Enforcing Safety at Runtime for Systems with Disturbances

Matthew Abate and Samuel Coogan

Abstract—Safety for control systems is often posed as an invariance constraint; the system is said to be safe if state trajectories avoid some unsafe region of the state space for all time. An assured controller is one that enforces safety online by filtering a desired control input at runtime, and control barrier functions (CBFs) provide an assured controller that renders a safe subset of the state-space forward invariant. Recent extensions propose CBF-based assured controllers that allow the system to leave a known safe set so long as a given backup control strategy eventually returns to the safe set, however, these methods have yet to be extended to consider systems subjected to unknown disturbance inputs.

In this work, we present a problem formulation for CBF-based runtime assurance for systems with disturbances, and controllers which solve this problem must, in some way, incorporate the online computation of reachable sets. In general, computing reachable sets in the presence of disturbances is computationally costly and cannot be directly incorporated in a CBF framework. To that end, we present a particular solution to the problem, whereby reachable sets are approximated via the mixed-monotonicity property. Efficient algorithms exist for overapproximating reachable sets for mixed-monotone systems with hyperrectangles, and we show that such approximations are suitable for incorporating into a CBF-based runtime assurance framework.

I. INTRODUCTION

Controllers whose safety guarantees are derived through the online enforcement of constraints, rather than a priori verification, are referred to in literature as runtime assurance architectures [1] or active set invariance filters (ASIF) [2], [3]. In this setting, system safety is posed as an invariance constraint, requiring that a system avoid some unsafe region of the state space for all time. Specifications of this class are often used to describe real-world safety specifications due to the fact that the definition of real-world safety often is presented as the ability to avoid unsafe scenarios during deployment.

Numerous mechanisms exist for enforcing invariance constraints, and in particular, control barrier functions (CBFs) are well suited for this task. CBF-based runtime assurance architectures modify a suggested desired input at runtime to create a safe forward invariant region in the state space. This is a main idea of [4], [5] where the resulting controller is formulated as a quadratic program for systems with no disturbances, and this idea is extended in [6] to the setting with disturbances. A limitation here is the need to verify a controlled forward invariant region a priori and in general this region should be large; this problem can also be formulated as the search for a backup strategy with a corresponding controlled forward invariant region [7], [8]. The authors of [2], [3] present a CBF-based runtime assurance architecture, here formed via a verified backup strategy and safe region, which allows the system to leave the safe region. The method eases the problem of verifying a forward invariant region a priori, however, these works do not consider systems with disturbances. In this work we present a problem formulation for CBF-based runtime assurance for controlled dynamical systems with disturbances, and we present an example solution to this problem where nondeterminism in the system model is assessed via the mixed-monotonicity property.

Mixed-monotone systems are separable via a decomposition function into increasing and decreasing components and this enables the approximation of reachable sets [9]–[11] and the identification of attractive and forward invariant sets [10]; a similar approach is first pioneered in [12], and we refer the reader also to [13], [14] for fundamental results on monotone dynamical systems.

Efficient algorithms exist for overapproximating reachable sets for mixed-monotone systems with hyperrectangles, and we show that such approximations are suitable for incorporating into a CBF-based runtime assurance framework.

In summary, the main contribution of this work are (a) we present a problem formulation for CBF-based runtime assurance for control systems with disturbances, and (b) we present a specific solution to the problem statement, whereby the nondeterminism in the system model is assessed through mixed-monotonicity based reachability methods.

This paper is structured as follows. We present our notation in Section II. In Section III we recall preliminary results on CBFs, and we also present a problem formulation for CBF-based runtime assurance for control systems with
disturbances. Throughout the remainder of the work, we present a solution to the problem statement, which relies on mixed-monotonicity based reachability methods. To that end, we present preliminary results on mixed-monotone systems in Section IV, and we present an assured controller architecture in Section V which solves the problem statement and which accommodates nondeterminism in the system model via the mixed-monotonicity property. We present a numerical example in Section VI, where we design and implement a runtime assurance architecture to enforce an interagent distance constraint on a platoon of vehicles.

II. NOTATION

We denote vector entries via subscript, i.e., \( x_i \) for \( i \in \{1, \ldots, n\} \) denotes the \( i \)th entry of \( x \in \mathbb{R}^n \), and we denote the empty set by \( \emptyset \) := \{\}.

Given \( x, y \in \mathbb{R}^n \) with \( x_i \leq y_i \) for all \( i \),

\[
[x, y] := \{ z \in \mathbb{R}^n \mid x_i \leq z_i \leq y_i \text{ for all } i \}
\]

denotes the hyperrectangle with endpoints \( x \) and \( y \), and

\[
\langle x, y \rangle := \{ z \in \mathbb{R}^n \mid z_i \in [x_i, y_i] \text{ for all } i \}
\]
denotes the finite set of \( 2^n \) vertices of \( [x, y] \). We also allow \( x_i \in \mathbb{R} \cup \{-\infty\} \) and \( y_i \in \mathbb{R} \cup \{\infty\} \) so that \( [x, y] \) defines an extended hyperrectangle, that is, a hyperrectangle with possibly infinite extent in some coordinates.

Let \( (x, y) \) denote the vector concatenation of \( x, y \in \mathbb{R}^n \), i.e., \( (x, y) := [x^T y^T]^T \in \mathbb{R}^{2n} \).

Given \( a = (x, y) \in \mathbb{R}^{2n} \) with \( x_i \leq y_i \) for all \( i \), we denote by \( \langle a \rangle \) the hyperrectangle formed by the first and last \( n \) components of \( x \), i.e., \( \langle a \rangle := [x, y] \), and similarly \( \langle a \rangle := \langle x, y \rangle \).

III. RUNTIME ASSURANCE FOR NONDETERMINISTIC SYSTEMS

In this section, we define the problem of runtime assurance for continuous-time nondeterministic systems and provide a discussion on the problem statement.

A. Problem Setting

We consider controlled dynamical systems with disturbances of the form

\[
\dot{x} = f(x) + g_1(x)u + g_2(x)w \tag{1}
\]

with state \( x \in \mathcal{X} \subseteq \mathbb{R}^n \), control input \( u \in \mathbb{R}^m \), and Lipschitz continuous disturbance input \( w \in \mathcal{W} \subseteq \mathbb{R}^n \). If \( \mathcal{W} \) is a singleton set—equivalently, if the term \( g_2(x)w \) is omitted from (1)—then the system is said to be deterministic; otherwise, the system is said to be nondeterministic.

We let \( \Phi(T; x, u, w) \) denote the state of (1) at time \( T \geq 0 \), when starting from an initial state \( x \in \mathcal{X} \) at time 0 and evolving subject to a feedback controller \( u : \mathcal{X} \to \mathbb{R}^m \) and the disturbance signal \( w : [0, T] \to \mathcal{W} \).

**Assumption 1.** We associate the system (1) with an unsafe subset of the system statespace \( \mathcal{X}_u \subset \mathcal{X} \).

A control policy is safe if it avoids the unsafe set as formalized next.

**Definition 1.** A controller \( u : \mathcal{X} \to \mathbb{R}^m \) is safe with respect to state \( x \in \mathcal{X} \) if \( \Phi(T; x, u, w) \in \mathcal{X} \setminus \mathcal{X}_u \) for all \( T \geq 0 \) and for all \( w : [0, T] \to \mathcal{W} \). We extend this notation to sets so that \( u \) is safe with respect to set \( \mathcal{S} \subset \mathcal{X} \) if \( u \) is safe with respect to \( x \) for all \( x \in \mathcal{S} \).

One way to establish safety is through invariance.

**Definition 2.** Given a controller \( u \), a set \( \mathcal{S} \subset \mathcal{X} \) is robustly forward invariant for (1) under \( u \) if \( \Phi(T; x, u, w) \in \mathcal{S} \) for all \( x \in \mathcal{S} \), all \( T \geq 0 \) and all Lipschitz continuous disturbance inputs \( w : [0, T] \to \mathcal{W} \).

**Remark 1.** It is immediate that if \( \mathcal{S} \) is robustly forward invariant for (1) under some control policy \( u \), then \( u \) is safe with respect to \( \mathcal{S} \).

Suppose \( \mathcal{S} = \{ x \in \mathcal{X} \mid h(x) \geq 0 \} \subset \mathcal{X} \setminus \mathcal{X}_u \) for some continuously differentiable \( h : \mathbb{R}^n \to \mathbb{R} \) and consider the pointwise-defined controller

\[
uCBF(x) = \arg \min_{u \in \mathbb{R}^m} \|u - \nuC(x)\|_2^2 \tag{2}
\]

s.t.

\[
\frac{\partial h}{\partial x}(f(x) + g_1(x)u + g_2(x)w) \geq -\alpha(h(x)) \tag{3}
\]

\[\forall w \in \mathcal{W}\]

where \( \alpha : \mathbb{R} \to \mathbb{R} \) is a given locally Lipschitz class-\( K \) function and \( \nuC(x) \) is some given controller. Provided the set of \( u \) satisfying the constraint (3) is nonempty for all \( x \), then \( \mathcal{S} \) is robustly forward invariant for (1) and \( \nuCBF \) is safe with respect to \( \mathcal{S} \) from Remark 1 and this statement is true even when \( \nuC \) is not safe with respect to \( \mathcal{S} \). In this instance, \( h \) is said to be a control barrier function (CBF) for \( \nuCBF \) as developed in [4]. The fundamental idea of the CBF formulation is that system safety is assured online by solving (2)–(3) to ensure \( \mathcal{S} \) is robustly forward invariant. Note that, as formulated, for each \( x \), (2)–(3) is a quadratic program with linear constraints, although there are potentially infinite constraints since (3) must hold for all \( w \in \mathcal{W} \). However, in certain cases, it is possible to exchange (3) for a finite number of constraints. For example, if \( \mathcal{W} \) is a polytope, as is the case below, then (3) need only be verified at the vertices of \( \mathcal{W} \) since the constraint is affine in \( w \).

Applying \( \nuCBF \) from (2)–(3) has added benefits beyond system safety and, in particular, \( \nuCBF \) will evaluate to \( \nuC \) whenever possible; thus, if \( \nuC \) has performance advantages over \( u \), then \( \nuCBF \) will retain these advantages.

It is the primary focus of this paper to design safe controllers for the system (1). To that end, we assume knowledge of a backup controller which is safe with respect to some subset of the statespace by virtue of a robustly invariant backup region as defined next.

**Definition 3.** The pair \( (u^b, \mathcal{S}_b) \) with \( u^b : \mathcal{X} \to \mathbb{R}^m \) and \( \mathcal{S}_b = \{ x \in \mathcal{X} \mid h(x) \geq 0 \} \subset \mathcal{X} \) for a continuously differentiable \( h : \mathcal{X} \to \mathbb{R} \) is a backup control policy if:

1. \( \mathcal{S}_b \) is compact and \( \mathcal{S}_b \cap \mathcal{X}_u = \emptyset \),
2. \( h \) is concave on \( \mathcal{X} \),
3. \( \frac{\partial h}{\partial x} \neq 0 \) on the boundary of \( \mathcal{S}_b \), and
4) there exists a class-$K$ function $\alpha : \mathbb{R} \to \mathbb{R}$ such that
\[
\frac{\partial h}{\partial x}(x) \left( f(x) + g_1(x)u^b(x) + g_2(x)w \right) \geq -\alpha(h(x))
\]
for all $x \in S_b$ and for all $w \in W$.

In particular, the last condition above implies $S_b$ is robustly forward invariant for $\hat{(1)}$ under $u^b$ via the CBF conditions discussed above and therefore $u^b$ is safe with respect to $S_b$ by virtue of the first condition [6]. In this case, $u^b$ is called a backup controller and $S_b$ is its backup region.

While applying the backup controller ensures system safety, there are two primary reasons why applying such a policy is generally not preferable:

1) Backup controllers are typically designed without considering performance objectives. In particular, another controller may exist which ensures safety and satisfies some performance objective.

2) $S_b$ may not be well-developed, i.e., $u^b$ may be safe with respect to a set larger than $S_b$, and it is possible that $S_b$ is too conservative to satisfy certain performance objectives.

We have already discussed how CBFs provide a solution to the first problem via, e.g., the controller $\hat{\hat{(4)}}$, in which knowledge of $u^b$ is not even needed; see [4], [6] for further details. However, traditional CBF based controllers are still subject to the limitations of the second problem. A solution to the second problem is presented in [2], [3] for deterministic systems, where the authors effectively increase the size of the safe region through the use of look-ahead methods.

We now have the necessary tools to define the problem of runtime assurance for nondeterministic control systems.

**Problem Statement** (Runtime Assurance for Nondeterministic Control Systems). Assume a system of form $\hat{(1)}$ and a set of unsafe states $X_u \subset X$. Additionally, assume a backup control policy $(u^b, S_b)$, and assume a desired controller $u^d$ which satisfies some performance objective but is perhaps not safe with respect to $S_b$. The objective is to design a controller $u^{\text{ASIF}}$ such that $u^{\text{ASIF}}$ is safe with respect to $S_b$ and such that $u^{\text{ASIF}}(x)$ evaluates to $u^d(x)$ when it is safe to do so.

A controller $u^{\text{ASIF}}$ which solves the problem statement is referred to as an assured controller or an active set invariance filter (ASIF).

**B. Discussion**

Note that the backup control policy itself is an assured controller when the performance control objectives are disregarded, i.e. letting $u^{\text{ASIF}}(x) = u^b(x)$ for all $x \in X$ we have that $u^{\text{ASIF}}$ is safe with respect to $S_b$. When performance control objectives are considered, one must incorporate the desired controller $u^d$ in the ASIF formulation. As such, the problem statement can be thought of as the task of integrating a backup strategy in an existing, perhaps unsafe, desired controller.

We particularly aim for a solution that provides an assured controller that need not render $S_b$ forward invariant; it may be the case, for instance, that for certain initial conditions $x \in S_b$, the system $\hat{(1)}$ will be driven out of $S_b$ by $u^{\text{ASIF}}$ and may not return. Nonetheless, by virtue of the fact that $u^{\text{ASIF}}$ is an assured controller we have that $u^{\text{ASIF}}$ is safe with respect to $S_b$, and, optimistically, it may be the case that $u^{\text{ASIF}}$ is safe with respect to certain states outside of $S_b$.

In Section [V] we present a solution to the problem statement which allows the system to leave the, perhaps conservative, safe set $S_b$. In our proposed solution, we specifically address nondeterminism in the system model through mixed-monotonicity based reachability methods.

**IV. PRELIMINARIES ON MIXED-MONOTONE SYSTEMS**

Before visiting the general setting of $\hat{(1)}$, we first consider the nondeterministic autonomous system
\[
\dot{x} = F(x, w)
\]
and recall fundamental results in mixed-monotonicity theory. As before, we let $X$ and $W$ denote the state and disturbance spaces of $\hat{(5)}$, respectively, where we now assume $X$ is an extended hyperrectangle and $W$ is a hyperrectangle, with $W := [w, \bar{w}]$ for $w, \bar{w} \in \mathbb{R}^m$ and $w_i \leq \bar{w}_i$ for all $i$.

**Definition 4.** Given a locally Lipschitz continuous function $d : X \times W \times X \times W \to \mathbb{R}^n$, the system $\hat{(5)}$ is mixed-monotone with respect to $d$ if all of the following hold:

- For all $x \in X$ and all $w \in W$, $d(x, w, x, w) = F(x, w)$.
- For all $i, j \in \{1, \ldots, n\}$ with $i \neq j$, $\frac{\partial d_i}{\partial x_j}(x, w, \tilde{x}, \tilde{w}) \geq 0$ for all $x, \tilde{x} \in X$ and all $w, \tilde{w} \in W$ whenever the derivative exists.
- For all $i, j \in \{1, \ldots, n\}$, $\frac{\partial d_i}{\partial w_j}(x, w, \tilde{x}, \tilde{w}) \leq 0$ for all $x, \tilde{x} \in X$ and all $w, \tilde{w} \in W$ whenever the derivative exists.
- For all $i \in \{1, \ldots, n\}$ and all $k \in \{1, \ldots, m\}$, $\frac{\partial d_i}{\partial w_k}(x, w, \tilde{x}, \tilde{w}) \geq 0$ and $\frac{\partial d_i}{\partial w_k}(x, w, \tilde{x}, \tilde{w}) \leq 0$ for all $x, \tilde{x} \in X$ and all $w, \tilde{w} \in W$ whenever the derivative exists.

If $\hat{(5)}$ is mixed-monotone with respect to $d$, $d$ is said to be a decomposition function for $\hat{(5)}$, and when $d$ is clear from context we simply say that $\hat{(5)}$ is mixed-monotone. The mixed-monotonicity property is useful for, e.g., efficient reachable set computation, and these techniques have been applied in domains including transportation system [16], biological systems [17]. In these works, the authors construct decomposition functions from domain knowledge, however, it was recently shown in [11] that all systems of the form $\hat{(5)}$ are mixed-monotone and, thus, for all $F$ as in $\hat{(5)}$ there exists a $d$ satisfying the conditions of Definition 4. Nonetheless, identifying an appropriate decomposition function for ones particular setting still generally requires domain expertise, and we exemplify this point in the case study presented at the end of this work.

Let $\Phi^F(T; x, w)$ denote the state of $\hat{(5)}$ reached at time $T \geq 0$ starting from $x \in X$ at time 0 under the piecewise
continuous input \( w : [0, T] \to W \), and let
\[
R^+(T; \mathcal{X}_0) := \left\{ \Phi^b(T; x, w) \in \mathcal{X} \mid x \in \mathcal{X}_0 \right\}
\]
for some \( w : [0, T] \to W \).

\[(6)\]

de note the time-\( T \) forward reachable set of \( 5 \) from the set of initial conditions \( \mathcal{X}_0 \subseteq \mathcal{X} \). We next recall how over-approximations of reachable sets can be efficiently computed by considering a deterministic auxiliary system constructed from the decomposition function.

Assume \( 5 \) is mixed-monotone with respect to \( d \), and construct
\[
\dot{x} = e(x, \tilde{x}) := \begin{bmatrix} d(x, w, \tilde{x}, w) \\ d(\tilde{x}, w, x, w) \end{bmatrix}.
\]

\[(7)\]
The system \( 7 \) is the embedding system relative to \( d \), and we let \( \Phi^e(T; a) \) denote the state of this system at time \( T \geq 0 \) when initialized at \( a \in \mathcal{X} \times \mathcal{X} \) at time 0.

Proposition 1 (from Proposition 1). Let \( \mathcal{X}_0 = [x, \tilde{x}] \) for some \( x, \tilde{x} \). If \( \Phi^e(t; (x, \tilde{x})) \in \mathcal{X} \times \mathcal{X} \) for all \( 0 \leq t \leq T \), then
\[
R^+(T; \mathcal{X}_0) \subseteq [\Phi^e(T; (x, \tilde{x}))].
\]

By abuse of notation, we let \( \Phi^e(T; x) := \Phi^e(T; (x, x)) \), and thus it follows from Proposition 1 that
\[
R^+(T; x) \subseteq [\Phi^e(T; x)].
\]

\[(8)\]
for all \( x \in \mathcal{X} \) and all \( T \geq 0 \).

V. MIXED-MONOTONICITY BASED ACTIVE SET INVARIANCE

In this section, we present a solution to the problem statement and design a controller architecture which both allows the system to leave \( S_b \) and ensures that the system never enters \( \mathcal{X}_u \). The proposed controller uses a modified CBF formulation, where we now use mixed-monotonicity based reachability methods to assess the nondeterminism in the system model.

A. Problem Formulation

As prescribed in the problem statement, we assume a system of the form \( 1 \), an unsafe set \( \mathcal{X}_u \subset \mathcal{X} \), and a backup controller \( u^b \) with a compact backup region \( R_b = \{ x \in \mathcal{X} \mid h(x) \geq 0 \} \). We fix a desired controller \( u^d \) which is assumed to be preferable to the backup controller by some performance metric and, as in Section IV, we assume \( \mathcal{X} \) is an extended hyperrectangle and \( W = [w, \tilde{w}] \).

We denote by
\[
\dot{x} = F^b(x, w) := f(x) + g_1(x)u^b(x) + g_2(x)w
\]
the closed-loop dynamics of \( 1 \) under \( u^b \) and we let \( \Phi^b(T; x, w) := \Phi(T; x, u^b, w) \) denote the state transition function of this system. Thus, \( h \) is a control barrier function for \( 9 \) and \( u^b \) is safe with respect to \( S^b \). Additionally, we denote by
\[
S^b(T) := \left\{ x \in \mathcal{X} \mid \Phi^b(T; x, w) \in S_b \right\}
\]
for all \( w : [0, T] \to W \).

\[(10)\]

the time-\( T \) basin of attraction of \( S_b \), which is the set of states in \( \mathcal{X} \) that are guaranteed to enter \( S_b \) along trajectories of \( 9 \) within the time horizon \([0, T] \).

Remark 2. As a result of the fact that \( S_b \) is robustly forward invariant for \( 9 \), we additionally have that \( S^b(T) \) is robustly forward invariant for \( 9 \) for all \( T \geq 0 \).

As in [2], the ASIF formulation presented in this section allows the system to leave the safe set \( S_b \) in instances where the backup control policy is known to return the system to \( S_b \) on some finite time horizon. For this reason, we associate the backup control policy \( (u^b, S_b) \) with a fixed backup time \( T_b \), as formalised next.

Assumption 3. The backup dynamics \( 9 \) under the backup dynamics \( 9 \) do not intersect the unsafe set, i.e., \( S^b(T_b) \cap \mathcal{X}_u = \emptyset \).

To verify Assumption 2 holds, one can overapproximate backward reachable backward reachable sets of \( S_b \) under \( 9 \), and check for intersection with the unsafe set \( \mathcal{X}_u \). Many techniques allow for such an overapproximation and in the case study presented later, we implement one such method based on the mixed-monotonicity property. Moreover, while we assume \( T_b \) is known \( a \) priori, \( S^b(T_b) \) itself may be difficult to calculate in closed form. Thus, while a natural solution to the problem statement may be to construct a CBF-based ASIF to ensure the forward invariance of \( S^b(T_b) \), this solution may not be practically implementable when \( S^b(T_b) \) is not known. The ASIF presented later in this section uses mixed-monotonicity based reachability methods to assess whether or not the current system state is contained in \( S^b(T_b) \), and in this way we avoid an explicit description of \( S^b(T_b) \).

Lastly, we assume the backup dynamics \( 9 \) are mixed-monotone.

Assumption 4. The backup dynamics \( 9 \) are mixed-monotone with respect to the decomposition function \( d \), and we let \( \Phi^e \) denote the transition function of its respective embedding system.

As discussed in the Introduction, Assumption 3 is not especially restrictive since large classes of systems have been shown to be mixed-monotone with closed form expressions for the decomposition function \( d \).

B. Construction Methodology

Given \( x \in \mathcal{X} \), possibly with \( x \notin S^b \), our goal is to determine a suitable value \( u^{\text{ASIF}}(x) \); as suggested by the problem statement, \( u^{\text{ASIF}}(x) \) should be equal or close to \( u^d(x) \) if it is safe to do so. One method to determine whether or not \( u^{\text{ASIF}}(x) \) should be equal to \( u^d(x) \) is to assess the safety of the backup controller with respect to \( x \), i.e., if \( R^b(T; x) \subseteq S^b \) for some \( T < T_b \), then \( u^{\text{ASIF}}(x) = u^d(x) \) is allowed, where we let \( R^b(T; x) \) denote the time-\( T \) forward reachable set of \( 9 \) as in \( 6 \). We next present a family of functions that, for given \( x \in \mathcal{X} \), can be used to assess whether or not \( R^b(T; x) \subseteq S^b \) for some \( T < T_b \), and these functions exploit the mixed-monotonicity of \( 9 \).
Define
\[ \gamma_{\text{ideal}}(T; x) := \inf_{z \in [\Phi^c(T; x)]} h(z) = \min_{z \in \{\Phi_c(T; x)\}} h(z), \] (11)
where the second equality comes from the concavity on \( h \).

Moreover, from Assumption 2 the fact that \( \gamma \) with respect to \( \Phi \) renders \( S \) robustly forward invariant on (9), we additionally have \( R \).

We show in the following lemma how \( \gamma_{\text{ideal}} \) is used to determine whether a state \( x \in \mathcal{X} \) is contained in \( S_b^+(T) \) for given \( T \geq 0 \).

**Lemma 1.** For all \( x \in \mathcal{X} \) and all \( T \geq 0 \),
\[ \gamma_{\text{ideal}}(T; x) \geq 0 \Rightarrow x \in S_b^+(T). \] (12)

**Proof.** Fix \( x \in \mathcal{X} \) and \( T \geq 0 \) such that \( \gamma_{\text{ideal}}(T; x) \geq 0 \). Then for all \( z \in [\Phi^c(T; x)] \) we have \( h(z) \geq 0 \), and thus \([\Phi^c(T; x)] \subset S_b \). From Proposition 1 we have \( R_b^+(T; x) \subset [\Phi^c(T; x)] \), and therefore \( \Phi^b(T; x, w) \subset S_b \) for all \( w \). Therefore \( x \in S_b^+(T) \). \( \square \)

Next define
\[ \Psi_{\text{ideal}}(x) = \sup_{0 \leq \tau \leq T_b} \gamma_{\text{ideal}}(\tau; x). \] (13)

We show in the following proposition how \( \Psi_{\text{ideal}} \) is used to assess whether the backup control policy \( u_b \) is safe with respect to a given state.

**Proposition 2.** If
\[ \Psi_{\text{ideal}}(x) \geq 0 \] (14)
for some \( x \in \mathcal{X} \), then applying the backup control policy starting from \( x \) at time 0 ensures that there exists a time \( T \leq T_b \) such that \( R_b^+(t; x) \subset S_b \) for all \( t \geq T \). In this case, we also have that \( u_b \) is safe with respect to \( x \).

**Proof.** Assume that there exists an \( x \in \mathcal{X} \) satisfying (14). Then we have
\[ \sup_{\tau \leq T_b} \gamma_{\text{ideal}}(\tau; x) \geq 0. \] (15)

Therefore, there must exist a time \( T \leq T_b \) such that \( \gamma_{\text{ideal}}(T; x) \geq 0 \) and, at this time \( R_b^+(T; x) \subset S_b \); see (12). Moreover, from Assumption 2 the fact that \( S_b \) is robustly forward invariant on (9), we additionally have \( R_b^+(t; x) \subset S_b \) for all \( t \geq T \). \( \square \)

As a corollary to Proposition 2 note that the set
\[ S_{\Psi} := \{ x \in \mathcal{X} | \Psi_{\text{ideal}}(x) \geq 0 \} \] (16)
is robustly forward invariant on (9), and we have
\[ S_{\Psi}^\text{ideal} \subset S_b^+(T_b). \] (17)

In summary, \( \Psi_{\text{ideal}}(x) \) is positive for states \( x \in \mathcal{X} \) for which the backup controller is safe, and applying the backup controller to (1) starting from \( x \) ensures the system enters \( S_b \) on the time horizon \([0, T_b]\). However, applying the backup controller may not be necessary; in fact, any control action that renders \( S_{\Psi}^\text{ideal} \) robustly forward invariant will be safe with respect to \( x \). Control barrier functions are well suited for this task when the relevant functions are differentiable, however, \( \gamma_{\text{ideal}} \) and \( \Psi_{\text{ideal}} \) are generally not differentiable due to the min construction in (17). In the next section, we present a novel soft-min construction of \( \gamma_{\text{ideal}} \) and \( \Psi_{\text{ideal}} \) which ensures differentiability.

C. Barrier-Based ASIF Construction

We next present a differentiable relaxation of the functions \( \gamma_{\text{ideal}} \) and \( \Psi_{\text{ideal}} \), and these new functions are later incorporated in a control barrier function based ASIF.

We first recall the Log-Sum-Exponential function.

**Definition 5 (Log-Sum-Exponential).** We denote by
\[ \text{LSE}(\mathcal{S}, p) = -\frac{1}{p} \log \sum_{s \in \mathcal{S}} \exp(-p \cdot s) \] (18)
the Log-Sum-Exponential of the finite set \( \mathcal{S} \subset \mathbb{R} \) with respect to the parameter \( p > 0 \).

The Log-Sum-Exponential has several useful properties:
- \( \text{LSE}(\mathcal{S}, p) \) is differentiable with respect to the elements of \( \mathcal{S} \), and
- \( \text{LSE}(\mathcal{S}, p) \) approximates \( \min \mathcal{S} \), i.e.,
\[ \min \mathcal{S} - \frac{n}{p} \log 2 \leq \text{LSE}(\mathcal{S}, p) < \min \mathcal{S} \] (19)
for all \( p > 0 \), and this approximation can be made arbitrarily tight by choosing \( p \) large enough.

Next we introduce a continuously differently relaxation of \( \Psi_{\text{ideal}} \) and \( \gamma_{\text{ideal}} \) from the previous section. To that end, fix \( p > 0 \) and consider
\[ \gamma(t; x) := \text{LSE}(\{ h(z) \mid z \in [\Phi^c(t; x)] \}, p) \]
\[ = -\frac{1}{p} \log \sum_{z \in [\Phi^c(t; x)]} \exp(-p \cdot h(z)), \] (20)
where, from (19) we have
\[ \gamma_{\text{ideal}}(t; x) - \frac{n}{p} \log 2 \leq \gamma(t; x) < \gamma_{\text{ideal}}(t; x). \] (21)

Next define
\[ \Psi(x) = \sup_{0 \leq \tau \leq T_b} \gamma(\tau, x), \] (22)
and likewise \( S_{\Psi} := \{ x \in \mathcal{X} | \Psi(x) \geq 0 \} \). Importantly, \( \Psi(x) \) is differentiable with
\[ \frac{\partial \Psi}{\partial x}(x) = \frac{\partial \gamma}{\partial x}(\tau^*(x), x), \] (23)
where \( \tau^* \) is the maximizer from (22), i.e., \( \tau^*(x) \) satisfies \( \Psi(x) = \gamma(\tau^*(x), x) \), and this is a result of [18, Theorem 1].

In practice, \( \frac{\partial \Phi^c}{\partial x}(t, x) \) is computed as follows. First, \( \Phi^c(t, x) \) is computed for \( t \) in the interval \([0, T_b]\) by simulating the embedding dynamics (7), and the numerically simulated trajectory is used to identify the minimizer \( \tau^*(x) \) for (22).

Next, \( \frac{\partial \Phi^c}{\partial x}(\tau^*(x), x) \) is computed numerically; for example, \( n \) additional simulations of horizon \( \tau^*(x) \) can be used to approximate the \( n \) columns of the Jacobian matrix \( \frac{\partial \Phi^c}{\partial x} \).

Lastly, \( \frac{\partial \Psi}{\partial x}(\tau^*(x), x) \) is obtained via the chain rule using prior computations.
Lemma 2. $S_\Psi$ is a strict under-approximation of $S_{\Psi}^{\text{ideal}}$, i.e. $S_\Psi \subset S_{\Psi}^{\text{ideal}}$.

The proof of Lemma 2 is a direct result of (19). As derived in Section IV-B, $S_{\Psi}^{\text{ideal}}$ is robustly forward invariant on $\Omega$, however, $S_\Psi$ may not be. Further, $S_\Psi$ may not be robustly forward invariant under any control policy, even though it is true that if $\Psi(x) \geq 0$ for some $x$, then applying $u^b$ will keep $\Psi(x)$ from decreasing sometime before $x$ enters $S_b$; $\Psi(x)$ could decrease by as much as $\frac{\beta}{p} \log 2$ due to the fact that $\gamma(t; x)$ is an under approximation of $\gamma^{\text{ideal}}(t; x)$. Thus, even though a natural barrier-function-based reasoning might lead one to choose an input such that

$$\frac{d\Psi}{dt}(x(t)) \geq -\alpha(\Psi(x(t)))$$

for some class-$\mathcal{K}$ function $\alpha : \mathbb{R} \to \mathbb{R}$, this may not be possible when $\Psi(x)$ is close to zero, and in particular, it may be the case that choosing $u^b$ violates (24). However, due to the fact that $S_\Psi \subset S_{\Psi}^{\text{ideal}}$, if for some $x \in S_\Psi$, we have that $u^b$ violates (24), then $u^b$ is safe with respect to $x$ from Proposition 2 and thus it is acceptable to immediately switch to the backup control policy to retain safety.

We next present our main result: an assured controller for nondeterministic control systems of the form (1). This controller is presented in pseudocode (see Algorithm 1) and control actions are chosen point-wise in time.

Let $\Phi^{\text{ASIF}}(T; x, w)$ denote the state of (1) at time $T \geq 0$ when inputs are chosen using Algorithm 1 and when beginning from initial state $x \in \mathcal{X}$ at time 0 and subjected to the piecewise continuous inputs $w$.

Theorem 1. For all initial conditions $x \in S_b$ and any Lipschitz continuous controller $u^d : \mathcal{X} \to \mathbb{R}^m$, the controller $u^{\text{ASIF}}$ from Algorithm 1 is such that $\Phi^{\text{ASIF}}(T; x, w) \notin \mathcal{X}_u$ for all $T \geq 0$.

Theorem 1 follows directly from Proposition 2 and the preceding discussion; we thus omit a formal proof.

In summary, the assured controller $u^{\text{ASIF}}$ defined by Algorithm 1(a) evaluates to the desired control input whenever possible, (b) allows the system to leave the safe region $S_b$, and (c) ensures the system never enters the unsafe set $\mathcal{X}_u$. Moreover, the optimization problem posed in Algorithm 1 contains only a finite number of affine constraints, where we note that the CBF constraint in Line 4 is only evaluated at the vertices of $\Omega$. Thus, the proposed assured controller can be computationally amenable to real-world applications, and we demonstrate the construction and implementation of such an ASIF through a case study provided in the next section.

VI. NUMERICAL EXAMPLE: ENFORCING INTER-AGENT DISTANCE CONSTRAINTS ON A VEHICLE PLATOON

In this section we demonstrate the applicability of Algorithm 1 and design an ASIF which enforces inter-agent distance constraints on a platoon of vehicles.

A. Problem Setting

Consider a platoon of $N \geq 2$ vehicles, whose velocity dynamics are given as

$$\dot{x}_i = \beta x_i + a_i + w_i,$$

where $x_i \in \mathbb{R}$ denotes the velocity of the $i^{\text{th}}$ vehicle, for $i \in \{1, \ldots, N\}$. Here, $a_i$ denotes the acceleration of the vehicle, which is controlled by a global planner, $\beta \leq 0$ denotes a friction coefficient and $w \in \mathcal{W} \subset \mathbb{R}^N$ denotes a bounded additive noise term. We additionally let $p_i \in \mathbb{R}$ denote the position of the $i^{\text{th}}$ vehicle, so that $\dot{p}_i = x_i$.

Control decisions are made after referencing the relative displacements of vehicles in the platoon. In particular, the accessible displacements are described by an undirected graph $G$ with each node of the graph representing a vehicle and each edge of the graph denoting a displacement measurement between neighboring nodes. We assume an arbitrary orientation of the edges of $G$, so that the network is described by the incidence matrix $A \in \mathbb{R}^{N \times K}$ with

$$A_{i,j} = \begin{cases} 1 & \text{if vertex } i \text{ is the head of edge } j \\ -1 & \text{if vertex } i \text{ is the tail of edge } j \\ 0 & \text{otherwise} \end{cases}$$

for $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, K\}$ for a graph with $K$ edges. In this case, the vector containing the accessible displacements is given by $z = Ap \in \mathbb{R}^K$, and the platoon dynamics then become

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \beta I & 0 \\ A^T & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} - \begin{bmatrix} D \\ 0 \end{bmatrix} u(z) + \begin{bmatrix} w \\ 0 \end{bmatrix}$$

with control input $u(z) = [u_1(z_1), \ldots, u_K(z_K)]^T$.

While the theoretical results apply in the general setting of (26) with an arbitrary number of vehicles and links, for the remainder of the study, we consider a 3-cart instantiation.
of (25) with 2 control inputs, i.e., we take $N = 3$ and $K = 2$, and connectivity is given by

$$A = \begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix}. \quad (27)$$

In this case $\mathcal{X} = \mathbb{R}^5$, and we fix $\beta = -1$ and $\mathcal{W} = [-0.1, 0.1]^3$. This problem setting is shown in Figure 1.

We aim to enforce inter-agent distance constraints on (26) by applying the ASIF controller presented in Algorithm 1. Specifically, we take an unsafe set

$$\mathcal{X}_u = \left\{ \begin{bmatrix} x \\ z \end{bmatrix} \in \mathbb{R}^5 \mid |z_1| \geq 8 \text{ or } |z_2| \geq 8 \right\}, \quad (28)$$

and we ignore vehicle collisions so that $z_1$ and $z_2$ are allowed to change sign over trajectories of (26).

In the next section, we form a backup controller for (1) which is safe by virtue of a forward invariant safe region.

### B. Constructing the Backup Controller

We choose a backup controller

$$u^b(z) = \kappa \begin{bmatrix} \tanh(\sigma z_1) \\ \tanh(\sigma z_2) \end{bmatrix}, \quad (29)$$

with $\kappa = 2$ and $\sigma = 1/2$. Roughly speaking, (29) acts as two identical nonlinear springs which pull the carts together when applied to (26); by this description, $\kappa$ describes the maximum force which the springs apply before saturation, and $\sigma$ describes the distance at which the springs saturate. The closed-loop dynamics of (26) under the backup control policy are

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = F^b \begin{bmatrix} x \\ z \end{bmatrix}, \quad w = \begin{bmatrix} \beta x + w \\ A^T x \end{bmatrix} - \begin{bmatrix} -A\kappa \\ 0 \end{bmatrix} \begin{bmatrix} \tanh(\sigma z_1) \\ \tanh(\sigma z_2) \end{bmatrix}, \quad (30)$$

and (30) is mixed-monotone on $\mathcal{X}$ with decomposition function

$$d\left( \begin{bmatrix} x \\ z \end{bmatrix}, \begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix}, w, \bar{w} \right) = \begin{bmatrix} \beta x + w \\ (A^+)^T x + (A^-)^T \bar{x} \end{bmatrix} - \begin{bmatrix} -A\kappa \\ 0 \end{bmatrix} \begin{bmatrix} \tanh(\sigma z_1) \\ \tanh(\sigma z_2) \end{bmatrix} - \begin{bmatrix} A^-\kappa \\ 0 \end{bmatrix} \begin{bmatrix} \tanh(\sigma \bar{z}_1) \\ \tanh(\sigma \bar{z}_2) \end{bmatrix}, \quad (31)$$

where $A^+$ and $A^-$ denote the positive and negative parts of $A$, respectively, and are given by

$$A^+_{i,j} = \begin{cases} A_{i,j} & \text{if } A_{i,j} \geq 0 \\ 0 & \text{otherwise} \end{cases}, \quad A^-_{i,j} = \begin{cases} A_{i,j} & \text{if } A_{i,j} < 0 \\ 0 & \text{otherwise} \end{cases}. \quad (32)$$

To construct a backup region $S_b$, we consider a local linearization of (30); that is, for small disturbances, (30) locally behaves as

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \beta I & -A\kappa \sigma \\ A^T & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix}. \quad (33)$$

Further, (33) is asymptotically stable to the origin and is certified by the quadratic Lyapunov function

$$V(x, z) = \begin{bmatrix} x^T \\ z^T \end{bmatrix} P \begin{bmatrix} x \\ z \end{bmatrix} \quad (34)$$

for

$$P = \begin{bmatrix} \kappa \sigma + A A^T \\ -A^T \end{bmatrix} \begin{bmatrix} \kappa^2 \sigma^2 + \beta^2 I + \kappa \sigma A A^T \end{bmatrix}. \quad (35)$$

Thus, we consider an invariant safe set

$$S_b = \left\{ \begin{bmatrix} x \\ z \end{bmatrix} \in \mathbb{R}^5 \mid V(x, z) \leq \delta \right\} \quad (36)$$

for appropriate $\delta > 0$. For the parameters taken in this study, $S_b$ from (35) was verified to be robustly forward invariant on (30) when $\delta = 9/4$.

Let

$$R^b_b(T; \mathcal{X}_1) := \left\{ x \in \mathcal{X} \mid \Phi^T(T; x, w) \in \mathcal{X}_1 \right\} \quad (37)$$

do not denote the time-$T$ backwards reachable set of (30). We next calculate a backup horizon $T_b$ such that

$$S^+_b(T_b) \cap \mathcal{X}_u = \emptyset, \quad (38)$$

and this is done by showing that $R^b_b(T; S_b) \cap \mathcal{X}_u = \emptyset$. In particular, we overapproximate $R^b_b(1; S_b)$ using [10, Proposition 2] and find that $R^b_b(1; S_b) \cap \mathcal{X}_u = \emptyset$. Therefore we take a backup time $T_b = 1$ which satisfies (37).

We now have the necessary tools to implement Algorithm 1. We demonstrate the creation and application of the active set invariance filter in the next section.

### C. Simulated Implementation

We next construct an ASIF to assure the system (26), where we take the backup controller $u^b$ from (29), safe set $S_b$ from (35), and backup time $T_b = 1$. In this case, $\gamma$ is given by (20) where we fix $p = 1000$ and $\Phi^c$ is taken in reference to $d$. Additionally, define $\Psi$ as in (22). Now an assured controller is given by Algorithm 1.

For the purpose of this study, we hypothesize an open-loop desired control input

$$u^d(t) = \begin{bmatrix} -0.3 \sin(\pi t/4) \\ 0.2 \cos(\pi t/2) \end{bmatrix}, \quad (39)$$

and simulate the system (26) under the ASIF controller Algorithm 1, where we let $\alpha(\psi) = 1000\psi^3$. Note that, even though the theory above was developed assuming a given desired closed-loop feedback controller, the same approach is applicable if an open-loop control input is provided instead as a function of time. A 4-second simulation is conducted using MATLAB 2020a and simulation results are
provided in Figure 2. The system response is simulated via Euler integration with a time-step of 0.01 seconds and the optimization problem Algorithm 1 is computed at each time-step using CVX, a convex optimization tool built for use with MATLAB. In the case of this experiment, the average solver time is 0.54 seconds per optimization.

In the simulation the assured controller $u^{\text{ASIF}}$ drives the system (26) out of the safe set; however, the system remains in $S_b^0(1)$ and all points along the system trajectory are safe with respect to $u^b$.

VII. CONCLUSION

This work presents a problem formulation for runtime assurance for control systems with disturbances, and a specific solution to the problem statement is presented, whereby the nondeterminism in the system model is accommodated via the mixed-monotonicity property. The proposed assured controller computes an optimization problem containing only a finite number of affine constraints, and we demonstrate the applicability of our construction through a case study.

REFERENCES

[1] M. Abate, E. Feron, and S. Coogan, “Monitor-based runtime assurance for temporal logic specifications,” in 2019 IEEE 58th Conference on Decision and Control (CDC), pp. 1997–2002, 2019.

[2] T. Gurriet, M. Mote, A. D. Ames, and E. Feron, “An online approach to active set invariance,” in 2018 IEEE Conference on Decision and Control (CDC), pp. 3592–3599, Dec 2018.

[3] T. Gurriet, M. Mote, A. Singlelery, E. Feron, and A. D. Ames, “A scalable controlled set invariance framework with practical safety guarantees,” in 2019 IEEE 58th Conference on Decision and Control (CDC), pp. 2046–2053, 2019.

[4] A. D. Ames, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs with application to adaptive cruise control,” in 53rd IEEE Conference on Decision and Control, pp. 6271–6278, Dec 2014.

[5] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” in 2019 18th European Control Conference (ECC), pp. 3420–3431, 2019.

[6] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” IEEE Transactions on Automatic Control, vol. 62, pp. 3861–3876, Aug 2017.

[7] T. Schouwenaars, Safe trajectory planning of autonomous vehicles. PhD thesis, Massachusetts Institute of Technology, 2006.

[8] S. Bak, D. K. Chivukula, O. Adekunle, M. Sun, M. Caccamo, and L. Sha, “The system-level simplex architecture for improved real-time embedded system safety,” in 2009 15th IEEE Real-Time and Embedded Technology and Applications Symposium, pp. 99–107, IEEE, 2009.

[9] S. Coogan and M. Arcak, “Stability of traffic flow networks with a polytree topology,” Automatica, vol. 66, pp. 246–253, Apr. 2016.

[10] M. Abate and S. Coogan, “Computing robustly forward invariant sets for mixed-monotone systems,” in 2020 IEEE 59th Conference on Decision and Control (CDC), 2020. An extended version of this work is available through ArXiv: https://arxiv.org/abs/2003.05912.

[11] M. Abate, M. Dutreix, and S. Coogan, “Tight decomposition functions for continuous-time mixed-monotone systems with disturbances,” IEEE Control Systems Letters, vol. 5, no. 1, pp. 139–144, 2021.

[12] G. Enciso, H. Smith, and E. Sontag, “Nonmonotone systems decomposable into monotone systems with negative feedback,” J. Differential Equations J. Differential Equations, vol. 22405007, pp. 205–227, 05 2006.

[13] H. Smith, Monotone Dynamical Systems: An Introduction to the Theory of Competitive and Cooperative Systems. Mathematical surveys and monographs, American Mathematical Society, 2008.

The code for this experiment is publicly available on the GaTech FACTS Lab Github: https://github.com/gtfactslab/Abate_CDC2020

Fig. 2: Implementing Algorithm 1 to assure the vehicle platoon (26). The carts begin with an initial velocity state $x_0 = [-1/4, 0, 1/2]^T$ and an initial displacement state $z_0 = [1/4, 1/2]^T$. A random disturbance $w : [0, 4] \rightarrow \mathcal{W}$ is also chosen.