Automated Vehicle Safety Guarantee, Verification and Certification: A Survey

Tong Zhao, Member, IEEE, Ekim Yurtsever, Member, IEEE, Joel A. Paulson, Member, IEEE and Giorgio Rizzoni, Fellow, IEEE.

Abstract—Challenges related to automated driving are no longer focused on just the construction of such automated vehicles (AVs), but in assuring the safety of their operation. Recent advances in Level 3 and Level 4 autonomous driving have motivated more extensive study in safety guarantees of complicated AV maneuvers, which aligns with the goal of ISO 21448 (Safety of the Intended Functions, or SOTIF), i.e. minimizing unsafe scenarios both known and unknown, as well as Vision Zero – eliminating highway fatalities by 2050. A majority of approaches used in providing safety guarantees for AV motion control originate from formal methods, especially reachability analysis (RA), which relies on mathematical models for the dynamic evolution of the system to provide guarantees. However, to the best of the authors’ knowledge, there have been no review papers dedicated to describing and interpreting state-of-the-art of formal methods in the context of AVs. In this work, we provide both an overview of the safety verification, validation and certification process, as well as review formal safety techniques that are best suited to AV applications. We also propose a unified scenario coverage framework that can provide either a formal or sample-based estimate of safety verification for full AVs. Finally, remaining challenges and future opportunities beyond the scope of current published research for assured AV safety are presented.

Index Terms—Automated vehicle, Verification and validation, Reachability analysis, Safety guarantees, Motion planning, Safety of the intended function, Scenario coverage

I. INTRODUCTION

Automated vehicle (AV) technology is expected to be massively deployed in the next decade or two, with multiple SAE J3016 level 2 automated driving systems developed and deployed over the last two decades[1], [2], and some level 3 system regionally approved and announced[3], [4]. As a daily means of transportation for humans and goods, more rigorous safety guarantee has to be investigated and eventually implemented in the automated fleet that will dominate the road traffic around 2050[5]. Recent poll suggests that safety is the primary concern for AV acceptance, compared to economic impact or privacy concern[6]. And it is believed that by mass adoption of automated vehicles, road safety will be significantly improved compared to the road traffic status quo[7]. The gap between this incoming autonomous future and current state-of-art safety measures is to be filled by continued effort of scientists and engineers, as well as legislators.

Previous surveys on AV technologies have been focusing on the various aspects of the technical challenges and implementation, such as ways to perform perception, decision making, vehicle control, human machine interaction[8], with a few reviews covering the issue of AV safety verification and validation[9], [10], [11], [12], [13], [14], [15], [16], [17].

With the rise of real word deployment of ADS systems, significant amount of verification and validation (V&V) is required against the scale of scenarios in real world that ADS will encounter. A common strategy of V&V to maximize scenario coverage is to verify ADS in virtual simulations and simulated large amount of generated scenario samples (see [14], [12], [17] for detailed reference).

The challenge in scenario sampling-based V&V is quantifying the amount of sample required to guarantee reasonable coverage, so that the risk created by ADS improper actuation can be curbed. A reasonable scenario coverage guarantee is also the prerequisite for type approval in recent legislation for highly automated vehicle (HAV) deployment[18], [19]. An alternative set of methods based on formal verification however, address the scenario coverage from a different approach through specification satisfaction[15]. Because of the potential of formal methods in scenario coverage, emerging research have started to combing formal properties with control synthesis[20], [21] and safety verification[16].

In light of the need for ADS quantifiable verification, we propose a unified scenario coverage framework to address the unresolved pain point in AV scenario coverage. We provide a quantitative definition of unified safe scenario coverage given an accepted coverage-representation volume of a single scenario (for sample-based methods), or given so called “specification penetration rate” of formal safety specifications (for formal methods). This quantitative definition unites sample-based methods and formal methods in the same quest to provide scenario coverage for safety verification. A candidate AV control policy can go through either formal or sample-based safety verification to be tested of its safe scenario coverage in a specified ODD. We compare the pros and cons of sample-based methods and formal methods based on how they reach this safety coverage and the associated costs and issues with each method.
The contributions of this paper are three-fold: (1) we provide a holistic overview of different aspects in ADS safety guarantee, verification, validation and certification; (2) we provide a unified safe scenario coverage framework for both sample-based and formal methods for safety verification; (3) we conduct a state-of-the-art literature review on formal methods for automated vehicle safety, and we categorize how in different ways formal methods (especially reachability analysis) can be used to provide safety verification or guarantee for motion control. Lastly, we point out trends and opportunities in the field of formal safety verification for ADS.

II. Certification

A. Certification Status Worldwide

Since the piloting AV programs such as the PROMETHEUS project[42] (1987–1995) and DARPA Grand Challenge[43] (2004–2007), the development of automated vehicles have entered the era of on-road deployment. An incomplete summary of current certification of automated driving on public roads is provided in Table I.

B. Safety Certification Challenges

Among all the concerning issues, four particular challenges stand out for AV safety certification: the determination of unavoidable collision; the determination of liability; the verification cost for reasonable scenario coverage; and the additional cost of re-verification for ADS update.

Determining whether a collision situation is avoidable or not, this is not just an academic problem but also a liability-related legal problem. Despite the high expectations for autonomy, as humans we can only expect robotic systems to optimize actions for the best possible outcome. For example, one can only expect automated vehicles to apply evasive maneuver to avoid an avoidable collision, but not an unavoidable collision. The requirement for such avoidability determination has been passed as an amendment to law in Germany in August 2021 regarding level 3 to level 4 function in HAV[26]. Specifically, in §1e paragraph (2), number 2 of the law (Straßenverkehrsordnung, or Road Traffic Act) states that (English translation):

Motor vehicles equipped with an automated driving function must have technical equipment that is capable to: independently comply with the traffic regulations directed at driving the vehicle and have an accident prevention system that:
(a) is designed to avoid and reduce damage,
(b) in the event of unavoidable alternative damage to different legal interests, the protection of human life being the highest priority, and
(c) in the event of an unavoidable alternative risk to human life, none provides further weighting based on personal characteristics.

In the legal text above, judging whether “unavoidable alternative damage” is the actual case requires a formal determination of damage avoidability, i.e. collision avoidability. Despite the different implementations of accident prevention systems in industry, from a safety verification standpoint, the avoidability judgement mechanism is better conservative than risky. At the same time, it is also undesirable to be “too conservative” to not use emergency maneuver when an incoming collision is still avoidable. To illustrate this delicate difference, two situations are given:

1) First, when the vehicle is very close to an unavoidable collision situation yet there is still room to avoid the collision, we don’t want the ADS to make a false unavoidable judgement and start preparing for the impact;
2) Second, when the vehicle is actually in an unavoidable collision situation and not colliding yet, from a safety verification standpoint, the ADS should be sensitive enough to immediately start preparing for the impact.

These two situations indicate that the accident prevention system in ADS should be able to make a clear-cut distinction between avoidable and unavoidable situations early enough, and it should not hesitate to engage conservative maneuver, evasive maneuver or preparation for impact. Otherwise, at least according to the German law amendment[26], AV control is liable for collision damage.

Liability determination is also an emerging issue to be addressed for ADS deployment. At level 3 the ADS is not liable as long as unexpected or difficult situations are handed over early enough to the human driver, as the ADS is only expected to guarantee safety in its limited ODD environment. Even if the human driver does not take over in time, a fail-safe maneuver such as braking, stopping and turning on hazard lights can free the ADS from liability, to a certain extent. When a level 4 or above AV is equipped with formal safety verified ADS, the liability determination can be streamlined according to a formal logic, illustrated in Figure 1 for example. For ADS developed with sample-based safety verification, it is yet to see how a similar programmatic formal liability determination can be performed.

Another challenge for certification is the cost associated to verify and validate the ADS. For an ADS industry stakeholder, already huge investments have been made to verify and validate the in-house ADS. When it comes to type-approval or certification, it is not fully clear how much of the V&V effort from the industry stakeholder can be “credit transferred” towards the safety certification.

Moreover, as more learning-based functions are incorporated into AV control design, the resulting control stack is constantly evolving based on collected learning data. The continuous learning, together with the continuous development of ADS software by the engineering team is adding another layer of uncertainty to safety V&V and certification. A question remains unanswered on whether regression testing is required for each major or minor ADS software update.

III. Automated Vehicle Safety Verification and Validation

A. Qualitative Safety Goal

Safety is system state free from harm. And safety assurance is the capability to be free from risks developing into harm when they emerge[44]. Although huge safety enhancement potential exists for AVs as 94% of road accidents are driver-related[45], but limitations of current safety V&V measures for AVs are
widely recognized. As pointed out by RAND Corporation report *Driving To Safety* [46], tens of billions of miles of on-road testing is required to ensure acceptable level of AV safety for all possible scenarios and inputs. Whereas programmatically generated simulation tests, although performed much faster, more flexible in scenario generation [12] and more economical, are susceptible to fidelity and realism concerns [9]. Even worse, both on-road testing and simulated testing cannot exhaustively cover all driving scenarios even for a small operational design domain (ODD) [22].

The widely used safety guideline standard *ISO 26262 Road Vehicles—Functional Safety* [47] is suited only for mitigating known unreasonable risks related to known component failure (i.e. known unsafe scenario), but it is not addressing AV driving risks due to complicated environment variants and how the ADS reacts to them, while nothing in the vehicle is technically failing.

Given the safety challenge above, a qualitative goal was proposed in *ISO 21448 Safety of the Intended Functions (SOTIF)* [48], which describes high-level goals of minimizing known and unknown unsafe scenario outcomes for ADS function design (Figure 2). The challenge for meeting this goal, however, is the intractability of traditional methodologies such as field operational test (FOT) [13] to cover all possible scenarios for automated driving during testing.

Despite the challenge, there are promising methodologies and directions in safety analysis to move towards the *ISO 21448* goal, such as maximizing scenario coverage in simulated testing [12], or creating safety guarantees through formal methods [20, 21].

### B. Quantitative Safety Threshold

For AV testing, large number of tested miles can be deceptive and does not always equal to adequate scenario coverage. Specifically, if most of the test miles are done on repeatedly tested, low traffic highway roads, the “per mile value” of the test data quickly deteriorates. In fact, event rates per exposure notions such as injuries/fatalities per 100 million mile mentioned in the RAND report [46], or disengagements per 1000 miles used for California DMV disengagement report [49] are only rough criteria to provide an intuitive summary of how statistically safe is a certain operation, with a risk of selective bias.

In technical terms, *safe enough* usually indicates full or sufficient scenario coverage for the designated ODD [13, 14]. In reality, relatively weak requirements have been seen in existing regulations [19] that only require to check “a number of scenarios that are critical”. A promising attempt to assign “volume” to scenario samples is seen in [16] by introducing a “δ-neighborhood” around a discrete scenario. Mathematical algorithms such as T-wise [50], [51] and Poisson process [52] have also been adopted to achieve “almost full” statistical coverage by cleverly select candidates with limited samples.

Let us optimistically assume that this quantitative coverage representation problem is solved and accepted across the community, and each single sampled scenario has a “coverage volume” (illustrated in shaded areas in Figure 4 and in gridded units in Figure 3(f)-(j)). Then the full scenario coverage task for sample-based methods is performing sufficient amount of sampled verification tests to ensure that each “coverage volume” unit is associated with at least one safe test result. Realistically, some scenarios in a scenario space may be impossible to produce a safe test result (e.g. too little reaction distance with an obstacle), in this case, extra effort is required to confirm that those cases are indeed *safety infeasible*. When it comes to type approval and certification, the authorities need to set an acceptable successful threshold for in order to match up with public expectations. The simplest threshold $r$ could be the ratio between the number of different tests verified safe $N_{sv}$ and the number of minimum tests required for the ODD.

### Table I

| Region | Testing | Deployment Legislation Level 4 | Safety Driver Removal Deployment | Approved Deployment Case(s) |
|--------|---------|-------------------------------|---------------------------------|-----------------------------|
| US     | ✓ ✓     | J3016                         | Conditional***                   | Nuro (California, Dec 2020) |
|        |         |                               | Conditional****                  | Waymo, Cruise (California, Sep 2021) |
| EU     | ✓       | UNECE No.79, Apr 2021 Germany | Germany, 2022                    | Mercedes-Benz (Level 3 ALKS, Dec 2021) |
| UK     | ✓       | Spring 2022                    | -                               | -                           |
| China  | ✓       | General legislation, Apr 2021  | Conditional                      | Robotaxi w/ safety driver: Baidu, Pony.ai (Beijing, Nov 2021) |
| Japan  | ✓       | May 2020                       | With remote monitor              | Honda (Level 3 for congestion situation, Nov 2020) |
| Korea  | ✓       | Jan 2020                       | -                               | -                           |

* As of Jan 16, 2022, 32 of the 50 states in the US allow AV testing [39].
** As of Jan 16, 2022, 17 of the 50 states in the US allow deployment of AV, but to the authors’ knowledge, details for the level of autonomy classification and due responsibility are not explicitly specified [39].
*** As of Jan 16, 20 of the 32 states that allow AV testing do not have requirements of safety driver if certain conditions are met [39].
**** Cruise applied for driverless deployment in California in May 2021 [30], and received test permit in June 2021, then partial deployment approval in Sep 2021, waiting for final approval from California Public Utilities Commission [41].
When safety specification is expressed in axioms and lemmas and robust controlled invariance set[16], both sample-based and formal methods, as well as field operational tests, is necessary during the development stage to boost the prototyping and design error discovery. In fact, novel approaches trying to combine sample-based methods with formal set-based methods have been proposed using a combination of resolution or probabilistic sample completeness and robust controlled invariance set[16].

Compared to testing-based methods for safety verification and assurance, formal methods have the advantage of high statement reliability[14] due to the rigorous logic foundation of formal methods.

Commonly used formal methods in AV safety includes model checking, reachability analysis and theorem proving[15]. Model checking originates from software development to ensure that the software behavior is according to the design specifications. When safety specification is expressed in axioms and lemmas as in[55], then theorem proving can be used to verify safety using worst case assumptions. Reachability analysis[56] has a special place among the three due to its inherent capability of producing safety statements for dynamical systems[57], which capture the main characteristic of dynamic driving task (DDT).

Real world road testing, or field operational testing (FOT), is the ultimate yet expensive approach for AV verification and validation. In a certain sense it is the only way of validation. Yet the shortcomings for FOT is also obvious: it lacks the ability to reach enough scenario coverage (especially near collision and collision scenarios) with a candidate AV controller on board. A comparison of sample-based, formal and field operational testing methods is illustrated in Figure 5. The different aspects are scored on a scale of 0 to 10, with 10 representing highest satisfaction.
where state $x$ with continuous dynamics has been a major limiting factor that can lead to some target state set $\mathcal{X}\text{goal}(t_0)$, which back-propagates to find all possible states from a previous time $t_0$. Thus, given any set of initial states $X_{\text{init}}(t_0) \subseteq \mathcal{X}$, the forward reachable set at time $t > t_0$ of a set of target states $\mathcal{X}_{\text{goal}}(t_0)$ is defined as:

$$maxFRS : \{x(t) \in \mathcal{X} | \exists u \in \mathcal{U}, \exists x(t_0) \in \mathcal{X}_{\text{goal}}(t_0),$$

$$x(t) = x(t_0) + \int_{t_0}^{t} f(u,t) dt \}$$

(4)

The operator $\exists$ before $u$ in the above expressions can be replaced with $\forall$ to create the “minimal” reachable sets, for example, minimal backward reachable set:

$$minBRS : \{x(t) \in \mathcal{X} | \forall u \in \mathcal{U}, \exists x(t_0) \in \mathcal{X}_{\text{goal}}(t_0),$$

$$x(t) = x(t_0) + \int_{t_0}^{t} f(u,t) dt \}$$

(5)

In some cases, reachability within defined time horizon is of more interest. Thus the definitions in the previous section can be extended to include time from current time to end of time horizon. For brevity, only the definition of maximal forward reachable tube is provided, the other variants can be easily derived accordingly:

**Definition (Maximal Forward Reachable Tube):** given a dynamical system as in (2), the forward reachable tube during time $[t_0, t]$ of a set of initial states $X_{\text{init}}(t_0) \subseteq \mathcal{X}$ is defined as:

$$maxFRT : \{x(s) \in \mathcal{X} | \exists u \in \mathcal{U}, \exists s \in [t_0, t], \exists x(t_0) \in \mathcal{X}_{\text{init}}(t_0),$$

$$x(s) = x(t_0) + \int_{t_0}^{s} f(u,t) dt \}$$

(6)

In situations where the interaction among different traffic participants play a critical role, it becomes imperative to include the influence of participating, sometimes even adversarial agents into the reachability analysis. In this case, an additional input $d$ is introduced to the system (2) to represent such adversarial control input:

$$\dot{x} = f_a(x, u, d)$$

(7)

where $d \in \mathcal{D} \subseteq \mathbb{R}^p, p \in \mathbb{R}$ is the adversarial input to the system. As an example, it can be control decisions of other vehicles where $f_a : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}^n$ represents the joint or relative dynamics of all participating vehicles. In such systems, the influence of adversarial inputs is not up to the ego vehicle control, and therefore must be modelled conservatively to provide guarantees in worst cases. The definition of maxFRT and minBRT in such systems are provided below as examples.

**Definition (Maximal Forward Reachable Set under Adversarial Influence):** given a dynamical system as in (7), the forward reachable set at time $t > t_0$ of a set of initial states $X_{\text{init}}(t_0) \subseteq \mathcal{X}$ is defined as:

$$\{x(t) \in \mathcal{X} | \exists u \in \mathcal{U}, \forall d \in \mathcal{D}, \exists x(t_0) \in \mathcal{X}_{\text{init}}(t_0),$$

$$x(t) = x(t_0) + \int_{t_0}^{t} f(x, u, d) dt \}$$

(8)
The above STL-translated safety specification still stays at an abstract level in set language, and it is the responsibility of AV control algorithm designers to honor this safety specification by either synthesizing it into the controller architecture, or performing safety verification over the designed controller to ensure that the specification is met with enough confidence or solid proof. Some examples of STL-based safety specifications are: (1) always keep the ability to return to the right-most lane within 5 seconds; (2) maintain collision-free and within road boundaries. When a finite time horizon (such as $I = [0, 25](s)$) is provided, the specifications (1) and (2) can be expressed as:

$$ (s, t) \models \Box_{[0, 25]} \varphi_{\text{lane}} \land \Box_{[0, 25]} \varphi $$

where $\varphi_{\text{lane}}$ represents the specification of being always within 5 seconds of returning to the right lane, and $\varphi$ represents collision-free and within road boundaries. The expression $\Box_{[0, 25]} \varphi_{\text{lane}}$ encodes “always” (in 5 seconds)” “have the existing capability to satisfy in 5 seconds” ($\varphi_{\text{lane}}$) the “going back to right lane” requirement ($\varphi$). The expression $\Box_{[0, 25]} \varphi$ encodes “always” (in 5 seconds)” satisfy collision-free and road boundary constraints” ($\varphi$). More sophisticated control goal can be encoded based on this STL expression, for example, to indicate the task of overtaking a slow car in front (expression modified from [20]):

$$ (s, t) \models \Box_{[0, 25]} \varphi_{\text{lane}} \land (\varphi \land \Box_{[0, 25]} \varphi_{\text{pass}} \lor \varphi \land \Box_{[0, 25]} \varphi_{\text{stay}}) $$

where $\varphi_{\text{pass}}$ represents successful overtaking the slow front car, and $\varphi_{\text{stay}}$ represents staying behind the slow front car. The expression $(\varphi \land \Box_{[0, 25]} \varphi_{\text{pass}} \lor \varphi \land \Box_{[0, 25]} \varphi_{\text{stay}})$ specifies two alternative goals ($\varphi_{\text{pass}}$ or $\varphi_{\text{stay}}$) on top of the safety constraint $\varphi$. STL expressions like equation (11) were implemented into control synthesis via connection between STL and reachability analysis in some recent work [20]. Apart from control synthesis, safety specifications expressed in STL can be used as assertions during controller prototyping to indicate safety violations, so that developers are constantly aware of safety specifications during control design. The logic calculation for determining the truth of a safety specification is often done by solving a satisfiability modulo theory (SMT) problem.

C. The Formal Safety Philosophy

Unlike the sampling-based methods where the verification against scenario variations is done through populating large number of scenario samples, in formal verification, the verification is mostly done when the safety specification is implemented in the controller to the fidelity level of the simulated environment. This is due to the different mechanism of how safety is verified. In the formal safety philosophy, a safety specification is either fulfilled or violated, and the

---

1. First order logic: a collection of formal systems that uses quantifiables over non-logic objects such as variables. For example, while propositional logic can only make statements like “the vehicle violates lane boundaries”, first order logic can incorporate variables and state “there exists a coordinate $C$ such that $C$ is inside the vehicle occupancy area and $C$ is outside the intended lane boundaries $B$”. In the later statement, “exists”, “inside” are quantifiables, and $C$ is a variable.

2. Assertion: in computer science, an assertion is a predicate connected to a point in the program that should always evaluate to true if the program is implemented correctly.

3. Satisfiability modulo theory (SMT) problem: decision problem that returns true or false for formulae expressed using first order logic, for reference see [61], [62], [63], [64].
fulfillment property can be designed by synthesizing the specification fulfillment into model-based control design. After such design is complete, the burden of ensuring such fulfillment during operation is transferred to online validation of model correctness: as long as the control-oriented model is validated to be correct, and the controller executes by the synthesized safety specification, then the system is provably safe. Without loss of generality, a safety specification φ may not be feasible in certain situations (such as an unavoidable collision), ∀u, (s, t) ⊄ φ. In this case, the φ-synthesized controller cannot find a feasible control sequence to fulfill φ, and the best practice would be to promote the situation to premeditated emergency or fall-back strategy (such as collision impact preparation). In any event, since the φ-synthesized controller by design has exhausted its available actions and still cannot find a φ-fulfilling action, the controller is not at-fault for the unavoidable damage.

V. Unified Scenario Coverage Framework

A. Scenario Space and Scenario Volume

According to ISO 21448[43], scenario is a description of the temporal development between several scenes in a sequence of scenes, and scene is a snapshot of the environment including the scenery, dynamic elements, and all actor and observer self-representations, as well as the relationship among those entities. Complying to this definition, the task of ensuring full safety in a defined ODD can be described as performing safety verification and validation for all possible scenario variations within the ODD. The variations of scenario within an ODD is two-fold: variations of the initial scene, and variations of the temporal development since the initial scene. The different dimensions of scenario variation is illustrated in Figure 5.

Given the specific ODD designated as O, one can define an entity to span all possible scenarios within the ODD description. We call this the scenario space. Using parameterization, the scenario space is defined as:

Definition 1. (Scenario space) The N-dimensional sub-space \( S \subset \mathbb{R}^N \), where \( S = \{(q_1, \cdots, q_N, q_{j1}, \cdots, q_{jM}) \in \mathbb{R}^{N+M} \mid p_i \in [p_{iL}, p_{iR}], q_j \in Q_j \} \) is the scenario space for an ODD specified by the continuous scenario parameters \( p_i \) (i = 1, \cdots, N) and discrete scenario parameters \( q_j \) (j = 1, \cdots, M). \( p_{iL}, p_{iR} \in \mathbb{R} \) are the lower and upper bounds of each continuous parameter \( p_i \), \( Q_j \subset \mathbb{R} \) is the set of possible values for each discrete parameter \( q_j \in Q_j \).

Remark 1. Note that in such a parameterized definition, no description of stochastic behavior is explicitly given, since the parameters are used to capture the scenario event as if it has already happened and fully describable without stochasticity. Assume that the number of continuous dimensions of scenario space \( S \) is \( N \), the number of discrete parameter combinations is \( \prod_{j=1}^{M} |Q_j| \), where \( |Q_j| \) is the cardinal of parameter set \( Q_j \). Assume the feasible range of each continuous parameter \( p_i \) is \( [p_{iL}, p_{iR}] \), then the total scenario space volume can be defined as:

Definition 2. (Scenario space volume) The scalar value \( V_S(O) = \prod_{i=1}^{N} (p_{iR} - p_{iL}) \cdot \prod_{j=1}^{M} |Q_j| \) is the scenario space volume of the scenario space \( S(O) \) specified by the
1. The meaning of symbols are aligned with Definition 1.

**Full scenario coverage** for safety verification requires that the verified scenario space volume $V_v$ encompasses the volume $V_S(O)$:

$$ V_v \supseteq V_{S(O)} = \prod_{i=1}^{N} (p_i - p_i') \cdot \prod_{j=1}^{M} |Q_j| \quad (12) $$

where $V_v$ is safety verified scenario volume.

Similarly, **partial scenario coverage** indicates that safety verification is carried out only on a portion $V_v'$ of the scenario space volume $V_{S(O)}$:

$$ (V_v' \cap V_{S(O)}) \subseteq V_{S(O)} \quad (13) $$

The scenario coverage ratio is therefore:

$$ \frac{V_v' \cap V_{S(O)}}{V_{S(O)}} \quad (14) $$

B. **Unified Scenario Coverage**

Sample-based methods do not explicitly possess the volume property associated with each sample scenario, as each scenario sample is rather atomic and volume-less. In order to make unified comparison and utilization of both methods for safety verification, notions of scenario volume must be assigned to sample-based methods. For simplicity, assume all $N$ continuous dimensions are orthogonal to each other, and a-axis-aligning polygonal volume in $N$-dimensional scenario space is assigned for a scenario sample specified by the parameter tuple $(p_1^0, p_2^0, \cdots, p_N^0)$. The assigned volume will take the width of $2w_i$ in the $i$th continuous scenario space dimension around the parameter value $p_i^0$, where $w_i$ is the tolerable resolution provided by the verification requirement:

**Definition 3. (Unit scenario volume)** Given a scenario space with $N$ continuous dimensions and tolerable resolution of $w_i, (i = 1, \cdots, N)$ in each dimension, the unit scenario volume is:

$$ V_0 = \prod_{i=1,\cdots,N} (2w_i) \quad (15) $$

Thus verifying full scenario coverage with sample-based methods requires to generate enough distinct, evenly spaced samples such that the overlapped volume of all verified samples supersedes $V_{S(O)}$. The number of samples required is:

$$ n = \lceil \frac{V_{S(O)}}{V_0} \rceil \quad (16) $$

where $\lceil \cdot \rceil$ is the ceiling function.

**Remark 2.** The number of samples required $n$ can possibly be reduced if evidence can be provided to show that verifying certain scenario sample can simultaneously lead to the verification of a certain group of scenario samples.

We also define safety for this unified framework. Without loss of generality, safety can be defined as a specification in natural language:

**Definition 4. (Safety Specification)** Safety is the verifiable truthfulness of statement specified by one or more safety requirement clause(s) $\mathcal{N}$ in human natural language.

As an example, $\mathcal{N}$ could be "the automated vehicle should be collision-free at all times", or "the automated vehicle should be fault-free at all times". If translated to STL, these two safety specifications can be expressed as:

$$ (s, t) \models \square \phi_{\text{collision-free}}, \quad (s, t) \models \square \phi_{\text{fault-free}} \quad (17) $$

In sample-based methods, the safety of a simulated scenario is checked at the end of each scenario sample. On the other hand, formal methods usually start with the safety specification before any simulation is performed, then the specification is translated into machine-interpretable language statement, for example, in linear temporal logic (LTL) or signal temporal logic (STL). Then the translated specification can be utilized by checking the validity of the specification in simulated/real world tests or by turning the specification into system constraints or other control design functions during control synthesis.

Ideally, for formal methods, if all safety related specifications are truthfully translated into such machine-interpretable statement in the first place, and if the synthesized controller fully respect these statements with a model-based controller that perfectly models the (simulated) environment, then the scenario coverage is 100% with respect to the simulation layer.

**Definition 5. (Specification translation)** is the process of turning a specification $\mathcal{N}$ in definition 3 to an interpreted equivalent verifiable expression $E_{\mathcal{N}}$ expressed in verification model states $(s, t)$.

As an example, the expression "the automated vehicle should be collision-free at all times" can be translated into a verifiable expression in the form of a mathematical function that evaluates occupancy overlap between the automated vehicle and any other objects in the simulation model and return true if overlap happens in the simulation. When a control policy is specified, we have the following variant:

**Definition 6. (Policy controlled specification translation)** is a specification translation under the influence of a control policy $\pi$ to the system.

However, practical challenges prevent ideal 100% translation of formal safety specifications due to the following reasons: first, formal methods rely on arguments based on abstraction of the real world, therefore discrepancies – whether large or small – between abstraction and reality, will occur and impair the effective safety guarantee. Second, the actual developed controller usually have performance limitations that prevents compliance with the safety specification in every case. Third, practical scalability difficulties exist in translating formal safety specifications into safety verification or control synthesis. In other words, formal methods usually have a penetration rate less than 100% in reality. Formally we define the penetration rate as: the percentage of verifiable scenarios in the scenario subspace specified by a formal safety specification:

**Definition 7. (Specification penetration rate)** The ratio between specification holding volume $V_h$ and specification claimed scenario space volume $V_{S(N, \pi)}$ during a policy $\pi$-controlled specification translation is the specification penetration rate $\mathcal{PR}(V_{S(N, \pi)}, \pi)$:

$$ \mathcal{PR}(V_{S(N, \pi)}, \pi) = \frac{V_h}{V_{S(N, \pi)}} \quad (18) $$
Fig. 6. Control policy evolution for safety verification. In (a)-(e), formal methods start from safety specification for the ODD (a), then an initial safety specification penetration test is performed to see how good the safety specification is kept (b). Then the feasibility check is performed to see how much of the failed scenarios are actually safety feasible (c). Then the A V controller is redesigned to remedy the failed scenario volume region(d), and eventually all safety feasible scenario volume is verified safe with the candidate A V controller. In (f)-(j), the sample-based methods start by segmenting the ODD into verifiable scenario units (f), then an incomplete sampling safety test is performed to check for major issues with the candidate A V controller (g). Subsequently fully saturated sampling is performed to ensure scenario coverage (h). After full scenario coverage test, controller weakness is exposed and the redesign process iterates (i). Eventually the controller weakness stops to shrink, and the control redesign process is likely finished(j). Notice that with formal methods safety infeasible scenario regions can be identified, and no further redesign is needed once all volume in the ODD except for the safety infeasible scenario volume is verified.

Remark 3. Definition 7 allows us to quantify the performance of a certain control policy $\pi$ in faithfully carrying out safety specification $\mathcal{N}$. Imperfections of any control policy are expected, but it is the responsibility of safety verification to quantify and curb such imperfections.

Practically, finding out about the specification penetration rate of a candidate controller for AVs in a specified ODD can be a challenge. First, deductive formal methods such as theorem proving have been limited to relatively simple discrete dynamic systems, and extending theorem proving to systems with continuous dynamics requires further established framework (a promising example [65]) and considerably higher computation resource. Second, traditional algorithmic formal methods such as model checking [66] also do not apply to systems with continuous dynamics, and running exhaustive tests of all possible actions would also create computation resource problem when fine discretization of continuous action space is needed. Recent progress in combining STL with reachability analysis [20] however, can be a possible solution to evaluate such penetration rate by calculating verification arithmetic on the candidate controller over the scenario space of the ODD, and predict the volume portion where the candidate controller will fail the STL specification. The specification penetration rate calculation problem is still an open one, and future work may combine different ways of evidence gathering, both deductive and algorithmic, to offer different alternatives.

Definition 8. (Scenario safe coverage) The scenario safe coverage $C_\pi(\mathcal{O})$ of a policy $\pi$ to a scenario space in specified ODD $\mathcal{O}$ is defined as the ratio between safety verified volume and the full scenario space volume with the candidate policy $\pi$, either using sample-based verification or formal verification. For simplicity, in the case of only one safety specification $\mathcal{N}$, the corresponding scenario coverage expressed in sample-based and formal methods are, respectively:

$$C_\pi(\mathcal{O}) = \frac{V_{v} \cap V_{S}(\mathcal{O})}{V_{S}(\mathcal{O})}, \quad C_{\pi}(\mathcal{O}) = \frac{V_{h}}{V_{S(S_{\mathcal{N}}, \pi)}} \quad (19)$$

Remark 4. In terms of scenario safe coverage, both sample-based methods or formal methods can provide confidence based on mathematical rigor at the simulation layer. Definition 6 is providing a unified framework for both school of methods to have level playing ground for comparison.

The route to complete scenario coverage is illustrated in Figure 6. Both formal and sample-based methods are valid tools to assist discovering AV safety control weakness, and AV control developers can select either or both route to their convenience and preference.

In summary, quantifying scenario coverage $C_{\pi}(\mathcal{O})$ of a candidate AV control policy $\pi$ in a specified ODD $\mathcal{O}$ (similar to solving the safety quantification problem in [16]) can be performed by either sample-based method or formal methods, or even a mixture of both.

C. Formal and Sample-based Approach Towards Full Scenario Coverage

An illustrative example of the process for verifying scenario coverage in both formal and sample-based methods is provided.
A difference between the two methods, is that formal methods have the inherent capability of determining a part of the scenario space where safety is infeasible, through for example reachability analysis.

VI. STATE-OF-THE-ART FORMAL CONTROL SAFETY VERIFICATION

Safety verification can be performed either during simulation (online) with a prediction module, or after simulation (offline) to simply check results (Figure 7). When the control policy of ego and participating traffic agents are known (white-box control) or partially known (grey-box control), formal methods can leverage this information and narrow down the reachable sets to provide ‘tighter’ prediction of vehicle motion, leading to valuable applications such as online safety monitoring and verification. Conversely, if the control policy is proprietary and completely unknown (black-box control), the safety verification process will have to involve statistical learning of the black-box controller’s behavior after each simulation, which in turn will guide the selection of next scenarios to be tested to better target counter-examples.

A. Formal Safety Verification of Known Control Policy

If the control policy is fully known (usually only limited to ego vehicle), then safety verification can be performed with much certainty, effectively removing the variable $u$ in the RA. This is usually done by computing reachable sets of controlled ego vehicle and comparing them with the occupancy set of obstacles in the interested time interval[67].

B. Formal Safety Verification of Black-box Control Policy

When the control policy is completely unknown, the verification needs to fall back to more general plans that do not require control policy information. Such plans treat the controller and the plant as an overall system, and investigate how the input (such as disturbance) to the overall system could lead to undesired output (such as safety violations). A survey on black-box verification[68] listed several such approaches as simulated annealing, evolutionary algorithms, Bayesian optimization, or extended ant colony optimization.

C. Formal Safety Verification of Grey-box Control Policy

If the control policy is partially known, modified forms of reachability analysis can be performed.

In the setting of a semi-autonomous vehicle where the human driver’s behavior is estimated at best, the safety verification has to be performed with partially known control policy (in this case the human driver policy). A particular lane keeping assist controller designed in this setting treats the human driver’s steering policy as known and immutable, while treating the human driver’s acceleration policy as unknown and mutable[69]. Such treatment allows RA to predict the range of reachable states of vehicle within a proper time horizon, and the system can determine if the human driver needs assistance by comparing the current vehicle state with backward reachable safe sets.
When at least the qualitative objective of a control policy or function is known (e.g., activate intervention to save the vehicle from an avoidable collision due to driver mis-handling), then RA can be used to verify if such control policy or function has stayed faithful to its qualitative description. For example, RA can be used to tell if a collision avoidance system have missed or abused the opportunity to intervene\cite{70}.

\section*{VII. State-of-the-art Control Synthesis with Formal Safety Guarantee}

In the literature, formal safety guarantees have been developed for automated driving simulations and tests mainly by performing reachability analysis. The types of guarantees and associated reference in summarized in Table \ref{tab:guarantees}.

\subsection*{A. Formal Control Mode Arbitration}

Formal methods can provide useful information of the criticality of current driving situation. This information can be used to develop criteria for whether a different control scheme is needed to bring ego vehicle to safety from a safety-critical scenario.

One such arbitration criteria is whether the vehicle is in an inevitable collision state (ICS)\cite{89}, which some argue is a more robust measure of collision proximity than traditional threat metrics such as time-to-collision (TTC) in estimating collision frequency\cite{52}, a vital quantity in determining automotive safety integrity level (ASIL) in ISO26262\cite{47}. If the vehicle is indeed in such a state, an impact preparation should be executed (usually with extensive use of braking to slow down\cite{20}, or by following not-at-fault trajectories\cite{77}) since collision is inevitable. This concept has been used to prototype ADAS functions such as determining when to trigger automatic emergency braking (AEB)\cite{91},\cite{92} or producing collision avoidance maneuver\cite{93}. To determine such ICS, reachability analysis is commonly used\cite{94},\cite{95},\cite{96},\cite{97}. When the prediction of moving obstacles is crucial in dynamic traffic scenarios, probabilistic framework should be adopted to create a probabilistic equivalent of ICS\cite{98},\cite{99}. For certain semi-autonomous or ADAS functions, the determination of whether system is in an inevitable collision state can help determine whether ADAS intervention is missed or unnecessary\cite{70}.

Another arbitration criteria is whether target states are achievable under the current motion mode of the system. As an example, motion modes of a vehicle can be categorized using the notion of trim trajectories characterized by the constant input required to maintain the motion\cite{100},\cite{101}. A trajectory library of a certain system can be built for the controller to choose from\cite{102},\cite{80}. Under such a context, continuous feasibility and control liveness are key concerns ensuring the accomplishment of higher level tasks, such as overtaking another slow vehicle, merging into a busy fast lane, or avoiding an obstacle at high speeds\cite{103}. When selecting mode from a large library of complicated dynamic modes become computationally demanding, learning based approach such as reinforcement learning can be leveraged to efficiently learn the proper mode arbitration decision according to the environment\cite{104},\cite{105}.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig8.png}
\caption{Robust MPC can take advantage of reachability to calculate a feasible set of states at the planning horizon (i.e. terminal state constraints), thus ensuring that the optimization process will not end up solving an infeasible constrained problem.}
\end{figure}

\subsection*{B. Formal Methods for Direct Control Integration}

Reachability can also be directly integrated in the control synthesis “at all times”, providing continuous guidance/preference to the incumbent controller to avoid catastrophic events caused by infeasible control.

One of such direct control integration methods is model predictive control (MPC) where formal methods especially reachability analysis can be used to develop robust MPC algorithms (Figure 8). Reachability analysis is used to find the feasible set of states for each MPC planning horizon so that resulting MPC algorithm will not try to find optimal solution in non-feasible region\cite{106},\cite{107},\cite{108},\cite{109}, leading to destabilization or control failure (Figure 8). This is also called recursive feasibility\cite{110} or persistent feasibility\cite{111}. When the reachable set constraints are probabilistic, a stochastic equivalent of robust MPC can be developed\cite{112},\cite{113},\cite{114}. Additional benefits of applying RA to MPC besides robustness and feasibility guarantee include a potential way of establishing system input-to-state stability\cite{115},\cite{116}, separation of robustness and performance as in learning-based MPC\cite{117}, and reducing complexity in explicit MPC by removing regions that will never be reached\cite{118}.

Another way to integrate formal safety into control design is through safe reference trajectory generation (Figure 9). Reachability can be used in the trajectory planning phase to generate optimally safe and physically-feasible trajectories for lower level controller to follow\cite{79},\cite{119},\cite{111}.

\section*{VIII. Open Questions and Resources}

\subsection*{A. Open Questions}

1) \textit{Fidelity Speed Trade-off:} In performing reachability analysis, the trade-off between dynamics fidelity and computation tractability remains a major limit. In many occasions, full reachable set (reachable states with any possible control inputs) is neither necessary or economic. With a given control policy, the policy-bounded reachability can usually be computed in
faster manner as a subset of the full reachable set. Moreover, the actual software/hardware implementation could further limit the configuration space, and by considering such limits into the RA, it is shown\cite{120} that up to 91% of computation load can be reduced.

In situations where actuation is less frequently demanded such as cruising on highway, computation can be further reduced by intermittently performing reachability analysis and skipping computation, while guaranteeing safety via control invariant set properties\cite{121}.

2) Data-Driven Approach to Reachability: An increasing number of studies are using data-driven approaches to address dynamical system verification problems such as estimation of dynamical system region of attraction\cite{122} or reachable set\cite{123}, \cite{124}, \cite{125}, \cite{126}. Dynamical systems can be approximated by deep neural networks(DNN), and to some extent the close-loop DNN-controlled system can be verified for safety by transforming DNN into equivalent hybrid system, and performing reachability analysis on the obtained hybrid system\cite{81}, \cite{82}, \cite{83}. As data-trained controllers are increasingly being tested and applied in autonomous systems such as aircraft collision avoidance system such as ACAS X\cite{127}, \cite{128}, the safety verification of neural network based controllers become research topic of interest. Feedforward neural network(FNN)\cite{85} based closed-loop controllers using ReLU activation functions can be verified at their output layers using exact or overapproximated reachable sets to comply to simple sets of linear state constraints\cite{129}, \cite{130}, \cite{131}, \cite{132}, \cite{133}. While the collision modeling in aircraft systems involve higher dimensions and susceptible to aerodynamic uncertainties, the scenarios are actually simpler and more straightforward. It is yet to see how such neural network based collision avoidance mechanism perform on automated ground vehicles, while some novel feedforward neural network training and verification using RA has been proposed for robotic systems\cite{134}. And if the collision avoidance modeling needs to be upgraded from its airborne counterpart, the methods to verify safety in\cite{129}, \cite{130}, \cite{131}, \cite{132}, \cite{133} also need to be upgraded accordingly.

While vehicle behavior is largely governed by well studied vehicle dynamics, pedestrian behavior is much more unpredictable, and rely more on data-based estimation\cite{135}, \cite{136}, also need to be upgraded accordingly.

While vehicle behavior is largely governed by well studied vehicle dynamics, pedestrian behavior is much more unpredictable, and rely more on data-based estimation\cite{135}, \cite{136}, also need to be upgraded accordingly.

3) Beyond Binary Safety Verification: Instead of using reachability for binary safety verification (e.g. safe or unsafe), it can also assist decision making in game theoretic traffic setting to increase ego vehicle payoffs\cite{139}. Although collision-avoidance is a hard constraint that shall be met at all times to ensure safety, in a more general case, when different objectives of driving can be intermittently be compromised, the study of formal verification of weakly-hard systems\cite{6} can become useful\cite{140}. In fact, considering that human perception is flawed or even delayed up to seconds\cite{141}, weakly-hard systems is not such a bad compromise.

4) Building New Logic For Safety Description: Although the majority of primary studies use linear temporal logic implicitly or explicitly for arguing safety guarantees, there exist developments of new forms of logic, such as Differential Dynamic Logic\cite{142}(based on concept in\cite{143}), allows...

\begin{table}[h]
\centering
\caption{Forms of Reachability Analysis Deployment and Types of Guarantees Provided}
\begin{tabular}{|c|c|c|}
\hline
Reference & Form of Reachability Analysis & Type of Guarantee \\
\hline
\cite{71}, \cite{72} & maxFRS + minBRS & guarantees collision if no solution for “anticipated reachable set” is found \\
\cite{73} & FRS of controlled dynamics & predicts unavoidable collision within control policy \\
\cite{74} & overapproximated maxFRS, maxFRT & all possible forward motions are included \\
\cite{75} & inaccuracy around trajectory-planned vehicle & checks probability of collision within the inaccuracy model bounds \\
\cite{76} & minBRT & guarantees collision free in horizon \\
\cite{77} & maxBRS for partially-controlled & uncontrolled vehicle & conservative collision-free guarantee for non-extreme driving \\
\cite{78} & overapproximated maxFRI & guarantee “not-at-fault” safety by checking potential collision in time intervals \\
\cite{79} & maxBRT & leading to end of time horizon \\
\cite{80} & maxFRS of uncontrolled dynamics & guarantees safe interoperability: possibly safe human takeover from autonomous driving system \\
\cite{81}, \cite{82}, \cite{83} & funnel (a variant of FRT for controlled vehicle) & guarantees collision-free if a funnel can be found to stay clear of obstacles at all times \\
\cite{84} & maxFRS implemented by Flow* & guarantees safety of deep neural network (DNN) controlled close-loop system \\
\cite{85} & maxFRS of (possibly) occluded vehicle & guarantees collision-free with possibly occluded vehicle(s) \\
\cite{86} & maxFRS of each agent communicated through a decentralized network & real-time collision-free guarantee for a group of autonomous agents \\
\cite{87} & maxFRT + collision state pruning + maxBRS & guarantees collision-free and successful reach of target state if “controller contract” (state constraint tubes) are honored \\
\cite{88} & maxFRT + pedestrian intent prediction & guarantees safety within a high probability bound of pedestrian motion \\
\hline
\end{tabular}
\end{table}

\footnote{Weakly-hard system: systems that allow certain degree of constraint or deadline misses. The degree of allowed misses can be characterized by the ratio of missed versus all constraints/deadlines.}

\footnote{Differential dynamic logic(dL): a logic for specifying and verifying hybrid systems, verifiable with dL verification calculus, explained at \url{https://symbolaris.com/logic/dL.html}.}
Fig. 9. Reachability-based safe trajectory design process: first possible trajectories are parameterized (arrows in figure); then the trajectory parameters are combined with initial states to produce FRS (red shaded area) and evaluated with obstacle occupancy (blue shaded area for obstacle geometric center, blue striped area for occupancy at a certain time); the safe candidates are filtered and an optimal set of trajectory parameter is selected by a cost function; finally a lower level tracking controller tracks the optimal trajectory.

statements and related proofs such as “an earlier engagement of the emergency brake yields a smaller collision speed”. It would be therefore interesting to see developments of new logic constructs for proving more complicated safety-related statements, as they could potentially help resolve liability issues generated by AVs.

5) Ethical Verification with Reachability: Ethical verification\cite{144} is an emerging field in automated vehicle decision making and planning. Some edge cases exist where zero-casualty is infeasible, and decision still have to be made based on ethical calculations\cite{145}. Deontic logic\cite{146}, which argues about the logic of duty and responsibility, is experimented to fit into traffic rule and accountability calculations\cite{147, 148, 144}.

B. Resources

1) Numerical Methods for Reachability Analysis: Reachable sets originate from the evolution of set of states in system dynamics. Therefore there are two major system model based methods to approximate reachable sets numerically, i.e., set shape approximation\cite{149, 150} and system dynamics approximation\cite{74, 80, 81}. Apart from model based methods, data sampling is another alternative to approximate reachable sets\cite{151} and their control invariant subsets\cite{123, 124}. Set contour approximation\cite{71, 72, 152, 69, 129, 130, 131, 133, 132, 134, 150, 149}, dynamics approximation\cite{154, 79, 155, 156, 84} and sample-based approximation\cite{125, 123, 124, 157, 158} are all viable numerical options. A summary of methods used in primary studies are listed in Table. IV. Academic competitions for reachability analysis exist, such as ARCH COMP\cite{159}, which is a competition of scientific software in the context of algorithmic verification of continuous and hybrid systems.

2) Software Tools for Reachability Analysis: For convenience, a set of software tools have been identified to perform reachability analysis. A list is provided in Table. V. Apart from various calculation approaches, efforts in constructing benchmark for reachability tests\cite{163} also help comparing effectiveness of different tools.

IX. Conclusion

In this paper we presented a holistic overview of the status quo in automated vehicle safety verification, validation and certification. In particular, the status and challenges of highly automated vehicle certification are pointed out, which should draw both academic and industrial attention as those challenges have a high reward as they are associated with major pain points in automated vehicle deployment. We identified a key missing link in quantifiable safety verification, which is a unified scenario coverage framework, and proposed such a definition to be expanded on by researchers of interest.

Formal safety verification is systematically over-viewed in this paper, including a state-of-art methodology review of its use in motion control safety verification and control synthesis with safety guarantee. Although reachability analysis, as a form of formal methods, have been used in multiple system control applications to guarantee robustness or constraint satisfaction,
the lack of a unified and convenient engineering routine for safety specification penetration prevents the wider application of reachability analysis to more safety-critical functions in industry. With emerging papers starting to bridge the formal temporal logic expressions with dynamical system control synthesis, we reachability analysis to more safety-critical functions in industry.

**REFERENCES**

[1] A. Vahidi and A. Eskandarian, “Research advances in intelligent collision avoidance and adaptive cruise control,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 3, pp. 143–153, 2003.

[2] J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels, and J. M. Zöllner, “Testing of advanced driver assistance systems towards automated driving: A survey and taxonomy on existing approaches and open questions,” in 2015 IEEE 18th International Conference on Intelligent Transportation Systems, pp. 1455–1462, 2015.

[3] S. Immen, “Kba erteil erster genehmigung zum automatisierten fahren,” Dec 2021. Accessed: 2022-01-10. [Online]. Available: https://www.kba.de/DE/Presse/Pressemeldungen/Allgemein/2021/pm49_2021_erste_Genehmigung_automatisiertes_Fahren.html.

[4] Yahoo, “Honda wins world-first approval for level 3 autonomous car,” Nov 2020. Accessed: 2022-01-10. [Online]. Available: https://news.yahoo.com/honda-wins-world-first-approval-115115427.html.

[5] T. Litman, *Autonomous vehicle implementation predictions*. Victoria Transport Policy Institute Victoria, Canada, 2017.

[6] S. S. Ho, “Complementary and competitive framing of driverless cars: framing effects, attitude volatility, or attitude resistance?,” *International Journal of Public Opinion Research*, 2021.

[7] J. Wang, L. Zhang, Y. Huang, and J. Zhao, “Safety of autonomous vehicles,” in *Autonomous vehicle implementation predictions*, pp. 176–188, Springer, 2019.

[8] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, “A survey of autonomous driving: Common practices and emerging technologies,” *IEEE access*, vol. 8, pp. 58443–58469, 2020.

[9] P. Koopman and M. Wagner, “Autonomous vehicle safety: An interdisciplinary challenge,” *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 90–96, 2017.

[10] T. F. Kone, E. Bonjour, E. Levrat, F. Meyer, and S. Géronimi, “Safety demonstration of autonomous vehicles: A review and future research questions,” in *International Conference on Complex Systems Design & Management*, pp. 176–188, Springer, 2019.

[11] J. Wang, L. Zhang, Y. Huang, and J. Zhao, “Safety of autonomous vehicles,” *Journal of advanced transportation*, vol. 2020, 2020.

[12] Z. Tahir and R. Alexander, “Coverage based testing for v&v and safety assurance of self-driving autonomous vehicles: A systematic literature review,” in 2020 IEEE International Conference On Artificial Intelligence Testing (AITest), pp. 23–30, IEEE, 2020.

[13] F. Batsch, S. Kanarachos, M. Cheah, R. Ponticelli, and M. Blundell, “A taxonomy of validation strategies to ensure the safe operation of highly automated vehicles,” *Journal of Intelligent Transportation Systems*, pp. 1–20, 2020.
TABLE V
SOFTWARE TOOLS FOR REACHABILITY ANALYSIS

| Name            | Website                                               | Feature                                                                                     |
|-----------------|-------------------------------------------------------|--------------------------------------------------------------------------------------------|
| Level-set Toolbox[164] | https://www.cs.ubc.ca/~mitchell/ToolboxLS/            | Solves Hamilton-Jacobi-Issacs equation for reachability analysis using level-set approximation |
| helperOC        | https://github.com/HJReachability/helperOC            | An optimal control toolbox for Hamilton-Jacobi reachability analysis, a tutorial is available in the appendix of [165] |
| BEACLS          | https://github.com/HJReachability/beacls             | A C++ implementation of the level-set methods for reachability analysis                      |
| FaSTrack[166]   | https://github.com/HJReachability/fastrack           | Fast planning methods with slower, reachability-based safety guarantees for online trajectory planning in ROS framework |
| KeYmaerax X[167] | https://keymaerax.org/                                | A theorem prover for differential dynamic logic                                             |
| MPT[168]        | http://people.ee.ethz.ch/~mpt/2/                      | Provides reachable set and invariance set calculation for linear, nonlinear, and hybrid systems |
| dReach[169]     | https://github.com/dreal/probreach                   | A bounded reachability analysis tool for hybrid systems                                     |
| SpaceEx[170]    | http://spaceex.imag.fr/                               | A platform for the implementation of algorithms related to reachability and safety verification |
| CommonRoad[171] | https://commonroad.in.tum.de/                         | A collection of composable benchmarks for motion planning on roads, which provides researchers with a means of evaluating and comparing their motion planners |
| SPOT[172]       | http://koschi.gitlab.io/spot/                         | A tool to predict the future occupancy of other traffic participants using reachable sets   |
| CORA            | https://github.com/TUMcps/CORA                        | A collection of MATLAB classes for the formal verification of cyber-physical systems using reachability analysis |
| Flow*[84]       | https://flowstar.org/                                 | A efficient tool to calculate reachability for polynomial-based hybrid systems               |
| ARIANDE         | https://www.ariadne-cps.org/                          | C++/Python library for formal verification of cyber-physical systems, using reachability analysis and rigorous numerics on nonlinear hybrid automata |
| HyCreate[173]   | http://stanleybak.com/projects/hycreate/hycreate.html | A tool for overapproximating reachability of hybrid automata using union of boxes to overapproximate reachable sets |
| Hylaa[174]      | https://stanleybak.com/hylaa/                          | A verification tool for system models with linear ODEs, time-varying inputs, and possibly hybrid dynamics |
| NNV             | https://github.com/verivital/nv/                      | Implements reachability methods for analyzing neural networks, particularly with a focus on closed-loop controllers in autonomous cyber-physical systems (CPS) |
| JuliaReach[175] | https://juliareach.github.io/                         | Reachability computations for dynamical systems in Julia                                      |
| C2E2[176]       | https://publish.illinois.edu/c2e2-tool/                | C2E2 can automatically check bounded time invariant properties of nonlinear hybrid automata |
| DryVR[162]      | https://dryvrtool.readthedocs.io/en/latest/           | Framework for probabilistic algorithm for learning sensitivity from simulation data, and bounded reachability analysis that uses the learned sensitivity |

[14] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, “Survey on scenario-based safety assessment of automated vehicles,” *IEEE access*, vol. 8, pp. 87456–87477, 2020.

[15] H. Kress-Gazit, K. Eder, G. Hoffman, H. Admoni, B. Argall, R. Ehlers, C. Heckman, N. Jansen, R. Knapper, J. Kretinsky, et al., “Formalizing and guaranteeing human-robot interaction,” *Communications of the ACM*, vol. 64, no. 9, pp. 78–84, 2021.

[16] B. Weng, L. J. Capito Ruiz, U. Orguner, and K. Redmill, “Towards guaranteed safety assurance of automated driving systems with scenario sampling: An invariant set perspective,” *IEEE Transactions on Intelligent Vehicles*, pp. 1–1, 2021.

[17] J. Sun, H. Zhang, H. Zhou, R. Yu, and Y. Tian, “Scenario-based test automation for highly automated vehicles: A review and paving the way for systematic safety assurance,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–16, 2021.

[18] BMVI, “Germany will be the world leader in autonomous driving,” May 2021. Accessed: 2022-01-10. [Online]. Available: https://www.bmvi.de/SharedDocs/EN/Articles/DG/act-on-autonomous-driving.html.

[19] U. N. E. C. for Europe, “Uniform provisions concerning the approval of vehicles with regard to automated lane keeping systems,” *Addendum 156 to the 1958 Agreement – UN Regulation No. 157.*, pp. 34–36, 2021.

[20] M. Chen, Q. Tam, S. C. Livingston, and M. Pavone, “Signal temporal logic meets hamilton-jacobi reachability: connections and applications,” in *Proceedings of International Workshop on the Algorithmic Foundations of Robotics*, 2018.

[21] K. Leung, E. Schmerling, M. Zhang, M. Chen, J. Talbot, J. C. Gerdes, and M. Pavone, “On infusing reachability-based safety assurance within planning frameworks for human–robot vehicle interactions,” *The International Journal of Robotics Research*, vol. 39, no. 10-11, pp. 1326–1345, 2020.

[22] SAE, “SAE J3016 Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles,” standard, Society of Automotive Engineers, USA, Apr. 2021.

[23] J. Cappereilla, “Dmv approves nuro to use autonomous vehicles for commercial service,” Dec 2020. Accessed: 2021-12-20. [Online]. Available: https://www.dmv.ca.gov/portal/news-and-media/dmv-approves-nuro-to-use-autonomous-vehicles-for-commercial-service/.

[24] DMV, “Dmv approves cruise and waymo to use autonomous vehicles for commercial service in designated parts of bay area,” Sep 2021. Accessed: 2022-01-12. [Online]. Available: https://www.dmv.ca.gov/portal/news-and-media/117199-2/.

[25] Auto2x, “Level 3-autonomous driving regulation finally coming after 3 years of delay,” Dec 2020. Accessed: 2022-01-12. [Online]. Available: https://auto2xtech.com/regulation-level-3-autonomous-driving-2021/.

[26] S. Schmid, “Bundestag nimmt gesetz zum autonomen fahren an,” May 2021. Accessed: 2022-01-10. [Online]. Available: https://www.bundestag.de/dokumente/textarchiv/2021/
H.-y. Lee, Ji-yong Lee, “Level 3 autonomous cars become legal in Germany,” Dec 2021. Accessed: 2022-01-12. [Online]. Available: https://www.caranddriver.com/news/a38475565/mercedes-drive-pilot-autonomous-germany/.

T. Stone, “Self-driving vehicles to become legal in uk in 2021,” Apr 2021. Accessed: 2022-01-15. [Online]. Available: https://www.traffictecnologytoday.com/news/autonomous-vehicles/self-driving-vehicles-to-become-legal-in-uk-in-2021.html.

D. Moss, “Self-driving cars could be legal by spring 2022,” Dec 2021. Accessed: 2022-01-15. [Online]. Available: https://www.whatcar.com/news/self-driving-cars-could-be-legal-by-spring-2022/n22965.

K. Mallesons, “China’s legislation on autonomous cars rolls out,” Apr 2021. Accessed: 2022-01-12. [Online]. Available: https://www.chinalawinsight.com/2021/04/articles/corporate-ma/chinas-legislation-on-autonomous-cars-rolls-out/.

C. Feng, “China rolls out autonomous driving standards as carmakers work towards making self-driving a reality,” Sep 2021. Accessed: 2022-01-12. [Online]. Available: https://www.scmp.com/tech/tech-war/article/3194903/china-rolls-out-autonomous-driving-standards-carmakers-work-towards.

B. Goh, “Baidu, pony.ai approved for robotaxi services in Beijing,” Nov 2021. Accessed: 2022-01-12. [Online]. Available: https://www.reuters.com/technology/baidu-ponyai-approved-robotaxi-services-beijing-2021-11-25/.

D. Roberts, “Germany will be the world leader in autonomous driving,” Jun 2019. Accessed: 2022-01-10. [Online]. Available: https://hsfnotes.com/car/2019/06/06/japan-allows-level-3-automated-vehicles-on-public-roads/.

N. staff writers, “Japan to create legal framework for level 4 self-driving cars,” Dec 2021. Accessed: 2022-01-10. [Online]. Available: https://asia.nikkei.com/Business/Automobiles/Japan-to-create-legal-framework-for-level-4-self-driving-cars.

METI, “Demonstration test to be launched for evaluation of unmanned last-mile autonomous driving in wajima city to achieve mobility revolution,” Dec 2017. Accessed: 2022-01-12. [Online]. Available: https://www.meti.go.jp/english/press/2017/1213_005.html.

H.-y. Lee, Ji-yong Lee, “Level 3 autonomous cars become legal in korea from july,” Jan 2020. Accessed: 2022-01-10. [Online]. Available: https://pulsenews.co.kr/view.php?year=2020&no=15420.

J. Jung, “Korean government announces safety standards for level 3 automated vehicles,” Jan 2020. Accessed: 2022-01-10. [Online]. Available: http://www.molit.go.kr/english/USRBORD2001/m_28286/DTL.jsp?dsid=mnlm_new&mode=view&idx=2905.

IHS, “Advanced driver assistance: Autonomous vehicle laws,” Jan 2022. Accessed: 2022-01-12. [Online]. Available: https://www.ihs.com/topics/advanced-driver-assistance/autonomous-vehicle-laws/#122.

P. Jin, Hyunjoo Dave, “Waymo, cruise seek permits to charge for autonomous car rides in san francisco,” May 2021. Accessed: 2022-01-12. [Online]. Available: https://www.reuters.com/business/autotransportation/exclusive-waymo-cruise-seek-permits-charge-self-driving-car-rides-san-francisco-2021-05-11/.

M. Subin, Samantha Wayland, “Alphabet’s waymo and gm’s cruise get california dmv approval to run commercial autonomous car services,” Sep 2021. Accessed: 2022-01-12. [Online]. Available: https://www.cnbc.com/2021/09/30/waymo-and-cruise-get-california-dmv-approval-to-run-driverless-cars.html.

M. Williams, “Prometheus the european research programme for optimising the road transport system in europe,” in IEE Colloquium on Driver Information, pp. 1–1, IET, 1988.

U. Ozguner, C. Stiller, and K. Redmill, “Systems for safety and autonomous behavior in cars: The darpa grand challenge experience,” Proceedings of the IEEE, vol. 95, no. 2, pp. 397–412, 2007.

T. Cohen, J. Stilgøe, S. Stares, N. Akyelken, C. Cavoli, J. Day, J. Dickinson, V. Fors, D. Hopkins, G. Lyons, et al., “A constructive role for social science in the development of automated vehicles,” Transportation Research Interdisciplinary Perspectives, vol. 6, p. 100133, 2020.

N. H. T. S. Administration et al., “Critical reasons for crashes investigated in the national motor vehicle crash causation survey,” Washington, DC: US Department of Transportation, vol. 2, pp. 1–2, 2015.

N. Kalra and S. M. Paddock, “Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?,” Transportation Research Part A: Policy and Practice, vol. 94, pp. 182–193, 2016.

ISO, “Road vehicles – Functional safety — Part 1: Vocabulary,” standard, International Organization for Standardization, Geneva, CH, Jan. 2018.

ISO, “Road vehicles – Safety of the intended functionality,” standard, International Organization for Standardization, Geneva, CH, Jan. 2019.

C. DMV, “Disengagement reports,” Accessed: 2021-12-06. [Online]. Available: https://www.dmv.ca.gov/portal/vehicle-industry-services/autonomous-vehicles/disengagement-reports/.

M. Grindal, J. Offutt, and S. F. Andler, “Combination testing strategies: a survey,” Software Testing, Verification and Reliability, vol. 15, no. 3, pp. 167–199, 2005.

C. Amersbach and H. Winner, “Defining required and feasible test coverage for scenario-based validation of highly automated vehicles,” in 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 425–430, 2019.

D. Åsjöng, J. Nilsson, and J. Fredriksson, “Comparing collision threat measures for verification of autonomous vehicles using extreme value theory,” IFAC-PapersOnLine, vol. 49, no. 15, pp. 57–62, 2016.

P. Junietz, F. Bonakdar, B. Klamm, and H. Winner, “Criticality metric for the safety validation of automated driving using model predictive trajectory optimization,” in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 60–65, 2018.

M. S. Blumenthal, L. Fraade-Blanar, R. Best, and J. L. Irwin, Safe Enough: Approaches to Assessing Acceptable Safety for Automated Vehicles. RAND Corporation, 2020. RR-A569-1.

S. Shahy-Schwartz, S. Shammah, and A. Shashua, “On a formal model of safe and scalable self-driving cars,” arXiv preprint arXiv:1708.06374, 2017.

P. J. Antsaklis and A. N. Michel, Linear systems, vol. 8, Springer, 1997.

D. P. Bertsekas and I. B. Rhodes, “On the minimax reachability of target sets and target tubes,” Automatica, vol. 7, no. 2, pp. 233–247, 1971.

J. M. Wing, “A specifier’s introduction to formal methods,” Computer, vol. 23, no. 9, pp. 8–22, 1990.

J. Kapinski, J. V. Deshmukh, X. Jin, H. Ito, and K. Butts, “Simulation-based approaches for verification of embedded control systems: An overview of traditional and advanced modeling, testing, and verification techniques,” IEEE Control Systems Magazine, vol. 36, no. 6, pp. 45–64, 2016.

N. Arechiga, “Specifying safety of autonomous vehicles in signal temporal logic,” in 2019 IEEE Intelligent Vehicles Symposium (IV), pp. 58–63, IEEE, 2019.

C. Barrett and C. Tinelli, “Satisfiability modulo theories,” in Handbook of model checking, pp. 305–343, Springer, 2018.

L. De Moura and N. Bjørner, “Satisfiability modulo theories: introduction and applications,” Communications of the ACM, vol. 54, no. 9, pp. 69–77, 2011.

N. Bjørner, K. L. McMillan, and A. Rybalchenko, “Program verification as satisfiability modulo theories...” SMT@ ICAR, vol. 20, pp. 3–11, 2012.

C. Barrett, L. De Moura, and A. Stump, “Smt-comp: Satisfiability modulo theories competition,” in International Conference on Computer Aided Verification, pp. 20–23, Springer, 2005.

F. Immler, “Verified reachability analysis of continuous systems,” in International Conference on Tools and Algorithms for the Construction and Analysis of Systems, pp. 37–51, Springer, 2015.

E. M. Clarke, O. Grumberg, and D. E. Long, “Model checking and abstraction,” ACM transactions on Programming Languages and Systems (TOPLAS), vol. 16, no. 5, pp. 1512–1542, 1994.

N. Kochdumper, P. Gassert, and M. Althoff, “Verification of collision avoidance for commonroad traffic scenarios,” EPiC Series in Computing, vol. 46, pp. 184–194, 2021.

A. Corso, R. J. Moss, M. Koren, R. Lee, and M. J. Kochenderfer, “A survey of algorithms for black-box safety validation,” arXiv preprint arXiv:2005.02979, 2020.
M. Althoff, O. Stursberg, and M. Buss, “Model-based probabilistic
reachability analysis of linear systems with uncertain parameters and inputs,” in 2007 46th IEEE Conference on Decision and Control, pp. 726–732, 2007.

M. Althoff, O. Stursberg, and M. Buss, “Model-based probabilistic
collision detection in autonomous driving,” IEEE Transactions on
Intelligent Transportation Systems, vol. 10, no. 2, pp. 299–310, 2009.

N. Isoda, K. Kogiso, and T. Asai, “Switching strategies of collision
avoidance and tracking control for vehicles based on non-cooperative
game and model predictive control,” in 2007 22nd International
Symposium on Intelligent Control, pp. 178–183, 2007.

J. Nilsson, P. Falcone, and J. Vinter, “Safe transitions from automated
to manual driving using driver controllability estimation,” IEEE Trans-
actions on Intelligent Transportation Systems, vol. 16, no. 4, pp. 1806–
1816, 2014.

S. Kousik, S. Vaskov, M. Johnson-Roberson, and R. Va-
sudevan, “Not-at-fault driving in traffic: A reachability-based approach,” in 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 2785–2790, IEEE, 2019.

J. Rosenson, P. Falcone, and J. Vinter, “Safe transitions from automated
to manual driving using driver controllability estimation,” IEEE Trans-
actions on Intelligent Transportation Systems, vol. 16, no. 4, pp. 1806–
1816, 2014.

A. Bemporad and M. Morari, “Robust model predictive control: A
review of results for linear-like systems,” Automatica, vol. 35, no. 3,
pp. 465–481, 1999.

A. Bemporad and M. Morari, “Robust control of systems described
by linear models with time-invariant uncertainty,” in American Control Conference, vol. 2, pp. 1355–1359, 1992.

A. Bemporad, M. Morari, and E. Zio, “Robust control of systems
described by linear models with time-varying uncertainty,” Automatica, vol. 28, no. 2, pp. 343–359, 1992.

A. Bemporad and M. Morari, “Robust model predictive control of
linear systems with uncertain time-varying parameters,” International Journal of Control, vol. 49, no. 6, pp. 1641–1651, 1989.

A. Bemporad, M. Morari, and E. Zio, “Robust control of systems
described by linear models with time-varying uncertainty,” Automatica, vol. 28, no. 2, pp. 343–359, 1992.

A. Bemporad and M. Morari, “Robust control of systems described
by linear models with time-varying uncertainty,” Automatica, vol. 28, no. 2, pp. 343–359, 1992.

A. Bemporad and M. Morari, “Robust control of systems described
by linear models with time-varying uncertainty,” Automatica, vol. 28, no. 2, pp. 343–359, 1992.

A. Bemporad and M. Morari, “Robust control of systems described
by linear models with time-varying uncertainty,” Automatica, vol. 28, no. 2, pp. 343–359, 1992.

A. Bemporad and M. Morari, “Robust control of systems described
by linear models with time-varying uncertainty,” Automatica, vol. 28, no. 2, pp. 343–359, 1992.
[156] H. A. Hansen, G. Schneider, and M. Steffen, “Reachability analysis of non-linear planar autonomous systems,” in *International Conference on Fundamentals of Software Engineering*, pp. 206–220, Springer, 2011.

[157] S. Bak, S. Bogomolov, P. S. Duggirala, A. R. Gerluch, and K. Potomkin, “Reachability of black-box nonlinear systems after koopman operator linearization,” *arXiv preprint arXiv:2105.00086*, 2021.

[158] L. Liebenwein, C. Baykal, I. Gilitschenski, S. Karunan, and D. Rus, “Sampling-based approximation algorithms for reachability analysis with provable guarantees,” *Proceedings of Robotics: Science and Systems 2018*, 2018.

[159] T. T. Johnson, “Arch-comp20 repeatability evaluation report,” in *ARCH20. 7th International Workshop on Applied Verification of Continuous and Hybrid Systems (ARCH20)* (G. Frehse and M. Althoff, eds.), vol. 74 of *EPIC Series in Computing*, pp. 175–183, EasyChair, 2020.

[160] M. Althoff, O. Stursberg, and M. Buss, “Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization,” in *2008 47th IEEE Conference on Decision and Control*, pp. 4042–4048, 2008.

[161] P. Flus and O. Stursberg, “Online control of affine systems in stochastically modeled contexts,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 3254–3260, 2020.

[162] C. Fan, B. Qi, S. Mitra, and M. Viswanathan, “Dyvr: data-driven verification and compositional reasoning for automotive systems,” in *International Conference on Computer Aided Verification*, pp. 441–461, Springer, 2017.

[163] X. Chen, S. Schupp, I. B. Makhlouf, E. Ábrahám, G. Frehse, and S. Kowalewski, “A benchmark suite for hybrid systems reachability analysis,” in *NASA Formal Methods Symposium*, pp. 408–414, Springer, 2015.

[164] I. M. Mitchell and J. A. Templeton, “A toolbox of hamilton-jacobi solvers for analysis of nondeterministic continuous and hybrid systems,” in *International workshop on hybrid systems: computation and control*, pp. 480–494, Springer, 2005.

[165] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, “Hamilton-jacobi reachability: A brief overview and recent advances,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 2242–2253, IEEE, 2017.

[166] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, “Fastrack: A modular framework for fast and guaranteed safe motion planning,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 1517–1522, 2017.

[167] A. Platzer, “A complete uniform substitution calculus for differential dynamic logic,” *J. Autom. Reas.*, vol. 59, no. 2, pp. 219–265, 2017.

[168] M. Kvasnica, P. Griener, and M. Baotic, “Multi-Parametric Toolbox (MPT),” 2004.

[169] S. Kong, S. Gao, W. Chen, and E. Clarke, “dreach: δ-reachability analysis for hybrid systems,” in *International Conference on TOOLS and Algorithms for the Construction and Analysis of Systems*, pp. 200–205, Springer, 2015.

[170] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler, “Spaceex: Scalable verification of hybrid systems,” in *International Conference on Computer Aided Verification*, pp. 379–395, Springer, 2011.

[171] M. Althoff, M. Koschi, and S. Manzinger, “Commonroad: Composable benchmarks for motion planning on roads,” in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 719–726, IEEE, 2017.

[172] M. Koschi and M. Althoff, “Spot: A tool for set-based prediction of traffic participants,” in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1686–1693, IEEE, 2017.

[173] S. Bak and M. Caccamo, “Computing reachability for nonlinear systems with hycreate,” 2013.

[174] S. Bak and P. S. Duggirala, “Hylaa: A tool for computing simulation-equivalent reachability for linear systems,” in *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, pp. 173–178, 2017.

[175] S. Bogomolov, M. Forets, G. Frehse, K. Potomkin, and C. Schilling, “JuliaReach: a toolbox for set-based reachability,” in *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, pp. 39–49, 2019.

[176] C. Fan, J. Kapinski, X. Jin, and S. Mitra, “Locally optimal reach set over-approximation for nonlinear systems,” in *2016 International Conference on Embedded Software (EMSOFT)*, pp. 1–10, IEEE, 2016.

[177] Tong Zhao (SM’17, M’21) received B.S. and M.S. degrees from Nanjing University, Nanjing, China, and the University of Waterloo, Waterloo, Canada in 2012, and M.S. in Electrical Engineering from Wuhan University, Wuhan, China in 2016. He is currently pursuing the Ph.D degree in Mechanical Engineering from the Ohio State University, Columbus OH. In 2019 he worked as a control algorithm intern at tuSimple, Tucson, Arizona. In 2020 and 2021 he worked as software engineer intern at the Mathworks. His current research focus is on limit handling vehicle control algorithm and automated vehicle control safety guarantee through formal methods.

Ekim Yurtsever (M’XX) received the B.S. and M.S. degrees from Istanbul Technical University, in 2012 and 2014, respectively, and the Ph.D. degree in information science from Nagoya University, Japan, in 2019. Since 2019, he has been a Postdoctoral Researcher with the Department of Electrical and Computer Engineering, Ohio State University, Columbus OH. His research interests include artificial intelligence, machine learning, and computer vision. He is currently working on machine learning and computer vision tasks in the intelligent vehicle domain.

Joel Paulson (M’XX) received the B.S. degree from University of Texas Austin, Austin TX, in 2011, M.S.CEP and the Ph.D. degree in Chemical Engineering from Massachusetts Institute of Technology, Boston MA in 2013 and 2016, respectively. From 2016 to 2019, he was a postdoc scholar at the Department of Chemical and Biomolecular Engineering, University of California, Berkeley CA. Since 2019, he has been an assistant professor in the Department of Chemical and Biomolecular Engineering at the Ohio State University. He focuses on the development of optimization, machine learning, and multi-scale simulation methods to improve the quality, efficiency, and sustainability of engineered products and processes. Two particular areas of focus in his group are: (i) construction of predictive models using a combination of physics-based equations and data and (ii) the formulation of real-time decision-support tasks under uncertainty as structured optimization problems that are solved efficiently using state-of-the-art algorithms. The developed strategies have been successfully applied to a broad range of systems including continuous pharmaceutical manufacturing, colloidal self-assembly, and non-equilibrium plasma jets.
Giorgio Rizzoni received B.S. (ECE) in 1980, M.S. (ECE) in 1982, Ph.D. (ECE) in 1986, all from the University of Michigan.

Since 1999 he has been the director of the Ohio State University Center for Automotive Research (CAR), an interdisciplinary university research center in the OSU College of Engineering. His research activities are related to modeling, control and diagnosis of advanced propulsion systems, vehicle fault diagnosis and prognosis, electrified powertrains and energy storage systems, vehicle safety and intelligence, and sustainable mobility. He has contributed to the development of graduate curricula in these areas, and has served as the director of three U.S. Department of Energy Graduate Automotive Technology Education Centers of Excellence: Hybrid Drivetrains and Control Systems (1998-2004), Advanced Propulsion Systems (2005-2011), and Energy Efficient Vehicles for Sustainable Mobility (2011-2016). Between 2011 and 2016 he served as the OSU Site Director for the U.S. Department of Energy China-USA Clean Energy Research Center - Clean Vehicles. He is currently leading an ARPA-E project in the NEXTCAR program. During his career at Ohio State, Prof. Rizzoni has directed externally sponsored research projects funded by major government agencies and by the automotive industry in approximately equal proportion.

Prof. Rizzoni is a Fellow of SAE (2005), a Fellow of IEEE (2004), a recipient of the 1991 National Science Foundation Presidential Young Investigator Award, and of many other technical and teaching awards.