyond the sensing range of the ego vehicle. Such a situation can also be accounted for using the method presented here. Specifically for this example, we first compute the set $\mathbf{R}_{ext}(t,\mathbf{S}_{sense};0)$, which is defined to be the forward reachable set of the external system when starting from any state beyond the ego-vehicle's sensing radius (relative to the origin):

$$\mathbf{S}_{sense} := \left\{ \begin{bmatrix} x_4 & x_5 & x_6 \end{bmatrix}^\intercal : x_4^2 + x_5^2 \geq r_{sense}^2 \right\}. \tag{33}$$

Here r_{sense} is the sensing radius.

Code for generating the sets presented in this example is provided at https://www.github.com/4estlaine/eyes_closed.

A. Notes on Computation

At the root of this method is the offline numerical computation of two partial differential equations. Numerical computation of PDEs without special structure requires defining a grid over the state space and repeatedly computing quantities at every grid point; the computation time thus scales exponentially with the state space dimension [9]. In practice, we are limited in the dimension of systems for which we can compute these safety guarantees by the computation required to do so. Nevertheless, with clever representation of the systems, many interesting problems more complicated than the one presented here can be solved using this framework. For example, a parameterized lane boundary can be included in the representation of the internal system, increasing the system dimension by 3, but allowing for avoiding lane-boundary departures despite loss of observability. Code for this example is also included in the above-linked repository.

Another noteworthy fact is that space discretization can introduce errors in the computation of the value functions of interest, although the discretization error tends to zero as bin size tends to zero [13]. Some artifacts from the discretization error can be seen in Figure 4, particularly in places where the set *creases*. The possibility that these artifacts may appear should be taken into account when constructing safety arguments for systems using this method.

The computational burden at run-time is minimal, and requires simply looking up the pre-computed value and corresponding maximizing control from the grid, which is indexed by the configuration of the internal and external systems. The time taken by this operation is on the order of milliseconds.

IV. Related Work

The framework and method presented here stems most directly from the large body of work which has leveraged the Hamilton-Jacobi (HJ) equations to transform optimal control problems into numerically-solvable partial differential equations. An overview of HJ optimal control formulations is given in [9]. The Hamilton-Jacobi-Isaacs (HJI) equations similarly allow for transforming fully observable differential games into partial differential equations [14] [13] [7]. The work presented in this article is a differential game of sorts, although due to the impersistent observability of the external system, the standard HJI pursuit-evasion formulation does not apply, necessitating the method we have developed. The set of all safe initial configurations in the persistently observed case is a strict super-set of the eyes-closed safety kernel defined here.

There are recent works that have leveraged HJ and HJI equations to capture other interesting information patterns arising in problems common to autonomous systems. For example, in [15], [16] and [17], a dynamic game is formulated to capture the mismatch between a motion planning model and a high-fidelity model of the system attempting to track the plans. This allows for planning trajectories that are guaranteed to be trackable by the high-fidelity model. These works are extended in [18] and [19], allowing for limited sensing radii. Meanwhile, [20] leverages a similar idea to generate trajectories that incorporate system tracking error and will not have any at-fault collisions in unforeseen environments with limited sensing radius. All of these methods assume that the internal system has persistent observations of all obstacles, at least within some sensing radius. The methods developed in these works do not apply when observation is lost.

Although we leverage exact reachable set computations to formulate our problem, techniques other than solving HJ PDEs can be used to find them. For example, [21], [22] and [23] use sampled particles to approximate such sets. In the latter two works, approximate reachable sets are used to generate safe trajectories in the context of occluded obstacles.

The formulation of avoiding occluded obstacles is indeed similar in nature to that presented here. There are a handful of recently-proposed methods designed to tackle the occlusion problem. For example, [24] leverages over-approximations of the forward-reachable sets of (potentially occluded) obstacles is used when planning trajectories. [25] handles the problem in a multi-object tracking formulation, which in general can be leveraged to handle similar problems, although in a probabilistic and inexact manner. See, for example, [26] for a survey of such methods.

Other works involve different considerations of the perception system in autonomous vehicles. In [27], for example, the authors consider the role of perception latency in safe avoidance of obstacles. [28] generates a fail-safe control trajectory that can be executed in case of emergencies such as loss of sensors, although that formulation relies on linear system dynamics. In [29], the authors attempt to learn a system that simultaneously detects perception failures and generates corrective action. However, that method does not include any formal safety analysis of the control actions that are generated. Recently, [30] formulated a robust control problem accounting for a learned perception module, although there the analysis requires that the dynamics of the autonomous system is a linear, time invariant ordinary differential equation.

Of course, many works have considered obstacle avoidance in the presence of noisy or uncertain measurements [31] [32] [33]. Some have tackled this problem in the context of formal methods, providing formal guarantees of safety in such observation models, as in [34]. However, again, consideration of noisy or purely partial observations is a different from the problem we address in this paper, and those works should be taken as complementary to the work presented here.

V. Conclusion

We present a framework for optimally accounting for detected loss of observations of autonomous systems. We show that in order to be robust to the situation defined, the safety controller of the internal system must be designed assuming it will lose and never regain observations of the external system. Thus, the internal system must always stay within a set of relative positions with respect to the external system, to ensure that when observations are lost, there still exists a safe maneuver that avoids all possible trajectories of the external system.

We formulate this problem as an open-loop differential Stackelberg game, and recast it in a manner that allows its solution to be found by solving two related partial differential equations. This method results in a safe operating region for the autonomous system; here, we call it the Eyes-Closed Safety Kernel. If the autonomous system is able to detect instantaneously that it has lost observations, then by staying within the Eyes-Closed Safety Kernel for all time, and executing the control (28) whenever observations are lost, the system is guaranteed to maintain safety as long as the system obeys the modeling assumptions made.

Although the detection of perception failure is a very challenging problem in practice, the framework presented here demonstrates that it is possible to safely account for these failures if they can be detected. This method therefore motivates further research on developing better methods for predicting when perception modules are incorrect, and can already be used with existing methods for predicting perception failures. Even if not every perception failure is detected, the framework we present provides a means for guaranteeing robustness to those that are predicted.

References

[1] S. Balakrishnan, L. Wasserman, *et al.*, “Hypothesis testing for densities and high-dimensional multinomials: Sharp local minimax rates,” *The Annals of Statistics*, vol. 47, no. 4, pp. 1893–1927, 2019.

[2] S. Liang, Y. Li, and R. Srikant, “Enhancing the reliability of out-of-distribution image detection in neural networks,” *arXiv preprint arXiv:1706.02690*, 2017.

[3] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *arXiv preprint arXiv:1610.02136*, 2016.

[4] D. Hendrycks, M. Mazeika, and T. G. Dietterich, "Deep anomaly detection with outlier exposure," *arXiv preprint arXiv:1812.04606*, 2018.

[5] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[6] H. Choi and E. Jang, "Generative ensembles for robust anomaly detection," *arXiv preprint arXiv:1810.01392*, 2018.

[7] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry, "Reach-avoid problems with time-varying dynamics, targets and constraints," in *Int. Conf. on hybrid systems: computation and control*. ACM, 2015.

[8] E. A. Coddington and N. Levinson, *Theory of ordinary differential equations*. McGraw-Hill, 1955.

[9] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-Jacobi reachability: A brief overview and recent advances," in *Conf. on Decision and Control*. IEEE, 2017.

[10] I. M. Mitchell and J. A. Templeton, "A toolbox of hamilton-jacobi solvers for analysis of nondeterministic continuous and hybrid systems," in *International Workshop on Hybrid Systems: Computation and Control*. Springer, 2005, pp. 480–494.

[11] MATLAB, *version 9.5.0.94444 (R2018b)*. Natick, Massachusetts: The MathWorks Inc., 2018.

[12] J. A. Sethian, *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Cambridge university press, 1999, vol. 3.

[13] L. Evans, *Partial Differential Equations*. Amer. Math. Soc., 2010.

[14] L. C. Evans and P. E. Souganidis, "Differential games and representation formulas for solutions of hamilton-jacobi-isaacs equations," *Indiana University mathematics journal*, vol. 33, no. 5, pp. 773–797, 1984.

[15] S. L. Herbert*, M. Chen*, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, "Fastrack: a modular framework for fast and guaranteed safe motion planning," *IEEE Conf. on Decision and Control*, 2017.

[16] S. Kousik, S. Vaskov, M. Johnson-Roberson, and R. Vasudevan, "Safe trajectory synthesis for autonomous driving in unforeseen environments," *ArXiV*, p. V001T44A005, 10 2017.

[17] S. Kousik, P. Holmes, and R. Vasudevan, "Safe, aggressive quadrotor flight via reachability-based trajectory design," *ArXiV*, 04 2019.

[18] D. Fridovich-Keil*, S. L. Herbert*, J. F. Fisac*, S. Deglurkar, and C. J. Tomlin, "Planning, fast and slow: A framework for adaptive real-time safe trajectory planning." *IEEE Conf. on Robotics and Automation*, 2018.

[19] S. Kousik, S. Vaskov, F. Bu, M. Johnson-Roberson, and R. Vasudevan, "Bridging the gap between safety and real-time performance in receding-horizon trajectory design for mobile robots," *arXiv*, 2018.

[20] S. Vaskov, S. Kousik, H. Larson, F. Bu, J. Ward, S. Worrall, M. Johnson-Roberson, and R. Vasudevan, "Towards provably not-at-fault control of autonomous robots in arbitrary dynamic environments," *Robotics: Science and Systems*, 02 2019.

[21] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *Journal of guidance, control, and dynamics*, vol. 25, no. 1, pp. 116–129, 2002.

[22] M.-Y. Yu, R. Vasudevan, and M. Johnson-Roberson, "Occlusion-aware risk assessment for autonomous driving in urban environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2235–2241, 2019.

[23] ——, "Risk assessment and planning with bidirectional reachability for autonomous driving," *arXiv preprint arXiv:1909.08059*, 2019.

[24] P. F. Orzechowski, A. Meyer, and M. Lauer, "Tackling occlusions & limited sensor range with set-based safety verification," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1729–1736.

[25] E. Galceran, E. Olson, and R. M. Eustice, "Augmented vehicle tracking under occlusions for decision-making in autonomous driving," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3559–3565.

[26] K. Granstrom, M. Baum, and S. Reuter, "Extended object tracking: Introduction, overview and applications," *arXiv preprint arXiv:1604.00970*, 2016.

[27] D. Falanga, S. Kim, and D. Scaramuzza, "How fast is too fast? the role of perception latency in high-speed sense and avoid," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1884–1891, April 2019.

[28] C. Pek and M. Althoff, "Computationally efficient fail-safe trajectory planning for self-driving vehicles using convex optimization," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1447–1454.

[29] D. M. Saxena, V. Kurtz, and M. Hebert, "Learning robust failure response for autonomous vision based flight," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5824–5829, 2017.

[30] S. Dean, N. Matni, B. Recht, and V. Ye, "Robust guarantees for perception-based control," *arXiv preprint arXiv:1907.03680*, 2019.

[31] N. E. Du Toit and J. W. Burdick, "Probabilistic collision checking with chance constraints," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 809–815, 2011.

[32] D. Althoff, J. J. Kuffner, D. Wollherr, and M. Buss, "Safety assessment of robot trajectories for navigation in uncertain and dynamic environments," *Autonomous Robots*, vol. 32, no. 3, pp. 285–302, 2012.

[33] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone,

"Multimodal probabilistic model-based planning for human-robot interaction," *arXiv*, 2017.

[34] S. Mitsch and A. Platzer, "Verified runtime model validation for partially observable hybrid systems," *arXiv preprint arXiv:1811.06502*, 2018.

[35] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control*, 2005.