2019 IEEE Intelligent Vehicles Symposium (IV)
Paris, France. June 9-12, 2019

Specifying Safety of Autonomous Vehicles in Signal Temporal Logic

Nikos Aréchiga[1]

***Abstract*— We develop a set of contracts for autonomous control software that ensures that if all traffic participants follow the contracts, the overall traffic system will be collision-free. We express our contracts in Signal Temporal Logic (STL), a lightweight specification language that enables V&V methodologies. We demonstrate how the specification can be used for evaluation of the performance of autonomy software, and We provide preliminary evidence that our contracts are not excessively conservative, i.e., they are not more restrictive than existing guidelines for safe driving by humans.**

I. Introduction

In order to attain public confidence in the safety of autonomous vehicles, it is important to develop industry-wide specifications on the minimal characteristics of autonomous vehicles. These specifications must be agnostic of the concrete implementation of the software of the autonomous vehicle, and it must enable the use of V&V processes, so that engineering firms can ensure that their designs meet the requires specifications.

We develop a set of contracts on the input-output behavior of an autonomy stack. These contracts are expressed formally in Signal Temporal Logic (STL), a lightweight specification language that enables a range of V&V activities, ranging from systematic testcase generation, falsification, and formal verification. Additionally, STL enables automatic synthesis of runtime monitors, which can be used as part of a regression test suite during the development of the autonomy sofware. STL monitors yield a quantitative metric of the *degree of satisfaction* of the requirement, which allows engineers to evaluate the relative performance of iterations of control software, as well as provide information for design trade-offs.

What is STL?

The most comprehensive recent work on developing safety specifications for autonomous vehicles is [13], which provides mathematical rules for assigning blame to vehicles involved in accidents depending on the actions of each vehicle leading up to the accident. These mathematical rules can be interpreted as contracts, namely, that each traffic participant should avoid taking actions that would cause it to be at blame. Our longitudinal contracts have similar structure to those of [13], but our lateral contracts are different to avoid problematic situations, including possible overshoot collisions. Furthermore, we take the further step of formalizing our contracts as STL specifications, which enable a variety of V&V activities, including automatic synthesis of runtime monitors, falsification, formal verification, and parameter synthesis.

mathematical rules = contracts

contract that participant should avoid to be at blame

2 different points between this paper and others

[1]Nikos Aréchiga is with Toyota Research Institute, 4440 El Camino Real, Los Altos, CA, USA nikos.arechiga@tri.global

The work of [12] develops a planner that generates "maneuver automata", and uses the theorem prover Isabelle to ensure that the maneuver automata satisfy specifications given in Linear Temporal Logic (LTL). Since LTL is primarily used to reason about discrete transition over automata, an additional reachable state computation is required to ensure that the continuous dynamics respect the high-level specification. In contrast, STL supports both discrete and continuous reasoning, which means that we do not need to separate discrete and continuous parts of the specification.

different between STL of this paper and LTL

The work of [14] considers the problem of ensuring safety for autonomous vehicles on a freeway by decomposing the freeway into Voronoi cells, and having the vehicles react to encroachment on their cells. However, the safety guarantees are tied to the specific planner implementation, and do not generalize to intersections or unstructure environments. In contrast, our approach provides input-output contracts that are agnostic of the specific autonomy stack, which allows us to use for safety testing and monitoring of proprietary autonomy systems.

The work of [9] presents machine-checked proofs for pairwise interactions between vehicles, including an additional manual proof that generalizes to arbitrarily many vehicles allowed to perform lane changes among arbitrarily many lanes. However, [9] does not generalize to intersections or unstructured environments. [2] and [1] consider the problem of verifying controllers with these non-deterministic policies. Similarly, [11] uses KeYmaera to prove collision-freedom of non-deterministic policies for robots, which are structurally similar to the rules of [13] as well as our own, but our STL-based contracts are more lightweight and easier to integrate into industry development processes. [1]

In this paper, we develop a set of safety contracts for autonomous vehicles in STL, and [2] show that the robustness semantics of STL can be used to provide natural monitoring capabilities for safety performance. In addition, the excellent existing tool support for STL means that the contracts can easily be [3] incorporated into the automotive development process. For example, falsification tools such as the one described in [5] can be used to find dangerous traces in which the vehicle violates basic safety conditions. Finally, we discuss the importance of avoiding excessive conservativeness in safety specifications, and conclude with directions for future work.

II. Background

A. Basic vehicle model

We present an abstract model of a vehicle as a body that may move laterally or longitudinally. Its position in cartesian

space is $p = (p_x, p_y)$, where p_x is the x-coordinate of the center of the vehicle and p_y is the y-coordinate of the vehicle. The velocity components of the vehicle are given by $v = (v_x, v_y)$ and its acceleration by $a = (a_x, a_y)$.

Additionally, we assume that the vehicle is rectangular, with width w and length ℓ. However, in what follows we will abstract this away and assume that distances between traffic participants are computed between the closest point of each participant.

what is global? how is it different from local vehicle

B. Coordinate transformation from global coordinates to local vehicle coordinates

In the local coordinates of the car, we define the *longitudinal* axis as the direction of travel, and the *lateral* axis as the direction that is perpendicular to the direction of travel.

If the vehicle has a heading in the direction of the (unit) vector $\hat{r}$, we can transform from global coordinates into local coordinates of the vehicle as follows. For any of the global quantities, position, velocity, and acceleration, we can transform it into lateral and longitudinal components. For example, the velocity vector $v = (v_x, v_y)$ can be transformed into $v = (v^{long}, v^{lat})$, such that $v^{long} = v \cdot \hat{r}$ and $v^{lat} = v - v^{long}$. Similarly, we can compute a^{long}, a^{lat}.

Why do the author mention the occlusion and perception systems in this part? What are the roles of those points

Reasoning about occlusions and perception systems is outside the scope of this work. A straightforward extension of our work to environments with occlusions would be to assume that there are always traffic participants behind all occlusions, but this is unsatisfactory. For example, suppose that the autonomous vehicle is driving along a residential street, and that there are parked cars along the side of the road. The na ive approach would be to assume that there are pedestrians behind *every* parked car, and that these pedestrians are ready to jump out at their maximum speed at any moment. This would result in excessive caution. A separate type of reasoning is required to ensure a reasonable level of safety in this scenario, possibly making use of context cues, such as proximity to a park and perhaps time of day—and certainly maintaining a low velocity. If future deployment of V2X systems becomes widespread, information about occluded pedestrians or cyclists could be obtained from the infrastructure or other vehicles. Similarly, specifications on perception systems are notoriously difficult. It is not sufficient to require a certain level of performance on a given test set—we need to ensure that the system is providing the right answers for the right reasons [7]. For the time being, however, the problem of occlusions and faulty perception are deferred to future work.

C. Parameters and state variables

We assume that the following parameters are given for acceleration and braking limits. These limits are *not* the physical limits of the car, but instead are parameters chosen at design time, as part of the specification of the autonomous software. They may be set by regulation or as an industry standard, and they may be different depending on the scenario. For example, a smaller value of μ may be used for extremely narrow lanes, and different acceleration bounds may be applied for highway and neighborhood driving.

- $\tau > 0$ is the end-to-end reaction time of the vehicle. This parameter encapsulates all possible delays between perception and actuation.
- $a^{long}_{max,accel} > 0$ is the maximal longitudinal acceleration.
- $a^{long}_{min,brake} > 0$ is the minimal longitudinal braking.
- $a^{long}_{max,brake} > a^{long}_{min,brake}$ is the maximal longitudinal braking.
- $a^{lat}_{max,accel} > 0$ is the maximal lateral acceleration.

Note that there is no such thing as "lateral braking". In the longitudinal case, braking is a deceleration that stops abruptly when the vehicle reaches zero longitudinal velocity, but there is no analogous concept in the lateral direction. There is only lateral displacement in one direction or the other, and if the vehicle applies a constant lateral acceleration in the opposite of its current lateral velocity, it will eventually be moving in the opposite direction rather than coming to a natural stop. For our chosen coordinate transformation, the right hand side of the vehicle is the positive lateral direction and the left hand side of the vehicle corresponds to the negative lateral acceleration.

D. Signal temporal logic

What is STL

Temporal logics are powerful tools to express specifications, and enable a wide array of V&V applications, including testcase generation, falsification, formal verification, and runtime monitoring. Applications for these languages include robotics [8], [15], [16], systems biology [3], and network planning [6]. STL [10], an extension to Linear Temporal Logic (LTL), enables specifications *real-valued signals* and can be applied to many continuous and hybrid systems. Automotive applications are naturally hybrid, since it involves continuous dynamics like vehicle states and trajectories as well as discrete dynamics, such as phases of a traffic light.

Form of STL

STL formulas are defined over predicates of the form $f(s) < c$, where s is a timed trace (signal), $f : \mathbb{R}^n \to \mathbb{R}$ is a function and $c \in \mathbb{R}$. STL formulas are written using the following grammar:

$$
\begin{aligned}
I &:= \ (a, b) \mid (a, b] \mid [a, b) \mid [a, b] \\
\phi &:= \ true \mid f(s) < c \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \\
&\quad \mid \Diamond_I \phi \mid \Box_I \phi \mid \phi \ \mathcal{U}_I \ \psi
\end{aligned}
$$

where $f(s) < c$ is a predicate, and the logical operators $(\neg, \wedge, \vee)$ have their usual meanings. In addition, $\Diamond$ (eventually), $\Box$ (always), $\mathcal{U}$ (until) and $\mathcal{T}$ (then) are temporal operators. We can achieve $f(s) > c$ by applying a negation on $f(s) < c$. The temporal operators have an associated time interval I where $0 \leq a < b$. For ease of notation, I is dropped from the grammar when $a = 0$, $b = \infty$.

STL Evaluaton

STL formulas are evaluated over time series data, packaged in a data structure that we call a *timed trace*. A timed trace s consists of an ordered sequence of states and their associated time, $s = (\mathbf{x}_0, t_0), ..., (\mathbf{x}_n, t_n)$ where $t_{i-1} < t_i$ and $\mathbf{x}_i \in \mathbb{R}^n$. To refer to the value of the trace at a given time, we use the notation, $s(t_i) = \mathbf{x}_i$. We use the

provide lots of equation without elaborraing their application and how they contribute to the paper

notations$s_i = (s, t_i)$ to refer to the *tail* of trace s, i.e. the trace that contains all of the same data s from time t_i onwards.

The *Boolean semantics* of STL formulas over a trace (s, t) are defined recursively as follows.

$$\begin{aligned}
(s,t) &\models f(s(t)) < c \quad &&\Leftrightarrow f(s(t)) < c \\
(s,t) &\models \neg\phi &&\Leftrightarrow \neg((s,t) \models \phi) \\
(s,t) &\models \phi \wedge \psi &&\Leftrightarrow ((s,t) \models \phi) \wedge ((s,t) \models \psi) \\
(s,t) &\models \phi \vee \psi &&\Leftrightarrow ((s,t) \models \phi) \vee ((s,t) \models \psi) \\
(s,t) &\models \Diamond_I \phi &&\Leftrightarrow \exists t' \in I \oplus t \;\; s.t. \; (s,t') \models \phi \\
(s,t) &\models \Box_I \phi &&\Leftrightarrow \forall t' \in I \oplus t \;\; s.t. \; (s,t') \models \phi \\
(s,t) &\models \phi \mathcal{U}_I \psi &&\Leftrightarrow \exists t' \in I \oplus t \;\; s.t. \; ((s,t') \models \psi) \wedge \\
& && \quad ((s,t) \models \Box_{[0,t']} \phi)
\end{aligned}$$

For a timed trace (s, t) starting at time t, satisfying $\Box\phi$ means ϕ is always true for the entire sequence (since the I is dropped, $I = [0, \infty)$). While satisfying $\Diamond\phi$ means at some time along the sequence, ϕ is true at least once. Since STL specifications are defined recursively, temporal operators can be composed with each other.

The robustness degree can be calculated recursively according to the *quantitative semantics*, which were originally introduced in [4]

$$\begin{aligned}
\rho(s,t,true) &= \rho_{\max} \\
\rho(s,t,f(s) < c) &= c - f(s(t)) \\
\rho(s,t,\neg\phi) &= -\rho(s,t,\phi) \\
\rho(s,t,\phi \wedge \psi) &= \min(\rho(s,t,\phi), \rho(s,t,\psi)) \\
\rho(s,t,\phi \vee \psi) &= \max(\rho(s,t,\phi), \rho(s,t,\psi)) \\
\rho(s,t,\Diamond_I \phi) &= \max_{t' \in I \oplus t} \rho(s,t',\phi) \\
\rho(s,t,\Box_I \phi) &= \min_{t' \in I \oplus t} \rho(s,t',\phi) \\
\rho(s,t,\phi \; \mathcal{U}_I \; \psi) &= \max_{t' \in I \oplus t} (\min(\rho(s,t',\psi), \\
& \qquad \min_{t'' \in [t,t']} \rho(s,t'',\phi)))
\end{aligned}$$

Note that the robustness value for strict and nonstrict inequalities is the same. We define a *robustness trace* to describe the robustness value of each timed trace subsequence. For a trace s starting at time t_0 and an STL formula ϕ, the robustness trace $\rho(s, t_0, \phi)$ is a sequence of robustness values for each subsequence s_i of s, given by:

$$\begin{aligned}
\rho(s,t_0,\phi) &= \rho_0, \rho_1, ...\rho_n \\
&= \rho(s,t_0,\phi), \rho(s,t_1,\phi), ..., \rho(s,t_n,\phi) \\
&= \rho(s_0,\phi), \rho(s_1,\phi), ..., \rho(s_n,\phi)
\end{aligned}$$

III. Pairwise Safety Contracts

Suppose two vehicles, c_1 and c_2 are driving on a route. The distance is considerd safe if the rear vehicle c_2 may accelerate or decelerate with any value $[a^{long}_{max,brake}, a^{long}_{max,accel}]$ as long as it is far enough that it can avoid a collision by braking in the next time step. By a straightforward calculation, this distance is

$$\begin{aligned}
d_{min} = \Bigg[& v_2\tau + \frac{1}{2} a^{long}_{max,accel} \tau^2 \\
& + \frac{(v_2 + \tau a^{long}_{max,accel})^2}{2a^{long}_{min,brake}} - \frac{v_1^2}{2a^{long}_{max,brake}} \Bigg].
\end{aligned}$$

To prevent collisions, we require that the lead vehicle c_1 must never brake more than $a^{long}_{max,brake}$, and it may not accelerate more than $a^{long}_{max,accel}$. This second restriction is not directly relevant to the pairwise interaction, but it is relevant when we consider that c_1 is in turn a rear vehicle to the car in front of it. If the distance is not large enough, c_2 must brake with a value in the range $[a^{long}_{max,brake}, a^{long}_{min,brake}]$. It is important that c_2 should not brake more strongly than $a^{long}_{max,brake}$, because c_2 may have a rear vehicle itself, and must uphold a contract of avoiding excessive braking to its own rear vehicle.

Theorem 1 (Longitudinal contract): If c_1 and c_2 are traveling in the same direction, the following contracts ensure that there are no collisions.

1) We assume that neither vehicle may reverse its direction and drive backwards, i.e. $v_1 \geq 0$ and $v_2 \geq 0$ for all time.
2) The acceleration of the lead vehicle must be within the longitudinal parameter bounds, $a_1 \in [a^{long}_{max,brake}, a^{long}_{max,accel}]$
3) If $p_1 - p_2 > d_{min}$, $a_2 \in [-a^{long}_{max,brake}, a^{long}_{max,accel}]$; otherwise, $a_2 \in [-a^{long}_{max,brake}, -a^{long}_{min,brake}]$.

This has been proven by a mechanical theorem prover in [9], as well as manually in [13].

The longitudinal contracts are given in STL below.

$$\Box(v_1 \geq 0 \wedge v_2 \geq 0) \tag{1}$$
$$\Box \; a_1 \in [-a^{long}_{max,brake}, a^{long}_{max,accel}] \tag{2}$$
$$\Box \; \Big(a_2 \in [a^{long}_{max,brake}, a^{long}_{max,accel}] \tag{3}$$
$$\wedge \;\; (p_1 - p_2 \leq d_{min} \tag{4}$$
$$\rightarrow a_2 \in [-a^{long}_{max,brake}, -a^{long}_{min,brake}] \Big) \Big) \tag{5}$$

A. Pairwise lateral contracts

We assume that vehicles will be controlled to stay within a margin $\mu/2$ from the edge of the lane. If this is observed, the minimum safe separation between two vehicles will be μ The lane-following controller of the autonomous vehicle is expected to perform well enough to maintain this margin, and the autonomous vehicle does not interpret another vehicle to be crossing a lane if it is within the margin.

At a high level, we expect that the lateral contract should say that we may merge if there is a "large enough gap" in the other lane. In order to provide latitude for the lane-following control, we would like to allow the controller to take any safe action. Naïvely, we might try to say that the controller may undertake any lateral acceleration $a^{lat}_{max,accel}$ away from the

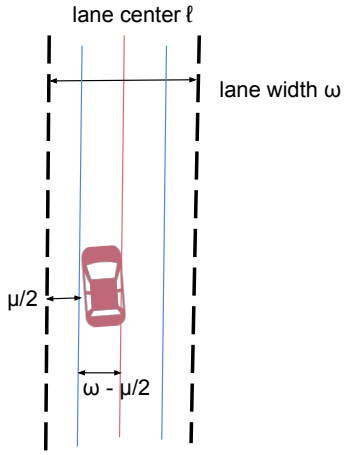


Fig. 1: Following a lane with margin $\mu/2$ from the edge.

Discuss downsides and provide recovery maneuver for the lateral case

center of its lane if it can avoid a lateral collision by, in the next time step, applying lateral acceleration $a^{lat}_{max,accel}$ towards the center of its lane. However, this formulation is not correct. The analogy with the longitudinal case does not hold, because "longitudinal deceleration" corresponds to braking, and will stop when the car reaches zero longitudinal velocity. This is not the case for lateral movement, since by applying $a^{lat}_{max,accel}$ for a duration of time, the car will not "stop" at zero lateral velocity. It is possible that, between two time steps, the car will actually find itself traveling in the opposite lateral direction, perhaps invading another lane on that side and causing a safety issue.

The correct recovery maneuver for the lateral case is to choose a lateral acceleration that is between the minimum required to return inside the margin $\mu/2$ and the maximum that would cause us to be on the edge of the margin on the other side in a single time step. Formally, the lateral distance between cars c_1, c_2 with lateral velocities v_1, v_2 is safe if during the time interval $[0, \tau]$, both cars move laterally towards each other with acceleration $a^{lat}_{max,accel}$, and then move laterally away from each other with an acceleration that is enough to return them to be μ-following their lanes in a time step of duration τ, but less than the acceleration that would take them out of their lanes on the opposite side. We define two functions, $a^{lat}_{min,away}$ to represent the lower bound, and $a^{lat}_{max,away}$ to represent the upper bound.

$$a^{lat}_{min,away} = \frac{2}{\tau^2}\left(v_{lat}\tau + p_{lat} - \omega - \frac{\mu}{2}\right),$$
$$a^{lat}_{max,away} = \frac{2}{\tau^2}\left(v_{lat}\tau + p_{lat} - \omega + \frac{\mu}{2}\right)$$

Lemma 1: The minimal safe lateral distance between c_1 and c_2 is:

$$d^{lat}_{min} = \mu + \left[\frac{2v_1 + \tau a^{lat}_{max,accel}}{2}\tau + \frac{(v_1 + \tau a^{lat}_{max,accel})^2}{2a^{lat}_{min,away}} - \left(\frac{2v_2 - \tau a^{lat}_{max,accel}}{2}\tau - \frac{(v_2 - \tau a^{lat}_{max,accel})^2}{2a^{lat}_{min,away}}\right)\right]$$

This is similar to Lemma 4 of [13], but the static parameter is replaced by a function that must be computed at runtime. More significantly, the safety contract requires an upper bound on the lateral acceleration, which prevents violating the margin on the other side of the lane.

We can give the lateral contract in STL as follows. Without loss of generality, we give the contract from the point of view of c_1, since in this case the contract is the same for both vehicles.

$$\square\ \left(a_1 \in [-a^{lat}_{max,accel}, a^{lat}_{max,accel}]\right. \quad (6)$$
$$\wedge\ \left(p^{lat}_1 - p^{lat}_2 \leq d^{lat}_{min}\right. \quad (7)$$
$$\rightarrow a_2 \in [a^{lat}_{min,away}, a^{lat}_{max,away}])) \quad (8)$$

IV. Unstructured Environments

As a vehicle moves through an unstructured environment, such as a parking lot, it is able to move both lontigudinally and laterally. When the vehicle chooses accelerations in a given time step, it will not be able to change its choice until the next timestep. We want to characterize the set of positions that the vehicle could reach if it makes the worst possible choice in this time step, and then recovers in the next. This set of positions, which we will call an *acceleration funnel*, will allow us to build a contract on the input-output relation of the autonomy stack, without having to know the detailed mechanics of how it makes its decisions. This type of input-output contract is well-suited to being used in situations in which the software is proprietary.

To simplify the presentation, we will define an *acceleration funnel* as the set of positions that a traffic participant can occupy, assuming that:

1) the participant chooses any longitudinal acceleration, $a^{long} \in [-a^{long}_{max,brake}, a^{long}_{max,accel}]$ for duration τ, and then brakes longitudinally with $a^{long} = -a^{long}_{max,brake}$ until coming to a complete stop.
2) the participant chooses any lateral acceleration $a^{lat} \in [-a^{lat}_{max,accel}, a^{lat}_{max,accel}]$ for duration τ, and then applies $a^{lat} = 0$ until the longitudinal braking described above brings it to a complete stop.

We use the notation AF_i to denote the acceleration funnel of vehicle i. A straightforward computation shows that the furthest longitudinal distance covered by the acceleration funnel is

$$x^{long} \leq \left(\frac{\tau^2 a^{long}_{max,accel}}{2} + \tau v^{long}\right)\left(\frac{a^{long}_{max,accel}}{a^{long}_{max,brake}} + 1\right) + \frac{(v^{long})^2}{2a^{long}_{max,brake}} + p^{long}$$

and the lateral distance covers the range

$$a^{lat} \leq \tau^2 \frac{a^{lat}_{max,accel}}{2} + v^{lat} + p^{lat} + \tau \left((a^{long}_{max,accel}\tau + v^{long}) \frac{a^{lat}_{max,accel}}{a^{long}_{max,brake}} + v^{lat} \right)$$

$$a^{lat} \geq \tau^2 \frac{-a^{lat}_{max,accel}}{2} + v^{lat} + p^{lat} + \tau \left((a^{long}_{max,accel}\tau + v^{long}) \frac{-a^{lat}_{max,accel}}{a^{long}_{max,brake}} + v^{lat} \right)$$

The contract between any pair of vehicles i and j can be expressed in STL as

$$\square((AF_i \cap Af_j) \rightarrow (a_i^{long} = -a^{long}_{max,brake}) \wedge (a_i^{lat} = 0) \wedge (a_j^{long} = -a^{long}_{max,brake}) \wedge (a_j^{lat} = 0)).$$

For any two traffic participants, if their acceleration funnels do not overlap, then they may safely take any action, since at the next time step they would be able to recover if needed. Note that we have used the set intersection notation to summarize inequalities over the states of the vehicles to shorten the presentation.

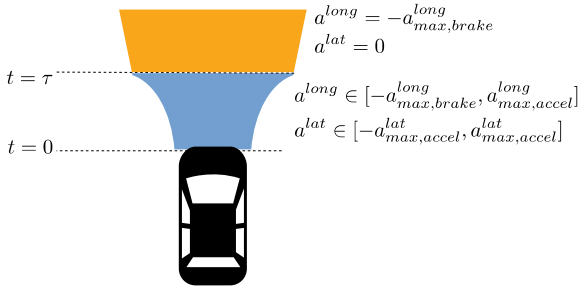


Fig. 2: Acceleration funnel

For any pair of traffic participants i and j, if their acceleration funnels do not intersect, then they do not have any contract obligations with each other. Otherwise, they must both apply $a^{long} = -a^{long}_{max,brake}$ and $a^{lat} = 0$.

V. Intersections

When approaching an intersection, one of the routes may or may not have priority. Examples of priority include having a green light at a singalize intersection, having the right-of-way at a lane merge, or having arrived first at a stop sign.

Intuitively, we expect that the vehicle that has the right of way should be allowed to proceed through the intersection, and that a traffic participant that does not have the right of way must yield, according to the rules of the route in question (e.g., at a red light, the vehicle must stop until the light changes, etc). However, in order to ensure safety, the autonomous vehicle should make a minimal effort to defend against a vehicle that is violating priority. This is a point of some contention in the literature on safety of intersections. For example, the work of [9] assumes that a vehicle never needs to be prepared for a red-light violation, whereas [13] assumes that we should take "minimal evasive action". Here, we ask that the autonomous vehicle should yield to a violating vehicle if the violating vehicle cannot prevent a crash by braking. In other words, if we are the only vehicle that can prevent a collision, we should yield, even if we have right of way.

Formally, suppose that vehicle c_1 has the right of way, and suppose that c_2 is nearby. We compute the reachable states of c_2 by a braking maneuver and zero lateral acceleration, as the set BM:

$$p^{long} \in BM \equiv \left[p_2^{long}, \frac{(v_2^{long})^2}{2B} + p_2^{long} \right].$$

We consider the intersection of the acceleration funnel of c_1, AF_1, and its set intersection with the intersection, $AF_1 \cap IX$. If this set is disjoint from c_2's braking maneuver, c_1 may proceed, otherwise he must stop. Formally,

$$\square(priority_1 = 1) \rightarrow (AF_1 \cap IX \cap BM = \emptyset \wedge a_1^{long} \in [-a^{long}_{max,brake}, a^{long}_{max,accel}]) \vee (a_1^{long} = -a^{long}_{max,brake}).$$

VI. Discussion

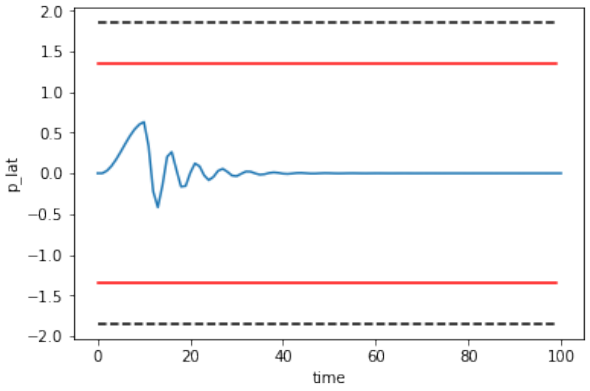
A direct implementation of the semantics of the operators of STL results in a powerful tool for automatically generating requirements monitors. Figure 3 shows the robustness trace of a lateral position constraint on synthetic data.

An important consideration to take into account when developing safety requirements is that they should not be excessively conservative, i.e., they should not be so restrictive that it becomes difficult to satisfy performance requirements. Specifically in the case of autonomous driving, if the behavior of the autonomous vehicle is excessively safe, it will not be able to successfully negotiate its way through traffic in which many vehicles are still controlled by humans.

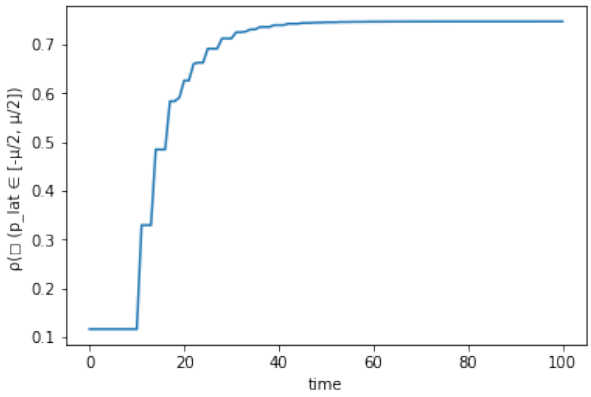
How do we measure the level of safety the behavior of the autonomous vehicle

One issue of particular concern is the following distance between vehicles in traffic. If the following distance is excessively large, it may cause human drivers to become impatient and aggressive towards the autonomous vehicle. The parameters of our longitudinal contract can be set to different values to tune the level of conservativeness, and we show that it is possible to establish reasonable values that allow appropriate driving behavior. For example, consider two vehicles driving at a typical highway speed of 65 mph ($\approx 29m/s$). By setting the reaction time $\tau = 0.5s$, the maximum acceleration allowed $a^{long}_{max,accel} = 2m/s^2$, and the maximum braking force to $a^{long}_{max,brake} = 8m/s^2$, we find that the minimum longitudinal following distance is 18.5 meters, which corresponds to a time-headway of $0.6s$. This is significantly less than the two-second rule that is recommended for defensive driving, which demonstrates that for reasonable choices of parameters, we are able to obtain appropriate performance.

Factors that contribute to the magnitude of this following distance include the maximum allowed acceleration,



(a) Synthetic data of a car driving in lane. The red lines are the $\mu/2$ margin, and the black dotted lines are the lane boundaries.



(b) Robustness trace of the STL formula $\square(p_{lat} \in [-\mu/2, \mu/2])$

Fig. 3: Monitoring an STL formula over synthetic data

$a^{long}_{max,accel}$, the reaction time τ, and the maximum braking $a^{long}_{max,brake}$. Other longitudinal contracts in the literature, including [9] and [13] make a distinction between the maximum braking force allowed by the lead vehicle and the maximum braking force required of the follower vehicle. We find that the difference between these braking forces is the largest single contributor to conservativeness. For example, maintaining the same parameter values as before, but requiring the follow vehicle to only brake at least as strong as $4.5m/s^2$ we obtain a following distance of approximately 52 meters, corresponding to a time-headway of 1.8 seconds which is slightly less than the two-second rule. Note that this paper has not provided the expression for minimum longitudinal distance under different braking requirements for lead and follow vehicles, but this quantity is well-known in the literature, for example in both [9] and [13].

VII. Future work

In this work, we have provided a set of contracts that will ensure safety of autonomous vehicles that satisfy them, relying on proofs constructed manually as well as in the existing literature. However, proofs constructed by a human are fallible, and in future work we propose to formally verify our contracts via a machine-generated proof. Furthermore, we will develop software infrastructure to check that requirements on systems and subsystems of the autonomous vehicles satisfy these higher-level contracts.

We have also provided preliminary evidence that our contracts are not excessively restrictive by demonstrating that the following distance they prescribe is not significantly larger than a following distance recommended for defensive driving by humans. In future work, we want to use naturalistic driving data to validate whether the contracts can fit safe behaviors, or whether modifications are requires.

In addition, specification languages are often seen by engineers as difficult to understand. We plan to work with engineering teams at our organization to improve the usability of the language, and to develop software infrastructure that will simplify its use and accelerate its adoption.

References

[1] N. Arechiga and B. H. Krogh. Using verified control envelopes for safe controller design. In *American Control Conference (ACC)*, 2014.

[2] N. Aréchiga, S. Loos, A. Platzer, and B. Krogh. Using theorem provers to guarantee closed-loop system properties. In *American Control Conference (ACC)*, 2012.

[3] E. Bartocci, L. Bortolussi, and L. Nenzi. A temporal logic approach to modular design of synthetic biological circuits. In *International Conference on Computational Methods in Systems Biology*, 2013.

[4] G. E. Fainekos and G. J. Pappas. Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science*, 2009.

[5] G. E. Fainekos, S. Sankaranarayanan, K. Ueda, and H. Yazarel. Verification of automotive control applications using S-TaLiRo. In *American Control Conference*, 2012.

[6] S. Karaman and E. Frazzoli. Linear temporal logic vehicle routing with applications to multiuav mission planning. *International Journal of Robust and Nonlinear Control*, 21(12):1372–1395, 2011.

[7] P. Koopman and M. Wagner. Toward a framework for highly automated vehicle safety validation. In *SAE World Congress*, 2018.

[8] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas. Temporal-logic-based reactive mission and motion planning. *IEEE Transactions in Robotics*, 25(6):1370–1381, 2009.

[9] Sarah M. Loos, André Platzer, and Ligia Nistor. Adaptive cruise control: Hybrid, distributed, and now formally verified. In Michael Butler and Wolfram Schulte, editors, *FM*, volume 6664 of *LNCS*, pages 42–56. Springer, 2011.

[10] O. Maler and D. Nickovic. Monitoring temporal properties of continuous signals. In *FORMATS/FTRTFT*, 2004.

[11] S. Mitsch, K. Ghorbal, and A. Platzer. On provably safe obstacle avoidance for autonomous robotic ground vehicles. In *Robotics: Science and Systems (RSS)*, 2013.

[12] A. Rizaldi, F. Immler, B. Schurmann, and M. Althoff. A formally verified motion planner for autonomous vehicles. In *International Symposium for Automated Technology for Verification and Analysis (ATVA)*, 2018.

[13] S. Shalev-Shwartz, S. Shammah, and A. Shashua. On a formal model of safe and scalable self-driving cars, 2018.

[14] M. Wang, Z. Wang, S. PAudel, and M. Schwager. Safe distributed lane change maneuvers for multiple autonomous vehicles using buffered input cells. In *International Conference on Robotics and Automation (ICRA)*, 2018.

[15] T. Wongpiromsarn, U. Topcu, and R. M. Murray. Receding horizon temporal logic planning. *IEEE Transactions on Automatic Control*, 57:2817–2830, 2012.

[16] T. Wongpiromsarn, U. Topcu, N. Ozay, H. Xu, and R. M. Murray. TuLiP: A software toolbox for receding horizon temporal logic planning. In *Hybrid Systems: Computation and Control*, 2011.