Intention-Aware Supervisory Control with Driving Safety Applications

Yunus E. Sahin*, Zexiang Liu*, Kwesi Rutledge*, Dimitra Panagou, Sze Zheng Yong, Necmiye Ozay

***Abstract*— This paper proposes a guardian architecture, consisting of an estimation and a supervisor module providing a set of inputs that guarantees safety, in driving scenarios. The main idea is to offline compute a library of robust controlled invariant sets (RCIS), for each possible driver intention model of the other vehicles, together with an intention-agnostic albeit conservative RCIS. At run-time, when the intention estimation module determines which driver model the other vehicles are following, the appropriate RCIS is chosen to provide the safe and less conservative input set for supervision. We show that the composition of the intention estimation module with the proposed intention-aware supervisor module is safe. Moreover, we show how to compute intention-agnostic and intention-specific RCIS by growing an analytically found simple invariant safe set. The results are demonstrated on a case study on how to safely interact with a human-driven car on a highway scenario, using data collected from a driving simulator.**

I. Introduction

As more and more autonomy-related functionalities are integrated into modern passenger vehicles, questions on safety and trust arise. Some recent research efforts have tried to address the safety issue from the formal verification [1], [16] and correct-by-construction control synthesis perspectives [15]. In these formal approaches, set invariance plays a central role in guaranteeing safety [5], [17]. The boundary of an invariant set can be thought of as a barrier that separates the part of the state-space the system can safely operate in from the part that is deemed unsafe. This boundary can be represented by level-sets of differentiable functions [2], polyhedra, or approximate solutions of partial differential equations capturing the safety problem [3].

Finding *robust controlled invariant sets*, sets that can be rendered invariant with the right choice of *control inputs* in a way that is *robust* to the factors controlled by external agents (such as behavior of other drivers, disturbances) and model uncertainty, is a key problem in safety control. However, in a driving scenario, trying to develop a single model that covers all possible behaviors of the other drivers often leads to conservatism, i.e., smaller invariant sets, as we assume the worst-case scenario in the invariant set computation. The goal of this paper is to show how online estimation of the behavior models (or intentions) of other drivers can reduce conservatism by developing a library of RCIS offline for different intention models and selecting an appropriate one at run-time. Learning or extracting models/intentions of other drivers or learning controllers that mimic humans [6], [19], [13] are relevant, yet orthogonal, to our work as our main focus is to develop a framework to show how such models and their online estimation can lead to more permissive, yet safe, driving.

* denotes equal contribution. YES, ZL, KL, and NO are with the EECS department and DP is with the Aerospace Engineering department of the University of Michigan, Ann Arbor. SZY is with the SEMTE, Arizona State University.

II. Preliminaries

This section introduces the notation and provides certain concepts that are used throughout the rest of the paper. For a given set S, $\mathcal{P}_{\geq n}(S)$ denotes the set of subsets of S with at least n elements, S^* is the set of all finite sequences from S (including the empty sequence).

A *discrete-time affine* system has the following state update equation:

$$q^+ = Aq + Bu + B_w w + F \tag{1}$$

where q is the state of the system, u is the controlled input to the system, w is the uncontrolled input (disturbance), and the matrices (A, B, B_w, F) are of the appropriate dimensions. The state space, space of allowed inputs, and the space of feasible uncontrolled inputs are referred to as $\mathcal{Q}$, $\mathcal{U}$, and $\mathcal{W}$, respectively.

A *piecewise-affine (PWA)* system is defined by a set $f = \{(f^i, D^i)\}_{i=1}^m$ that describes the evolution of the states in different regions of the state space, that is,

$$q^+ = f^i(q, u, w) \text{ for } q \in D^i \tag{2}$$

where $\mathcal{D} = \{D^i\}_{i=1}^m$ form a partition of the state space $\mathcal{Q}$ and each $f^i : D^i \times \mathcal{U} \times \mathcal{W} \mapsto \mathcal{Q}$ is a discrete-time affine system in the form of (1), denoting the dynamics used in D_i. With a slight abuse of notation, we also write

$$q^+ = f(q, u, w) \tag{3}$$

to represent the PWA system corresponding to f.

Given a PWA system f, a set $\mathcal{C} \subseteq \mathcal{Q}$ of states is called *robust controlled invariant* if

$$\forall q \in \mathcal{C} : \exists u \in \mathcal{U} : \forall w \in \mathcal{W} : f(q, u, w) \in \mathcal{C}. \tag{4}$$

In words, trajectories that start inside an RCIS can be enforced to stay there indefinitely.

A. Invariant Set Computation

There are many methods in the literature for computing or approximating controlled invariant sets [4], [5], [8], [18]. The main computational building block of these algorithms is the *one-step backward reachable set operation*, that we denote as $\text{Pre}(\cdot)$. For a given set R and dynamics f, the one-step backward reachable set of R under f is defined as

$$\text{Pre}^f(R) = \{q \in \mathcal{Q} \mid \exists u \in \mathcal{U} : \ f(q, u, \mathcal{W}) \subseteq R\}. \quad (5)$$

Given a safe set $\mathcal{Q}_{safe}$, under mild conditions, the following iterations converge from outside to the maximal controlled invariant set in $\mathcal{Q}_{safe}$ when initialized with $\mathcal{C}_0 = \mathcal{Q}_{safe}$:

$$\mathcal{C}_{i+1} = \text{Pre}^f(\mathcal{C}_i) \cap \mathcal{Q}_{safe}. \quad (6)$$

If the update rule reaches a fixed point, i.e., $\mathcal{C}_i \subseteq \text{Pre}^f(\mathcal{C}_i)$, then the solution to that equation is the maximal invariant set contained in $\mathcal{Q}_{safe}$. On the other hand, although this is a monotonically non-increasing (in the set inclusion sense) sequence, the iterations are not guaranteed to terminate in finitely many steps, a problem that can be mitigated by approximation techniques [8], [18].

Alternatively, if one has an initial simple RCIS $\mathcal{C}_0$, computed either analytically or numerically, contained in some safe set $\mathcal{Q}_{safe}$, this set can be progressively expanded again via the same update rule (6). In this case, we obtain a monotonically non-decreasing sequence of sets $\Gamma_k \doteq \bigcup_{i=1}^{k} \mathcal{C}_i$, each of which themselves are robustly controlled invariant. Therefore, it can be terminated at anytime and one would obtain an RCIS. We call this method the *inside-out algorithm*.

Crucially, for PWA systems and sets described with unions of polytopes, the invariant set computation reduces to a set of polytopic operations. Moreover, when finding the exact $\text{Pre}(\cdot)$ is computationally hard, using an under-approximation does not compromise correctness when using the iterative algorithms in the sense that upon termination, the algorithm still results in an RCIS.

III. Problem Statement and Architecture

We start by describing the abstract problem that we are interested in solving. Let PWA system f of the form (3) represent the interaction of an ego agent with other agents where $q \in \mathcal{Q}$ is the combined states of all agents, control input $u = [u_e^\top \ \ u_o^\top]^\top \in \mathcal{U}_e \times \mathcal{U}_o$ is partitioned into two parts where *ego input* u_e is controlled by the ego agent and *external input* u_o is controlled by all other agents, and *disturbance* $w \in \mathcal{W}$ captures model uncertainty. We assume that the other agents behave according to a fixed *intention model* $I_i : \mathcal{Q} \to \mathcal{P}_{\geq 1}(\mathcal{U}_o)$, which is a set valued mapping that returns a set of external control inputs given a state. That is, if the system is currently at q, then the external control input u_o is restricted such that $u_o \in I_i(q) \subseteq \mathcal{U}_o$. While the actual specific intention model I_i is unbeknownst to the ego agent, a finite set $\mathcal{I} = \{I_1, \ldots, I_n\}$ of intention models is known a priori such that $I_i \in \mathcal{I}$. There are two sources of uncertainty from the perspective of the ego agent: one due to the fact that i^* is not known, another due to I_i being a set-valued map, capturing the variability within a specific intention. With a slight abuse of notation, we define $\mathcal{I}(q) \doteq \bigcup_{I \in \mathcal{I}} I(q)$, the set of all possible external control inputs that the ego agent presumes, given the current state q.

Our goal is to design a *supervisor module*, which restricts the inputs of the ego agent when needed, to ensure that the states of the system remain indefinitely in a safe set $\mathcal{Q}_{safe} \subseteq \mathcal{Q}$. However, due to the dynamics and disturbances in (3), we can only enforce that the system stays in a subset of $\mathcal{Q}_{safe}$, which is an RCIS that is computed according to Section II-A.

Let us define a supervisor module before stating the problem of interest formally.

Definition 1. *Given a system in the form of (3), a set of intention models $\mathcal{I}$, and a safe set $\mathcal{Q}_{safe}$, a supervisor module*

$$\mathcal{S}_{\mathcal{I}} : \mathcal{Q}_{safe} \mapsto \mathcal{P}(\mathcal{U}_e) \quad (7)$$

*takes a state measurement q and outputs a set $\mathcal{S}_{\mathcal{I}}(q) \subseteq \mathcal{U}_e$ of **admissible** ego inputs such that the admissible inputs $u_e \in \mathcal{S}_{\mathcal{I}}(q)$ enforce the system to indefinitely remain in the safe set regardless of the external input and the disturbance, i.e., $\mathcal{S}_{\mathcal{I}}(q) \neq \emptyset \implies \mathcal{S}_{\mathcal{I}}(q^+) \neq \emptyset$ for all $u_e \in \mathcal{S}_{\mathcal{I}}(q)$, $u_o \in \mathcal{I}(q)$ and $w \in \mathcal{W}$ where $q^+ = f(q, u, w)$.*

A supervisor's goal is to keep the system in the safe set. If the admissible ego input safe is empty, the system must either be in an unsafe state, or it is not possible for the ego agent to guarantee that the system stays in the safe set indefinitely. That is, there exists a finite sequence of external inputs, over which the ego agent has no control, and a finite sequence of disturbances that would eventually steer the system into an unsafe state, regardless of the ego input. On the other hand, the above definition implies that the set $\mathcal{C} = \{q \in \mathcal{Q}_{safe} \mid \mathcal{S}_{\mathcal{I}}(q) \neq \emptyset\}$ is an RCIS. Given two supervisors $\mathcal{S}_{\mathcal{I}}^1$ and $\mathcal{S}_{\mathcal{I}}^2$, we say $\mathcal{S}_{\mathcal{I}}^1$ is more *permissive* if $\mathcal{S}_{\mathcal{I}}^2(q) \subseteq \mathcal{S}_{\mathcal{I}}^1(q)$ for all $q \in \mathcal{Q}_{safe}$. The key insight in this paper is that, intuitively, a smaller set of intention models should lead to more permissive supervisors. That is, if $\tilde{\mathcal{I}} \subset \mathcal{I}$, for any $\mathcal{S}_{\mathcal{I}}$, there exists $\mathcal{S}_{\tilde{\mathcal{I}}}$ that is more permissive.

We now formally define the problem we are interested in solving and provide a solution method.

Problem 1. *Let a PWA system f in the form of (3), a set of intention models $\mathcal{I}$ and a safe set $\mathcal{Q}_{safe} \subset \mathcal{Q}$ be*

Fig. 1: Guardian architecture proposed to solve Problem 1

given. Find a supervisor module S_I as in Definition 1 and a set of initial states $C \subseteq Q_{safe}$ such that any trajectory that starts from an arbitrary state $q^0 \in C$ is guaranteed to indefinitely remain in C as long as the control input u_e is chosen from the set of admissible inputs, i.e., $u_e^t \in S_I(q^t)$ for all t.

Problem 1 can be solved using existing methods such as [15]. However, as previously mentioned, uncertainty in the external input u_o is larger from the perspective of the ego agent since the intention of other agents is unbeknownst to the ego agent a priori. As a result, the supervisor S_I must be designed so that it would guarantee safety for any intention model, which is conservative and not desirable. In reality, the ego agent could observe the other agents and decrease the uncertainty by invalidating intention models that are not consistent with the observed external inputs. Inspired by this observation, we propose a less conservative guardian architecture, which is illustrated in Figure 1, to solve Problem 1, that consists of a *library of supervisor modules* and an *intention estimation module*.

Definition 2. An intention estimation module

$$E : (Q \times U_e) \to 2^{\mathcal{I}_1}(\mathcal{I})$$

maps any state-ego input trajectory $qu_e^t = \{(q^0, u_e^0), \ldots, (q^t, u_e^t)\}$, to a non-empty subset $\mathcal{I}_v^{t+1} = E(qu_e^t) \subseteq \mathcal{I}$ of valid intentions such that there exist an external control input u_o^k and disturbance w^k that satisfy the following for all $k \in \{0, \ldots, t\}$:

$$\begin{aligned} q^{k+1} &= f(q^k, [u_e^k \; u_o^k], w^k), \text{ and} \\ u_o^k &\in \mathcal{I}_i(q^k) \text{ for all } \mathcal{I}_i \in \mathcal{I}_v^{t+1}. \end{aligned} \tag{8}$$

An estimation module indicates the set of intention models that are valid by invalidating the intentions that are inconsistent with a given state-input pair. Since the true intention $\mathcal{I}_i$ of the other agents is assumed to be constant over time, it is always included in the set of valid intentions, i.e., $\mathcal{I}_i \in E(qu_e), \forall qu_e \in (Q \times U_e)$. Note that, lengthening the state-input pair can only refine the set of valid intentions, thus, intention estimation over time is a monotonically non-increasing set for a system.

Given an instance of Problem 1, a more permissive supervisor can be designed by leveraging the information gained from such an intention estimation module. To do so, we compute a library of supervisors $\{S_I, S_{\mathcal{I}_1}, S_{\mathcal{I}_2}, \ldots, S_{\mathcal{I}_n}\}$[1]. As the notation indicates, we design a supervisor $S_{\mathcal{I}_i}$ for each possible intention model $\mathcal{I}_i$, together with an intention-agnostic supervisor S_I. During run-time, we switch between the supervisors, depending on the output of the intention estimation module E. This approach enables us to change the level of permissiveness depending on the observations, while still guaranteeing safety. That is, we use the supervisor module S_I when the true intention of the other agents is not yet known, and guarantee that the system remains in the safe set. Once the true intention $\mathcal{I}_i$ is revealed by the estimation module E, we switch to the corresponding supervisor $S_{\mathcal{I}_i}$, which is more permissive. As a result, the overall architecture is less conservative.

IV. The Scenario and System Models

To illustrate the concepts that are presented in this paper, we choose a simple autonomous driving scenario and explain the solution method referring to this scenario. However, the concepts we propose in this paper apply to the general framework explained in Section III.

Imagine two vehicles moving on a straight road with two lanes as illustrated in Fig 2. One of these vehicles, the *ego vehicle*, is controllable through u_e and can move both in lateral and longitudinal directions. The other vehicle is called the *lead vehicle* and its longitudinal motion is controlled by a fixed intention model chosen from a set of intention models. Intention models are assumed to react to the ego vehicle when the distance between the cars is less than some threshold. As stated earlier, while this set of intention models is known to the ego vehicle, the specific intention model that controls the lead vehicle is not. We assume that the lead vehicle has no lateral motion and always drives along the center of the right lane. The safety requirement for the ego vehicle is to keep a minimum safe distance between the vehicles, in both the longitudinal and the lateral directions.

We now provide dynamics that captures the aforementioned scenario and formally define the safety requirements.

A. Dynamics

The vehicles are treated as point masses, and their motion is modeled as follows:

$$\begin{aligned} v_{e;x}^+ &= v_{e;x} + (a_{e;x} - b_e v_{e;x})\,\Delta t + w_{e;x}\,\Delta t; \\ y_e^+ &= y_e + v_{e;y}\,\Delta t + w_{e;y}\,\Delta t; \\ v_{L;x}^+ &= v_{L;x} + (a_{L;x} - b_L v_{L;x})\,\Delta t + w_{L;x}\,\Delta t; \end{aligned} \tag{9}$$

[1] An even more permissive design can be achieved if we compute a supervisor for each subset of intentions, i.e., compute $S_{\mathcal{I}_v}$ for each $\mathcal{I}_v \in 2^{\mathcal{I}_1}(\mathcal{I})$. However, such an approach would be computationally more expensive as a trade-off.

where t (= 0.1) is the sampling time, $v_{e;x}$ is the longitudinal velocity of the ego vehicle, y_e is the lateral displacement of the ego vehicle with respect to the center of the right lane, and $v_{L;x}$ represents the longitudinal velocity of the lead vehicle. The ego vehicle is controlled through its longitudinal acceleration $a_{e;x}$ and lateral velocity $v_{e;y}$. The longitudinal acceleration of the lead vehicle, $a_{L;x}$, depends on the intention and is treated as external disturbance. Terms b_e (= 0.1) and b_L (= 0.1) are drag coefficients and $w_{e;x}(k) \in [-.15; .15]$, $w_{e;y}(k) \in [-.09; .09]$ and $w_{L;x}(k) \in [-.05; .05]$ are process noises. The relative longitudinal distance between the two vehicles is denoted by h and evolves according to the following:

$$h^+ = h + (v_{L;x} - v_{e;x})\, t. \tag{10}$$

As indicated by (10), positive values for h imply that the ego vehicle is behind the lead vehicle.

We now define the vectors $q = [v_{e;x};\ y_e;\ h;\ v_{L;x}]^\top$, $u_e = [a_{e;x};\ v_{e;y}]^\top$, $u_o = [a_{L;x}]$, $u = [u_e;\ u_o]^\top$, $w = [w_{e;x}, w_{e;y}, w_{L;x}]^\top$, and combine (9) and (10) in to the form (1), where $Q = [v_{e;x}^{min}; v_{e;x}^{max}] \times [y_e^{min}; y_e^{max}] \times [v_{L;x}^{min}; v_{L;x}^{max}]$.

B. Intention Models

We consider two driver intentions, denoted by $\in \{I_a; I_c\}$, corresponding to _Aggressive and _Cautious drivers[2]. Here, these drivers react to the ego vehicle only when it is close enough, that is, when the absolute value of the longitudinal distance is less than some threshold $|$. This area is called the _reaction zone and is illustrated in Fig. 2. When the ego vehicle is inside the reaction zone, the external input u_o is determined by an affine state-feedback policy; otherwise only a bound on the velocity is imposed in the choice of u_o. Having the reaction zone captures two properties: (i) since intentions are feedback policies in our setup, it is reasonable to assume feedback occurs when the vehicles are in the vicinity of each other, (ii) fixed intention assumption is automatically relaxed to intention being unchanged only within the reaction zone as outside the reaction zone the assumptions on all vehicles are the same. In addition to the acceleration bounds captured by U_o, we assume the lead car velocity is bounded by $v_{L;x} \in [v_{L;x}^{min}; v_{L;x}^{max}]$. One thing to note is that an affine state-feedback might lead to violation of the assumed acceleration and velocity bounds. These bounds mimic the physical limitations of the vehicles, thus, it is assumed not possible to exceed them. Thus, external input u_o is saturated when needed.

[2]We choose two intentions to clearly illustrate these concepts and stress to the reader that our framework is general enough to incorporate as many intention models as available.

The resulting dynamics for each intention model can be represented as a PWA system as shown below[3]

1) Aggressive Driver: Tries to match the speed of the ego vehicle when the ego vehicle is inside the reaction zone, thus making it harder to overtake:

$$a_{L;x} = \begin{cases} \max(\min(K_a q; \ _1); \ _2) + w; & \text{if } |h| \le h_r; \\ \max(\min((v_{L;x}^{des} - v_{L;x}); \ _1); \ _2) + w; & \text{o.w.} \end{cases} \tag{11}$$

where

$$_1 = \min\left(a_{L;x}^{max}; \frac{v_{L;x}^{max} - (1 - b_L t) v_{L;x}}{t}\right) \quad w^{max} \le w_{L;x}^{max}$$

$$_2 = \max\left(a_{L;x}^{min}; \frac{v_{L;x}^{min} - (1 - b_L t) v_{L;x}}{t}\right) \quad w^{min} \ge w_{L;x}^{min}. \tag{12}$$

The min and max operations in (11) and (12) ensure that the acceleration and velocity bounds for the lead vehicle are always respected. Note that the action of the aggressive driver is non-deterministic due to the term $w \in [w^{min}; w^{max}]$, which captures the variability within each intention model. Due to min and max operators used, resulting dynamics $f_a = \{f(f_a^j; D_a^j)\}_{j=1}^9$ is a PWA system with nine regions.

2) Cautious Driver: Tends to maintain its desired speed and makes it easier for ego vehicle to change lane or overtake. The cautious driver is modeled as follows:

$$a_{L;x} = \begin{cases} \max(\min(K_c q + k_c v_{L;x}^{des}; \ _1); \ _2) + w; & \text{if } |h| \le h_r; \\ \max(\min((v_{L;x}^{des} - v_{L;x}); \ _1); \ _2) + w; & \text{o.w.} \end{cases} \tag{13}$$

where $_1$ and $_2$ are defined as in (12). The resulting dynamics $f_c = \{f(f_c^j; D_c^j)\}_{j=1}^9$ is a PWA system with nine regions.

3) Bounded Velocity: When the intention of the lead vehicle is not known, we assume the worst case scenario and let $v_{L;x}$ to change arbitrarily fast. That is, $v_{L;x}^+$ can take any value between the lower and the upper bound, regardless of $v_{L;x}$. By doing so, we capture the behavior of both intentions. We use this conservative model when the intention of the lead vehicle is not known.

C. Safety Requirements

The ego vehicle is required to keep a minimum distance between two vehicles at all times. In this case, we can represent the set Q_{safe} of safe states as follows:

$$Q_{safe} \doteq Q_{safe}^1 \cap Q_{safe}^2 \cap Q_{safe}^3; \tag{14}$$

where $Q_{safe}^1 \doteq \{q \in Q \mid |h| \ge h_{min} \text{ or } y_e \notin [y_e^{min}; y_e^{max}]\}$ capturing safe distance during takeover, $Q_{safe}^2 \doteq \{q \in$

[3]The parameter values used in our experiments for these models are: $a_{L;x}^{max} = -a_{L;x}^{min} = 3\,m/s^2$, $w^{max} = -w^{min} = 0.1$, $v_{L;x}^{min} = 0\,m/s$, $v_{L;x}^{max} = 33.5\,m/s$, $K_{des} = 1$, $K_a = [1.0; 0; 0; 1]$, $K_c = [0; -0.1; 0.1; -0.01]$, $k_c = 0.01$, $v_{L;x}^{des} = 30\,m/s$, $h_r = 60\,m$, $h_{min} = 10\,m$, $v_{e;x}^{min} = 16\,m/s$, $v_{e;x}^{max} = 36\,m/s$, $y_e^{min} = -0.9m$, $y_e^{max} = 2.7m$. The input bounds used are given by $U_e = [-3; 3] \times [-1.8; 1.8]$ and $U_o = [-3; 3]$.

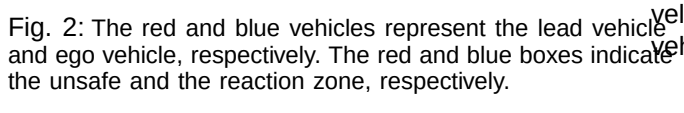


Fig. 2: The red and blue vehicles represent the lead vehicle and ego vehicle, respectively. The red and blue boxes indicate the unsafe and the reaction zone, respectively.

$Q \mid y_e \in [y_e^{min}; y_e^{max}]\}$ capturing lane keeping constraints, and $Q^3_{safe} \doteq \{q \in Q \mid v_{e;x} \in [v_{e;x}^{min}; v_{e;x}^{max}]\}$ capturing the speed limits. Note that, the resulting set Q_{safe} of safe states is not convex, but it can be represented as a union of polyhedra.

V. The Guardian for the Overtake Scenario

Together, a library of RCIS for each intention in IV-B and an intention estimation module define the guardian for the overtake scenario. So, this section begins by discussing guarantees and methods for constructing a library of RCIS. Then, an intention estimation module is formally defined. Finally, we prove that integrating these two parts provides safety and is less conservative than previously considered models.

A. Library of RCIS

An RCIS can be constructed using any of the methods described in Section II-A. Specifically, we leverage the inside-out algorithm of [15] to compute an RCIS for each intention model $I_j \in \mathcal{I}$. The reader can recall that the inside-out algorithm uses an initial RCIS and expands it to obtain a final RCIS. One fact that we can use to generate such an initial, simple RCIS is given as follows:

Proposition 1. The set $C_{left} \doteq \{q \in Q \mid y_e \in [0.9; 2.7]\}$ of states corresponding to the left lane is an RCIS for any intention.

The proposition is stated without proof because the lead car cannot move laterally (i.e., it cannot change its y position in the lane); thus, the proposition immediately follows from the model definition.

Given this proposition, one can apply the inside-out algorithm by setting the `left lane' states as the initial RCIS, i.e., $C_0 = C_{left}$, for any of the intention models discussed in Section IV-B. A more involved, but helpful result that can be used to ease computation is:

Proposition 2. Any set $C_{bnd} \subseteq Q_{safe}$ that is a controlled invariant set for the bounded velocity model is also a controlled invariant set for the aggressive and the cautious driver intention models.

Proof (sketch). While the acceleration of the lead vehicle $a_{L;x}$ has a specified bound for the aggressive and the cautious driver intention models, the bounded velocity model has no such bound on the lead vehicle's acceleration (i.e., the lead car may change its velocity arbitrarily fast). Thus, if it is possible to remain robustly safe in the bounded velocity model, then when the lead car's acceleration is more restricted than the bounded velocity model allows, it should be the case that the ego vehicle can remain safe in all states C_{bnd}. □

Thus, the previous two propositions can be used to synthesize a set of RCIS, corresponding to each of the intention models described in Section IV-B $\{C_{bnd}; C_a; C_c\}$. Specifically, one can use Proposition 1 to identify the left lane as the initial RCIS, i.e., set $C_0 = C_{left}$, and apply the inside-out algorithm for the bounded velocity model to obtain C_{bnd}. After that, the resulting set C_{bnd} can be used as the initial RCIS for the inside-out algorithm according to Proposition 2, for each of the two intentions. Each of these RCISs induces a supervisor. For instance, for $i \in \{a; c\}$, we have $S_{I_i}(q) = \{u_e \in U_e \mid f_i(q; u; w) \in C_i; \forall w \in W; \forall u_o \in I_i(q)\}$. And, S_I is defined similarly from C_{bnd}. Moreover, these supervisors by construction satisfy the following:

Proposition 3. $S_I \subseteq S_{I_i}$ and $C_I \subseteq C_{I_i}$ for $i \in \{a; c\}$.

B. Intention Estimation

Intention estimation techniques can roughly be categorized into two categories: active [7], [9] and passive [12], [14] methods. The former assumes that the intention estimation method can modify the controller's commands. The latter, on the other hand, assumes that the intention estimation module cannot modify control signals and must perform the discrimination operation using the observations gathered by the sensors. Our guardian architecture uses a passive intention estimation scheme to allow maximal permissiveness and to avoid violation of any safe input constraints.

Given a state-input trajectory $qu^t_e = \{(q^0; u^0_e); \ldots; (q^m; u^t_e)\}$ and two intention models $\mathcal{I} = \{I_a; I_c\}$ as in Section IV-B, intention estimation aims to determine whether or not the state-input trajectory is consistent with model $I_i \in \{a; c\}$. This problem can be posed as a linear program at each time

t, similar to [10]:

$$\begin{aligned} \text{nd} \quad & \{u_o^k, w^k\}_{k=\max(t-N,0)}^{t-1} \\ \text{s.t.} \quad & \text{for all } k \in \{\max(t-N,0), \dots, t-1\} \\ & q^{k+1} = f_i^j(q^k, u^k, w^k) \text{ if } q^k \in D_i^j, \\ & u_0^k \in I_i(q^k) \text{ and } w^k \in W \end{aligned} \tag{LP$_i^t$}$$

where N is a horizon to keep the estimator of nite memory. Note that, infeasibility of LP$_i^t$ implies that the intention model is not I_i. Therefore, the estimator E is de ned as:

$$E(qu_e^{t-1}) = \begin{cases} I_a, & \text{if } E(qu_e^{t-2}) = I_a \\ & \text{or LP}_c^t \text{ is infeasible,} \\ I_c, & \text{if } E(qu_e^{t-2}) = I_c \\ & \text{or LP}_a^t \text{ is infeasible,} \\ I, & \text{otherwise} \end{cases} \tag{15}$$

C. Putting things together

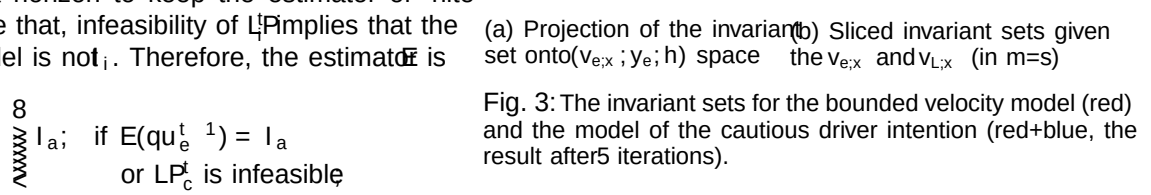
Having designed a library of RCIS and the intention estimation module, at run-time, we initialize the estimated intention for the intention-aware supervisor as the bounded velocity model, i.e., $I_v^0 = I$. As the intention estimation model E re nes the valid intention models I_v by collecting data, the intention-aware supervisor is updated accordingly.

Theorem 1. Assume that the intention of the other vehicle is not changing with time (i.e., I_i is constant for the driving scenario) and $I_i \in I = \{I_a, I_c\}$. If $q^0 \in C_{bnd}$ and $u_e^t \in S_{I_v^t}(q^t)$ for all t where $I_v^t = E(qu_e^{t-1})$, then we have $q^t \in Q_{safe}$ for all t.

Proof. First note that the linear program (LP$_i^t$) will always be feasible for $i = i$ as we assume I_i is constant over time. Therefore, $u_{e,t} \in I_v^t$ for all t. The intention estimation is initialized with I. By construction, $S_I(q^0) \neq \emptyset$ for all $q \in C_{bnd}$. Now, assume that the intention estimation module never detects the correct intention (i.e., $I_v^t = I$ for all t). Since $S_I(q^0) \neq \emptyset$, it follows from Def. 1 by induction that $S_{I_v^t}(q^t) \neq \emptyset$ and $q^t \in C_{bnd} \cap Q_{safe}$ for all t. Now, assume that intention estimation module eventually reveals the true intention I_i, i.e., there exists a t such that $I_v^t = I_i$. We know that the state of the system is safe ($q^t \in C_{bnd} \cap Q_{safe}$) for $t < t$ by using S_I. Moreover, by Proposition 3, at time t, $S_{I_i}(q^t) \subseteq S_I(q^t) \neq \emptyset$ and $q^t \in C_{bnd} \cap C_i$. By Eq. (15) and the assumption on constant intention, we will have $I_v^t = I_i$ for all $t \geq t$. Now, again, it follows from Def. 1 by induction that $S_{I_v^t}(q^t) \neq \emptyset$ and $q^t \in C_i \cap Q_{safe}$ for all $t \geq t$. □

VI. Results

In this section, we discuss the results of the proposed solution to Problem 1 for the driving scenario presented in Section IV. We brie y describe the tools and methods used to implement the invariant set algorithms. We then illustrate the intuitive conclusions that can be made about the RCIS and safe (admissible) input sets of various estimated intentions.



(a) Projection of the invariant set onto $(v_{e,x}, v_e, h)$ space (b) Sliced invariant sets given the $v_{e,x}$ and $v_{L,x}$ (in m/s)

Fig. 3: The invariant sets for the bounded velocity model (red) and the model of the cautious driver intention (red+blue, the result after 5 iterations).

A. Implementation and Experimental Setup

We use the inside-out method described in Section II-A to compute RCIS and safe input sets. We use polyhedra (or union of polyhedra) representation of sets in our algorithm, since it forms a closed class of objects under set operations such as intersection and projection. The code is implemented on top of the Multi-Parametric Toolbox 3.0 (MPT3) [11], a MATLAB toolbox with ef cient implementations of polyhedra manipulations. The system dynamics, intention models and the safety requirements are as stated in Section IV.

B. RCIS Computation Results and Discussion

We rst compute an RCIS for the bounded velocity model. The seed set for the inside-out algorithm is chosen as the left lane, i.e., $C_0 = C_{left}$, which is shown to be robust controlled invariant in Proposition 1. The algorithm converges in 12 iterations and the resulting RCIS is shown as the red regions in Figures 3a and 3b.

Due to Proposition 2, RCIS for the bounded velocity model is also robust controlled invariant for the other intentions. Thus, we initialize the inside-out algorithm with this new seed in the following computations. The resulting set after 5 iterations for the cautious driver intention model is shown as the union of the red and blue regions in Figures 3a and 3b. The blue region indicates the difference between the RCIS of the cautious driver and the bounded velocity model. The results show that, by estimating the intention model, we indeed have a larger invariant set. On the other hand, RCIS obtained for the aggressive intention is almost visually indistinguishable with the invariant set for the cautious intention, but as can be shown in Figure 4, their sets of admissible inputs corresponding to the same state can be different.

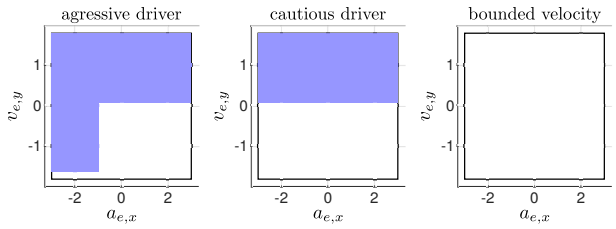


Fig. 4: Safe inputs (blue regions) at state $[25;\ 0.297;\ 16.52;\ 20]^\top$ for aggressive driver intention (left), cautious driver intention (middle) and bounded velocity model (right).

Note that, as shown in Figure 4, the safe input set can be non-convex. In that case, the projection to each dimension can be done in an order, according to a user defined priority. For example, speed change may be perceived as less "invasive" compared to a steering angle change from the human user perspective. In this case, projection onto the throttle input space may be preferred over the projection onto the steering input space.

C. Overtaking Simulation

We perform an overtaking simulation in MATLAB to show how the ego car and the lead car behave with and without the supervisor, with a baseline switched MPC controller for the ego car that is chosen to mimic a human driver that undertakes the overtaking task. In the supervised case, the supervisor is implemented using the controlled invariant sets obtained by our proposed algorithm. On the other hand, the lead car behaves according to one of the two intentions. To view the simulation videos, please refer to our YouTube channel [4].

Figure 5a shows the MPC control inputs over time in the simulations with no supervision for the case where the lead car driver is cautious. The red lines show the MPC inputs and the blue shadow shows the safe range of throttle/steering inputs (obtained by slicing the safe input set at each time) given the user-selected steering/throttle inputs. The region without blue shadow corresponds to the time when the ego car is out of the invariant set, since no supervision is applied. In Figure 5a, the blue shadow in the second row covers more time steps than the first row, which implies that the invariant set for the cautious driver intention contains more states than the invariant set for the bounded velocity model. Therefore, once the intention of the lead vehicle is discovered (shown by cyan vertical dashed lines), the supervisor will behave less conservatively (i.e., will allow more user-selected inputs) by switching to the supervisor for the estimated intention. This is indeed the case, as can be seen in Figure 5b, where the intention estimation and the guardian/supervisor are engaged.

[4] https://tinyurl.com/y69w989x

(a) MPC without supervision

(b) MPC with supervision

(c) Human driver without supervision

Fig. 5: The control inputs (red lines) of the ego vehicle over time (in seconds) for the following scenarios with and without supervision: ego car tailgates the lead car for a few seconds and then overtakes. The ego car in (a), (b) is controlled by an MPC controller, but in (c) is controlled by a human driver using the vehicle simulator in Figure 6. The lead car has cautious intention. The blue lines and shadow label the range of safe inputs given by the invariant sets. The cyan dash line labels the time when the intention estimation gives the correct intention. The green line in (b) labels the time when the ego car's inputs are overridden by the supervisor. The safe input ranges in the first and second rows in (a), (c) are computed with respect to the bounded velocity model and the cautious driver intention model respectively.

In the YouTube video list, Simulation 1 shows the animation that compares the results in Figure 5a and 5b. The same scenario with the aggressive intention is shown in Simulation 2. In addition, in the videos of Simulations 3 and 4, MPC is tuned to mimic a safe driver and a "*bad*" driver (more likely to crash with the lead car), respectively. Simulation 3 shows how such a "bad" driver crashes into the lead car in this scenario, but with supervision the driver is prevented from causing a crash. Furthermore, experimental results in Simulation 4 suggest that if the ego driver is already very careful, e.g., always keeping a safe distance with the lead car, the supervisor rarely needs to override.

D. Results from Driving Simulator

We also collected data using a driving simulator, where a human-driver is asked to perform an overtaking maneuver as described in the previous subsection.