Automatica 152 (2023) 110692

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Survey paper

Formal methods to comply with rules of the road in autonomous driving: State of the art and grand challenges☆

Noushin Mehdipour[a], Matthias Althoff[b], Radboud Duintjer Tebbens[a], Calin Belta[a,c,*]

[a] Motional, Boston, MA, USA
[b] Technical University of Munich, Munich, Germany
[c] Boston University, Boston, MA, USA

ARTICLE INFO

Article history:
Received 18 February 2021
Received in revised form 26 July 2022
Accepted 5 September 2022
Available online 24 March 2023

Keywords:
Autonomous driving
Formal methods
Temporal logic
Formal verification
Formal synthesis
Falsification
Monitoring
Machine learning

ABSTRACT

We provide a review of recent work on formal methods for autonomous driving. Formal methods have been traditionally used to specify and verify the behavior of computer programs and digital circuits. Enabled by abstraction techniques for dynamical systems and the availability of verification and synthesis tools for finite systems, they have been adopted by the control and robotics communities. In particular, in autonomous driving, recent research proposes formal languages such as temporal logics to specify driving behaviors ranging from safety, such as collision avoidance, to compliance with complex rules of the road. Our review focuses on formal verification, monitoring, and synthesis techniques enabling autonomous vehicles to adhere to such specifications. We only consider works about system-level methods that have an ego-centric perspective, i.e., we focus on the behavior of an autonomous vehicle in its entirety, rather than specific software code within the vehicle or traffic networks consisting of multiple vehicles. This paper also identifies the main remaining challenges.

1. Introduction

The development and integration of cyber–physical and safety-critical systems in various engineering disciplines requires their verification and control with respect to rich specifications. A prominent example is autonomous driving, which received a lot of attention during the last decade. Autonomous vehicles (AVs) aim to optimize common control objectives, such as minimizing the energy consumption and travel time, and satisfy constraints on control variables, such as maximum acceleration. In addition, AVs aim to drive safely and follow the rules of the road (ROTRs), which include traffic laws and other informal rules or cultural expectations of reasonable driving behavior. For example, an AV tries to avoid collisions with other road users, avoid obstructing traffic, maintain longitudinal clearance with the lead vehicle, yield when required, and stop at red lights and stop signs. These rules could be prioritized, e.g., by specifying that maintaining clearance to pedestrians is more important than staying in lane, which, in turn, takes precedence over observing the maximum speed limit. There currently exists no consensus on how and to what extent AVs should follow such complex (possibly prioritized) driving specifications.

Formal methods is an area of computer science, traditionally focused on checking the correctness of digital circuits and computer programs. Correctness can pertain to safety (something bad should never happen), liveness (something good should eventually happen), or general statements expressed as formulas of Temporal Logics (TL), such as Linear Temporal Logic (LTL), Computation Tree Logic (CTL) (Baier & Katoen, 2008; Clarke, Grumberg, & Peled, 1999), or Signal Temporal Logic (STL) (Maler & Nickovic, 2004). Due to the high expressivity of these specification languages, the existence of verification, monitoring, and control synthesis tools for finite systems, and recent developments on abstractions for systems with infinite state spaces, formal methods have been adopted by the control community, and successfully used for dynamical (Belta, Yordanov, & Gol, 2017; Mitra, 2021; Tabuada, 2009) and autonomous systems (Luckcuck, Farrell, Dennis, Dixon, & Fisher, 2019; Plaku & Karaman, 2016).

In particular, there is a growing body of work on the use of formal methods for autonomous driving. TLs have been proposed for the formal specification of safety requirements and complex ROTRs. Formal verification, monitoring, and synthesis techniques have been used for analysis and control of autonomous driving behavior. Machine learning algorithms have been employed to infer formal rules describing ROTRs and desired behaviors from

☆ The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Editor Sonia Martinez.
* Corresponding author at: Boston University, Boston, MA, USA.
E-mail addresses: noushin.mehdipour@gmail.com (N. Mehdipour), althoff@in.tum.de (M. Althoff), radboud.tebbens@motional.com (R.D. Tebbens), cbelta@bu.edu (C. Belta).

https://doi.org/10.1016/j.automatica.2022.110692
0005-1098/© 2022 Published by Elsevier Ltd.

Abbreviation

ACC	Adaptive Cruise Control
AV	Autonomous Vehicle
CBF	Control Barrier Function
CLF	Control Lyapunov Function
CTL	Computation Tree Logic
ICS	Inevitable Collision State
LTL	Linear Temporal Logic
MI(L)P	Mixed-integer (Linear) Programming
MPC	Model Predictive Control
MTL	Metric Temporal Logic
QP	Quadratic Program
ROTR(s)	Rule(s) of the road
RRT	Rapidly-exploring Random Tree
RSS	Responsibility-Sensitive Safety
STL	Signal Temporal Logic
TL	Temporal Logic

data, to assess the relative importance of different formal rules from data, and to generate driving strategies.

In this paper, we review the literature on formal methods used for autonomous driving. The focus is on studies that consider ROTRs from the ego vehicle's point of view. We *do not include* traffic networks, e.g., traffic light control, conflict resolution at intersections, congestion control, and merging control. We also focus on formal methods at the system level. We do not consider formal methods for the software running on the autonomous cars.

Recent review papers covering the fast growing field of autonomous driving include Dahl, de Campos, Olsson, and Fredriksson (2019), Ersal et al. (2020), Guanetti, Kim, and Borrelli (2018), Rajabli, Flammini, Nardone, and Vittorini (2021), Riedmaier, Ponn, Ludwig, Schick, and Diermeyer (2020), Schwarting, Alonso-Mora, and Rus (2018), Seshia, Sadigh, and Sastry (2015) and Yurtsever, Lambert, Carballo, and Takeda (2020). A comprehensive review of a broad range of topics, including system architectures, localization, mapping, perception, planning, and human machine interfaces is provided in Yurtsever et al. (2020). The work in Schwarting et al. (2018) also provides a general overview of the field, with particular emphasis on planning, but does not survey the state of the art in formal methods for autonomous driving. With particular relevance to our review, Schwarting et al. (2018) include a discussion on formal methods for planning. A comprehensive review of the state of the art in software verification and validation of AVs is provided in Rajabli et al. (2021). Another comprehensive review, which includes a discussion on safety verification of controllers for AVs, is provided in Ersal et al. (2020).

Among more focused reviews, Riedmaier et al. (2020) survey scenario-based approaches, in which individual traffic situations are tested through simulation. The focus is on safety assessment. A literature review and analysis of threat-assessment methods used for collision avoidance is presented in Dahl et al. (2019), including the use of formal methods. The focus of Seshia et al. (2015), which reviews a limited number of papers, is on human cyber–physical systems, with particular focus on semi-autonomous driving, and formal methods. Finally, Guanetti et al. (2018) introduce a control and planning architecture for connected vehicles and AVs and surveys the state of the art on each functional block therein.

Compared to the survey papers covering general topics in autonomous driving mentioned above, this review focuses solely on formal methods. It provides a comprehensive overview of the state of the art that is more in-depth and detailed than the reviews referenced above, which only contain sections of formal methods. Finally, it covers very recent papers in the fast growing field of autonomous driving, with specific emphasis on formalization of ROTRs.

The remainder of this paper is organized as follows. In Section 2, we review the methods used to formalize ROTRs and other driving behaviors. We discuss the formal verification approaches used to analyze vehicle models and behaviors from such formal specifications in Section 3. We review monitoring algorithms and formal synthesis strategies in Sections 4 and 5, respectively. We discuss remaining challenges in the field in Section 6 and conclude with final remarks in Section 7.

For quick reference, each section concludes with a table that lists all the papers cited in that section. Its columns represent the particular categories covered in that section. For example, Table 3 from Section 4 that covers monitoring, has two columns "Offline" and "Online", corresponding to the two monitoring techniques from the reviewed papers. Each table has three rows, which correspond to the three application areas: "Vehicle following", "Lane keeping/changing", and "Other". Papers listed under "Vehicle following" and "Lane keeping/changing" focus on the respective application only. The "Other" category corresponds to papers that discuss at least one application area different from the ones listed above (e.g., pedestrian clearance or speed limit), or that includes discussion of both "Vehicle following" and "Lane keeping/changing". We believe that these tables make it easy for the reader to find a paper using a specific technique in a particular application area. For example, if she wants to find a paper that uses online monitoring from STL specifications with applications to lane keeping, then she would use Table 3 to find the papers that apply online monitoring to lane keeping: Aréchiga (2019) and Kojchev, Klintberg, and Fredriksson (2020). She would then check which of these papers appear at the intersection of the "STL" column and the "lane keeping/changing" row in Table 1. In this example, it turns out that Aréchiga (2019) is the only paper that meets these specific criteria. Note that, given the organization of the review, a paper can appear in several sections and tables.

2. Formal specifications

Most ROTRs are stated in natural language in traffic legislations or driving manuals, which can differ among countries and localities within countries. Some ROTRs can be formalized as simple *safety specifications* that guarantee safety when satisfied. For example, a safety specification might be a formal rule that the ego vehicle should maintain a given minimum clearance from pedestrians on the road for all times. Safety specifications are sometimes given equivalently as reachability specifications. In the above example, the reachability specification is that the ego vehicle can only reach distances to pedestrians that are larger than the minimum clearance.

Safety specifications are a particular case of TL specifications, which specify formal rules that can also express eventuality (e.g., "reach destination in at most 10 min", "maintain a speed less than 25 mph until the end work zone sign is reached"), logical conditions ("use the left lane only when passing"), and combinations of the above. Formal rules are sometimes prioritized. For example, a safety specification such as "maintain clearance from pedestrians at all times" might have precedence over "reach destination in at most 10 min". In Section 2.1, we briefly discuss safety specifications. Richer, TL formalisms are covered in Section 2.2. Papers dealing with rule priorities are reviewed in Section 2.3.

It is important to note that, throughout the paper, "safety specification" refers to a formal specification of the form: "for all times, an undesired outcome never happens". In other words, as stated above, it is just a particular type of a TL formula. Safety in this context is not necessarily a stringent requirement such as collision avoidance. For example, "Stay in lane for all times" is a safety specification in formal methods (and in this survey) but not a safety requirement since changing lanes is a perfectly acceptable behavior in many driving situations (e.g., when preparing for a left turn).

2.1. Safety specifications

In most of the papers included in this review, the formal specification is given simply as a safety specification only (see Table 1). Therefore, even though safety is just a specific kind of temporal logic formula, we dedicate Section 2.1 and the first column of Table 1 to such papers. The most predominant safety specifications are collision avoidance, maintaining a minimum clearance from the preceding car, staying in lane and/or on the road. In the rest of this section, we briefly discuss works that provide safety specifications using control theoretic or motion planning concepts, or combine safety specifications with other specifications, such as lawfulness.

The safety specifications in Bouraine et al. (2012), Parthasarathi and Fraichard (2007) and Lawitzky et al. (2014) refer to avoiding collisions with static and dynamic obstacles, and are formalized using Inevitable Collision States (ICS) (i.e., states for which, no matter what the future trajectory followed by the ego vehicle is, a collision with an obstacle eventually occurs Fraichard & Asama, 2004). ICS are used to enforce safety during motion planning. Predictions of future occupancies for surrounding traffic participants are used for safety specifications in Althoff and Dolan (2014), Althoff and Magdici (2016), Koschi and Althoff (2017a, 2021), Söntges and Althoff (2015, 2018) and Wu and How (2012), and applied to the influential Responsibility-Sensitive Safety (RSS) modeling framework (Shalev-Shwartz et al., 2017) in Orzechowski et al. (2019).

Positive invariant sets (e.g., sets that are guaranteed to contain all trajectories of the vehicle for all times are used to formalize safety in some works. In Berntorp et al. (2017), these are used to ensure that the ego vehicle stays on the road. Positive invariant sets are also used in Berntorp et al. (2020) to prove safety as defined through velocity and obstacle collision constraints. Control invariant sets (i.e., sets that are made positive invariant using control) are used in Hoehener et al. (2016), Jalalmaab et al. (2017), Sadraddini et al. (2017) and Smith et al. (2016). The more recent, closely related concept of control barrier functions (CBF) is used in Ames et al. (2014), Mehra et al. (2015) and Xu et al. (2018). Compositional and contract-based principles are used for formal verification of safety specifications (DeCastro et al., 2020; Liebenwein et al., 2020).

Finally, safety specifications are combined with lawfulness and liabilities of traffic participants in Pek et al. (2017b), Rizaldi and Althoff (2015), Rizaldi et al. (2016) and Vanholme et al. (2013). The authors of Pek et al. (2017b), Rizaldi and Althoff (2015) and Rizaldi et al. (2016) focus on liabilities of traffic participants if a collision occurs using formal rules based on the Vienna Convention on Road Traffic. The concept of legal safety is defined in Vanholme et al. (2013) as a set of rules that could safely and efficiently manage mixed traffic of human drivers and AVs, and illustrated for automated driving on highways with distance keeping, speed adaptation, and lane-changing. Requirements induced by legal safety on perception and control components are also presented. Legal safety is used in Pek et al. (2020) for critical urban scenarios, which have been recorded in real traffic.

2.2. TL specifications

Most of the reviewed papers use standard TLs, such as LTL and a common fragment called syntactically co-safe LTL (scLTL) (Kupferman & Vardi, 2001), STL (Maler & Nickovic, 2004), and MTL (Koymans, 1990). Others propose new logics, specifically tailored for formalizing ROTRs (Dokhanchi et al., 2018; Jha et al., 2018; Wongpiromsarn et al., 2021). With few exceptions (e.g., Dokhanchi et al., 2018; Ody, 2017; Rizaldi & Althoff, 2015; Rizaldi et al., 2018), which use high-order temporal logic (i.e., logics that allow for universal and existential quantifiers), all the reviewed works focus on propositional and predicate temporal logics.

Informally, LTL formulas are made of three ingredients: (1) atomic propositions (e.g., *por* = "pedestrian on the road") or predicates (e.g., $v_{ego} < 30$ = "the ego vehicle's speed is less than 30 miles per hour"); (2) Boolean operators $\vee$ (disjunction), $\wedge$ (conjunction), $\neg$ (negation), etc.; and (3) temporal operators, such as **G** (globally, or always), **F** (in the future, or eventually), **X** (next), and **U** (until). For example, the LTL formula $\mathbf{G}(por \rightarrow (sd\, \mathbf{U}\, pos))$, reads "for all times, if a pedestrian is on the road, slow down until she reaches the sidewalk" (*por*, *sd*, and *pos* are propositions that are true when the pedestrian is on the road, the ego vehicle slows down, and the pedestrian is on the sidewalk, respectively). LTL formulas are interpreted over infinite executions. scLTL is a strict fragment of LTL, in which the satisfaction of formulas can be decided in finite time (Kupferman & Vardi, 2001). For example, formula $por \rightarrow (sd\, \mathbf{U}\, pos)$ is in scLTL, while $\mathbf{G}(por \rightarrow (sd\, \mathbf{U}\, pos))$ is in LTL but not in scLTL. For both LTL and scLTL, time is abstract, i.e., only the order of the events matter.

MTL is an extension of propositional LTL, in which time is concrete, and formulas can refer to both past and future times. Informally, the main difference is that the temporal operators are timed. For example, the requirement that the ego vehicle slows down and comes to a complete stop within 5 seconds translates to the MTL formula $sd\, \mathbf{U}_{[0,5]}\, stop$, where *sd* is the same as above and *stop* is a proposition that is true when the ego vehicle stops. STL is an extension of LTL with real-time and real-valued constraints, and its formulas are usually over predicates. For example, a formal specification to comply with the maximum speed limit is written in STL as $\mathbf{G}_{[0,T]}(v(t) < v_{max})$, where $v(t)$ is the ego vehicle's speed at time t, v_{max} is the maximum speed limit, and T is the total duration of the scenario during which compliance with this specification is evaluated. In addition to Boolean semantics, in which a word or signal satisfies or violates a formula, MTL and STL have quantitative semantics. This is defined using a robustness function that gives the degree of satisfaction of a formula by a word or signal. Many papers reviewed below use the robustness function for monitoring and/or controller synthesis.

LTL. ROTRs based on the German concretization of the Vienna Convention on Road Traffic are formalized using LTL in Esterle et al. (2020). The focus is on dual carriageways, such as highways, and the formulas are restricted to the particular form "premise implies conclusion", or $\mathbf{G}(\phi^p \rightarrow \phi^c)$, where ϕ^p is the premise and ϕ^p is the conclusion. An LTL formula of this form states that "at all times, if ϕ^p is True, then ϕ^c must be True". The authors provide algorithms for constructing such formulas from ROTRs with the help of graphical representations. To define semantics for the LTL formulas over vehicle trajectories, the atomic propositions are concretized to predicates, e.g., atomic proposition $acc^{(i)}$ corresponds to predicate "i accelerates with $a > a_{lim}$". A related approach, for a related set of German ROTRs, is proposed in Rizaldi et al. (2017), where the focus is on overtaking. In this work, the LTL formulas are more general and the predicates are concretized through legal and engineering analyses.

Table 1
Papers organized by type of formal specification and application area. **Safety only** refers to papers that only specify safety.

	Safety only	LTL	STL	MTL	Other TLs	Priorities
Vehicle following	Alam, Gattami, Johansson, and Tomlin (2014), Althoff, Maierhofer, and Pek (2021), Alvarez and Horowitz (1999), Ames, Grizzle, and Tabuada (2014), Dolginova and Lynch (1997), Ligthart, Semsar-Kazerooni, Ploeg, Alirezaei, and Nijmeijer (2018), Loos, Witmer, Steenkiste, and Platzer (2013), Lygeros, Godbole, and Sastry (1996, 1998), Magdici and Althoff (2017), Mehra et al. (2015), Park and Özgüner (2012), Rizaldi, Immler, and Althoff (2016), Sadraddini, Sivaranjani, Gupta, and Belta (2017), Shalev-Shwartz, Shammah, and Shashua (2017), Stursberg, Fehnker, Han, and Krogh (2004), Xu, Grizzle, Tabuada, and Ames (2018)	Nilsson et al. (2016)		Maierhofer, Rettinger, Mayer, and Althoff (2020), Rodionova et al. (2020)		Althoff et al. (2021)
Lane keeping/changing	Althoff and Dolan (2012), Berntorp, Weiss, Danielson, Kolmanovsky, and Di Cairano (2017), Hilscher, Linker, and Olderog (2013), Hoehener, Huang, and Del Vecchio (2016), Jula, Kosmatopoulos, and Ioannou (2000), Kojchev et al. (2020), Mirchevska, Pek, Werling, Althoff, and Boedecker (2018), Naumann, Königshof, and Stiller (2019b), Pek, Zahn, and Althoff (2017b), Shao, Chen, Kousik, and Vasudevan (2021), Wongpiromsarn, Mitra, Murray, and Lamperski (2012)		Aréchiga (2019), Hekmatnejad et al. (2019)			
Other	Ahn, Berntorp, Inani, Ram, and Di Cairano (2021), Althoff, Althoff, Wollherr, and Buss (2010), Althoff and Dolan (2011, 2014), Althoff and Magdici (2016), Berntorp et al. (2020), Bouraine, Fraichard, and Salhi (2012), Brüdigam, Olbrich, Wollherr, and Leibold (2021), Chou, Yoon, and Sankaranarayanan (2020), Dai and Koutsoukos (2016), Danielson, Berntorp, Weiss, and Cairano (2020), DeCastro et al. (2020), Du et al. (2020), Falcone, Ali, and Sjöberg (2011), Fan (2019), Fan, Qi, and Mitra (2018), Gerdts and Xausa (2013), Herbert et al. (2017), de Iaco, Smith, and Czarnecki (2020), Jalalmaab, Fidan, Jeon, and Falcone (2017), Karimi and Duggirala (2020), Kianfar, Falcone, and Fredriksson (2013), Koschi and Althoff (2017a, 2017b, 2021), Koschi, Pek, and Althoff (2018a), Koschi, Söntges, and Althoff (2018b), Kousik, Vaskov, Johnson-Roberson, and Vasudevan (2017), Lawitzky, Nicklas, Wollherr, and Buss (2014), Liebenwein et al. (2020), Lin, Chen, Khurana, and Dolan (2020), Linker and Hilscher (2013), Loos, Platzer, and Nistor (2011), Macek, Vasquez, Fraichard, and Siegwart (2009), Magdici and Althoff (2016), Nager, Censi, and Frazzoli (2019), Naumann, Konigshof, Lauer, and Stiller (2019a), Neel and Saripalli (2020), Orzechowski, Li, and Lauer (2019), Orzechowski, Meyer, and Lauer (2018), Parthasarathi and Fraichard (2007), Pek and Althoff (2018), Pek, Koschi, Werling, and Althoff (2017a), Pek, Manzinger, Koschi, and Althoff (2020), Rizaldi and Althoff (2015), Schmidt, Oechsle, and Branz (2006), Schürmann et al. (2017), Smith, Nilsson, and Ozay (2016), Soloperto, Köhler, Allgöwer, and Müller (2019), Söntges and Althoff (2015, 2018), Stahl and Diermeyer (2021), Vanholme, Gruyer, Lusetti, Glaser, and Mammar (2013), Vaskov et al. (2019), Völker, Kloock, Rabanus, Alrifaee, and Kowalewski (2019), Wang, Li, and Sifakis (2020), Wu and How (2012), Xiao et al. (2021)	Esterle, Aravantinos, and Knoll (2019), Esterle, Gressenbuch, and Knoll (2020), Rizaldi, Immler, Schürmann, and Althoff (2018), Rizaldi et al. (2017), Rong and Luan (2020), Vasile, Tumova, Karaman, Belta, and Rus (2017)	Aasi, Vasile, and Belta (2021), Cho, Ha, Lee, and Oh (2019), Corso and Kochenderfer (2020), Li, Rosman, Gilitschenski, DeCastro et al. (2021), Li, Rosman, Gilitschenski, Vasile et al. (2021), Sahin, Quirynen, and Cairano (2020), Tuncali, Fainekos, Ito, and Kapinski (2018), Tuncali, Fainekos, Prokhorov, Ito, and Kapinski (2020)	Kane, Chowdhury, Datta, and Koopman (2015), O'Kelly, Abbas, and Mangharam (2017)	Dokhanchi, Amor, Deshmukh, and Fainekos (2018), Jha, Raman, Sadigh, and Seshia (2018), Wongpiromsarn, Slutsky, Frazzoli, and Topcu (2021)	Censi et al. (2019), Cho et al. (2019), DeCastro et al. (2020), Helou et al. (2021), Castro et al. (2013), Sadigh, Dragan, Sastry, and Seshia (2017), Vasile et al. (2017), Wongpiromsarn et al. (2021), Xiao et al. (2021)

LTL specifications obtained by interpreting relevant Adaptive Cruise Control (ACC) standards are considered in Nilsson et al. (2016) and used to produce correct-by-construction controllers. LTL formulas over a semantic abstraction obtained by partitioning the continuous state space corresponding to a traffic scenario are considered in Esterle et al. (2019), where the authors consider a subset of the Vienna Convention of Road Traffic that cover the interaction of the ego vehicle with only one other traffic participant. In the related work Rizaldi et al. (2018), the authors perform AV motion planning from ROTRs expressed as LTL formulas, which are interpreted over maneuver automata, and allow for automatic satisfiability checking. LTL is proposed in Rong and Luan (2020) to specify a small set of ROTRs, together with a quantitative semantics used for reinforcement learning. scLTL is used in Vasile et al. (2017) to formalize a set of ROTRs that need to be satisfied, while customer demands (e.g., pick ups, drop offs) are met within desired deadlines.

STL. Our review shows that STL is the preferred logic for specifying ROTRs. One of the main advantages of STL is its quantitative semantics, which allows for monitoring, and also to map verification and control synthesis problems to optimization problems (Aasi et al., 2021; Sahin et al., 2020). Recent work also points to an interesting connection between the quantitative semantics of STL and deep learning for autonomous driving. The work in Cho et al. (2019) represents ROTRs as STL formulas and uses its quantitative semantics and a deep learning framework to predict future behavior of nearby vehicles and to recognize the importance of predefined formal rules. Using a parameterized version of STL (pSTL), Li, Rosman, Gilitschenski, DeCastro et al. (2021) proposes a method for integrating TL formulas into a neural network, which allows incorporating ROTRs into deep learning-based trajectory prediction approaches. This framework is extended in Li, Rosman, Gilitschenski, Vasile et al. (2021), where ROTRs expressed as STL formulas are integrated as inductive biases into deep learning-based prediction models.

A requirements-driven approach for test case generation is proposed in Tuncali et al. (2018, 2020), which covers both component-level and system-level behaviors for an AV. Test cases are evaluated against STL formulas and the requirements are used to automatically discover test cases that fail to satisfy the requirements. The related work Corso and Kochenderfer (2020) describes an approach for finding interpretable failures of an AV system. The failures are described as STL formulas and optimization is used to produce likely failures.

Recent works showed that STL can be efficiently used to formalize assume-guarantee conditions. In particular, in Aréchiga (2019), the author develops a set of contracts for control software for AVs ensuring that if all traffic participants follow the contracts (i.e., the assumption), then the overall traffic system is collision-free (i.e., the guarantee). In Hekmatnejad et al. (2019), it is shown that RSS assumptions can be encoded in assume-guarantee logical conditions in STL, which enables the use of verification and testing tools to verify and validate AV compliance with RSS.

MTL. MTL is used in Maierhofer et al. (2020) to formalize ROTRs for interstates based on the German Road Traffic Regulation, the Vienna Convention on Road Traffic, and legal decisions from courts. In this paper, the authors also use first-order logic to define the predicates and functions used in the formulas. Specifications for a case study using the RSS model are given as MTL formulas in Rodionova et al. (2020). The work in O'Kelly et al. (2017) proposes a scenario description language to create driving scenarios with different numbers of agents and on different road topologies, which also enables the specification of formal correctness specifications in MTL. A future-bounded, propositional MTL is used in Kane et al. (2015) to specify correctness properties for components of an AV.

Other TLs. Three out of the 20 reviewed papers that use TLs for formalizing ROTRs propose new logics specifically tailored to autonomous driving. The work in Dokhanchi et al. (2018) introduces Timed Quality Temporal Logic (TQTL), an extension of STL, to monitor and test the performance of object detection and situation awareness algorithms. An example of a vision quality requirement in this framework is "at every time step, for all the objects *id* in the frame, if the object class is cyclist with probability more than 0.7, then in the next 5 frames the object *id* should still be classified as a cyclist with probability more than 0.6". A probabilistic TL, called Chance Constrained Temporal Logic (C2TL), is proposed in Jha et al. (2018) to specify correctness requirements in the presence of probabilistic uncertainty. The main addition of C2TL over STL is the inclusion of chance constraints as predicates. A chance constraint is a probabilistic extension of deterministic predicates and is of the form $Pr(\phi_{det}) \geq 1-\delta$, where $0 \leq \delta \leq 1$ represents uncertainty about whether the inequality holds and ϕ_{det} is a Boolean combination of linear predicates, where the coefficients are random variables with Gaussian probability distributions. Finally, a version of LTL, called stutter-invariant Finite Linear Temporal Logic (si-FLTLGX), is introduced in Wongpiromsarn et al. (2021). While sufficient to describe many ROTRs, si-FLTLGX also allows for prioritized ROTRs and for efficient computation of optimal motion plans through sampling.

2.3. Rule priorities

Only a few studies deal with the formal specification of multiple potentially competing driving objectives. The work in Althoff et al. (2021) encodes a specification of an adaptive cruise controller that ensures compliance with safety specifications while maintaining comfortable control actions. In Vanholme et al. (2013), the authors discuss the need to not *a priori* exclude trajectories that violate a formal rule like staying in the driving lanes, because in an emergency such a trajectory might be necessary. However, they stop short of specifying rule priorities. In DeCastro et al. (2020), the authors ensure the satisfaction of safety specifications and eight formalized ROTRs. They encode an implicit notion of rule priorities by relaxing a subset of the rules in some experiments. The work in Castro et al. (2013) considers a set of safety specifications as formal rules and defines a notion of global minimization of rule violation based on discrete priority levels for each formal rule. While all of these studies consider multiple formal ROTRs, none of them explicitly captures priorities.

The work in Censi et al. (2019) provides a general framework to specify how an AV can transparently resolve conflicts between formal rules using a priority structure. The proposed priority structure is a pre-ordered set of formal rules, which induces a pre-order on any set of potential trajectories in a scenario. Several studies build on this framework to develop algorithms for planning (Wongpiromsarn et al., 2021) or control (Vasile et al., 2017; Xiao et al., 2021). In Xiao et al. (2021), the authors propose an offline methodology for pass/fail evaluation of AV behavior to determine whether a given AV trajectory complied with a priority structure of formal rules. They do so by defining a candidate AV trajectory as non-compliant if another trajectory exists that violates only lower priority rules than the candidate AV trajectory, which they determine through iterative relaxation of the rules.

Some studies explore the use of a learned priority structure among various formal rules. For example, Cho et al. (2019) learn the margins of satisfaction for formal rules and then apply them in Model Predictive Control (MPC) of the ego vehicle and surrounding vehicles. Another study Sadigh et al. (2017) queries pairwise preferences between trajectories to learn the weights that can be viewed as quantitative measures of how well a trajectory satisfies rules for staying on the road and avoiding collisions.

Table 2
Papers covering verification techniques organized by application areas, suitability for online use, and applicability in mixed traffic.

	Online	Mixed traffic	References
Vehicle following	✗	✗	Ames et al. (2014), Loos et al. (2011, 2013), Sadraddini et al. (2017), Stursberg et al. (2004)
	✗	✓	Alam et al. (2014), Alvarez and Horowitz (1999), Dolginova and Lynch (1997), Lygeros et al. (1996, 1998), Mehra et al. (2015), Nilsson et al. (2016)
	✓	✗	Park and Özgüner (2012)
	✓	✓	Althoff et al. (2021), Ligthart et al. (2018), Magdici and Althoff (2017), Wang et al. (2020)
Lane keeping/changing	✗	✗	Hilscher et al. (2013), Linker and Hilscher (2013)
	✓	✗	Jula et al. (2000), Shao et al. (2021)
	✓	✓	de Iaco et al. (2020), Mirchevska et al. (2018), Naumann et al. (2019b), Pek et al. (2017b)
Other	✗	✗	Lawitzky et al. (2014), Smith et al. (2016), Wongpiromsarn et al. (2012)
	✓	✗	Althoff et al. (2010)
	✗	✓	Dai and Koutsoukos (2016), Fan et al. (2018), O'Kelly et al. (2017), Völker et al. (2019), Xu et al. (2018)
	✓	✓	Ahn et al. (2021), Althoff and Dolan (2011, 2012, 2014), Althoff and Magdici (2016), Bouraine et al. (2012), DeCastro et al. (2020), Koschi and Althoff (2017a, 2017b, 2021, 2021), Koschi et al. (2018a, 2018b), Kousik et al. (2017), Liebenwein et al. (2020), Lin et al. (2020), Nager et al. (2019), Neel and Saripalli (2020), Orzechowski et al. (2018), Pek et al. (2017a, 2020), Schmidt et al. (2006), Schürmann et al. (2017), Söntges and Althoff (2015), Stahl and Diermeyer (2021), Vaskov et al. (2019)

Table 3
Papers organized by type of monitoring and application area (there are no works in the "Vehicle following" application area).

	Offline	Online
Lane keeping/changing	Hekmatnejad et al. (2019)	Kojchev et al. (2020)
Other	Esterle et al. (2019), Ody (2017), Rizaldi et al. (2017)	Aasi et al. (2021), Chou et al. (2020), Du et al. (2020), Esterle et al. (2020), Kane et al. (2015), Sahin et al. (2020)

The work in Helou et al. (2021) creates a dataset consisting of 92 traffic scenarios and used crowd-sourced annotations to compare an instance of the rulebook pre-ordered priority structure from Censi et al. (2019) with models obtained using machine learning with varying degree of interpretability, such as Bayesian networks, decision trees, and logistic regression.

3. Formal verification

Formal verification is the process of verifying that all the possible executions of a system satisfy a formal specification, such as safety or a TL formula. While autonomous system verification can proceed with incomplete or gray-box models by combining statistics with structural reasoning (see Fan, 2019; Fan et al., 2018, for treatments of such approaches), in this review we focus on a formal, traditional approach to verification that requires a model of the system. The model typically consists of the ego vehicle and its environment in autonomous driving. Online verification is performed during the execution of the system and it only requires checking the satisfaction of a specification against all possible behaviors originating at the current time.

Since finding a suitable non-deterministic model of the environment as well as formalizing all ROTRs are challenging, most formal verification methods focus on vehicle following (which includes ACC, emergency braking systems, and platooning) and lane keeping/changing. In the first part of this section, we focus on these types of maneuvers. Afterwards, we discuss more general methods for arbitrary traffic situations. A summary of the reviewed papers is listed in Table 2. All reviewed papers performing formal verification use safety as specifications. Consequently, we do not list the considered type of specification in Table 2. Instead, we list whether the method is applied offline (during design time) or online (during vehicle operation) and whether the approach can be applied to mixed traffic, i.e., traffic with autonomous vehicles, manually-driven vehicles, and other forms of non-automated movements, such as riding a bicycle or walking. Some approaches require that all vehicles are autonomous or that the behavior of other traffic participants is known. For instance, some papers assume that a leading vehicle moves with constant velocity. These would not necessarily prove safety in mixed traffic.

Theorem proving. The first formally-correct controllers have been developed for vehicle following and verified using handwritten proofs, see e.g., Alvarez and Horowitz (1999), Dolginova and Lynch (1997) and Lygeros et al. (1996); an extension to game-theoretic techniques for cooperative controlled vehicles is presented in Lygeros et al. (1998). Lane following is especially amenable for handwritten proofs since it only requires one-dimensional movement along a lane, and the corresponding dynamics are monotone (Angeli & Sontag, 2003). To avoid human error in proofs, a theorem prover is used in Loos et al. (2011, 2013), which, however, assumes that all vehicles are automated. Theorem proving has also been extended to prove the safety of lane changes by reserving space for vehicles (Hilscher et al., 2013; Linker & Hilscher, 2013). An advantage of theorem proving is that the number of traffic participants is unbounded, however, it typically cannot be used for online verification, because most theorem provers are not fully automatic.

Barrier certificates. Barrier certificates verify systems by proving that a barrier between the set of initial states and unsafe states always exists. This idea was applied to ACC (Ames et al., 2014) and was experimentally validated in Mehra et al. (2015). An extension for varying velocities of the leading vehicle and lane keeping is presented in Nilsson et al. (2016) and Xu et al. (2018), respectively. Barrier certificates are particularly useful for proving the correctness of specific controllers, such as controllers for following vehicles and staying within a lane (the construction of the controllers is discussed in Section 5). So far, no approach has been presented to automatically create barrier certificates for a given traffic situation so that no universal online verification scheme has yet been realized.

Worst-case behaviors. Due to the previously-mentioned monotone dynamics of vehicles staying within the same lane, vehicle-following problems can be verified through worst-case behaviors. Those are used to safeguard exchangeable nominal controllers, by embedding them in an emergency controller that only engages if the nominal controller performs an unsafe action (Magdici & Althoff, 2017; Wang et al., 2020). This idea is also applied to vehicle platooning (Ligthart et al., 2018) and was later extended to handle cut-in vehicles and also lane changes of the leading vehicle (Althoff et al., 2021). A lane change of the leading vehicle can suddenly reveal an occluded obstacle, which either requires detecting further vehicles ahead or assuming a standing obstacle within the occluded region.

By conservatively separating the dynamics into a lateral and longitudinal dynamics, one can also use worst-case behaviors for lane changes. The work in Pek et al. (2017b) uses safe distances to ensure that lane changes are safe, despite the uncertainty in the movement of other traffic participants. The special case of a fixed sinusoidal lane change and given accelerations of surrounding traffic participants is shown in Jula et al. (2000). The approach in Pek et al. (2017b) is also used to safeguard reinforcement learning for lane changes (Mirchevska et al., 2018). Additional formal rules are added in Naumann et al. (2019b) under which a lane change is deemed to comply with ROTRs. As a special case, swerve maneuvers are verified in de Iaco et al. (2020).

Reachability analysis. Currently, the most popular verification technique for AVs is reachability analysis. Reachability analysis automatically verifies systems by computing the set of reachable states. If no reachable state enters an unsafe region, safe behavior is proven (Alam et al., 2014; Park & Özgüner, 2012). Invariant sets are a special case of reachable sets in which the state of a system stays indefinitely. Thus, if the invariant set does not contain any unsafe states, correctness can be proven analogously to reachability analysis. Most of the reviewed papers that use reachability analysis and invariant sets perform both verification and control synthesis. Here, we focus on verification. We revisit some of these papers and discuss their control strategies in Section 5.

Safety is proven for ACC in Dai and Koutsoukos (2016), Sadraddini et al. (2017), Smith et al. (2016) and Wongpiromsarn et al. (2012) using invariant sets. Since the relatively simple task of safe vehicle following can be verified by handwritten proofs, only Stursberg et al. (2004) verified this problem using reachability analysis. Another sub-problem we consider is the problem of verifying whether a safe solution still exists. This can be used to prove that an aggressive evasive maneuver has to be executed or that a collision can no longer be avoided and a collision mitigation procedure needs to be initiated. The work in Schmidt et al. (2006) computed the reachable set of the ego vehicle to check whether it becomes empty—in this event, the ego vehicle is in an inevitable collision state (Bouraine et al., 2012). To reduce the conservatism of that work, the velocity information within the reachable set and road geometry are explicitly considered in Söntges and Althoff (2015). This work was later extended to compute the time to react in a formal way, i.e., the remaining time to avoid a potential collision (Koschi et al., 2018b). Instead of determining whether the current state is an inevitable collision state, one can also compute the set of inevitable collision states (Lawitzky et al., 2014); however, this is computationally expensive and thus currently not real-time capable. The work in Pek et al. (2017a) does not only compute the first point in time when a collision is possible, but also the last point in time.

The set of possible scenarios for fully autonomous driving cannot be constrained in the same way as it is done for vehicle following or lane changing. Thus, most approaches compute reachable sets online for fully autonomous driving so that all occurring situations are considered—an offline procedure might have missed potentially dangerous situations. To the best knowledge of the authors, the first work using online reachability analysis for autonomous driving is Althoff et al. (2010). The disadvantage of that work is that it requires that vehicles communicate with each other and that they have to travel with constant velocity. These restrictions were later removed in Althoff and Dolan (2011); however, the used vehicle model is just a single-track model. A method to consider high-dimensional models through non-deterministic low-dimensional models is presented in Althoff and Dolan (2012) and Kousik et al. (2017); this approach is extended in Schürmann et al. (2017) to show conformance with real vehicles. The first work that applied online reachability analysis to a real vehicle is Althoff and Dolan (2014); later works can be found in Ahn et al. (2021), Lin et al. (2020), Pek et al. (2020), Stahl and Diermeyer (2021) and Vaskov et al. (2019). Although this approach works in principle for all kinds of traffic situations, it does not contain an algorithm for computing the reachable set of other traffic participants on arbitrary road networks—this is addressed in Althoff and Magdici (2016) and implemented by the tool SPOT (Koschi & Althoff, 2017a). Further developments, in particular with respect to handling occlusions, are presented, e.g., in Koschi and Althoff (2021), Nager et al. (2019), Neel and Saripalli (2020) and Orzechowski et al. (2018). Online reachability analysis was recently used to safeguard reinforcement learning for AVs (Shao et al., 2021). To engage safe but aggressive maneuvers more comfortably, the work in Naumann et al. (2019a) additionally considers probabilistic information to slow down the AV early when a dangerous situation is likely to occur. An approach that combines ideas from contract-based verification with reachability analysis is presented in DeCastro et al. (2020) and Liebenwein et al. (2020); however, this approach is not yet real-time capable. Other approaches, such as Fan et al. (2018), O'Kelly et al. (2017) and Völker et al. (2019), are primarily designed for formal offline verification for specific scenarios. To reduce computation times for online use, some approaches consider exemplary traces instead of the set of possible solutions (Esterle et al., 2019).

4. Monitoring

Monitoring (or runtime verification) refers to lightweight formal verification methods designed to check system executions against formal requirements. The main difference from the verification approaches discussed in Section 3 is that the latter reason over all possible system executions and uncertainties. Online monitoring refers to checking the current execution of a system, while offline monitoring is the process of checking a (finite set of) recorded execution(s). In most monitoring applications, including autonomous driving, execution traces are long, and are only available incrementally. Waiting for and storing an entire execution trace and then performing offline monitoring can be expensive. Moreover, in offline monitoring, verification might occur too late to allow the system to recover or take a shutdown action. For this reason, online monitoring is the prevalent technique in autonomous driving.

Safety specifications. Monitoring for compliance with safety specifications is presented in Chou et al. (2020), Du et al. (2020), Kojchev et al. (2020) and Ody (2017). The authors of Kojchev et al. (2020) use backward reachability analysis to construct the monitor. Monitoring for Multi-Lane Spatial Logic is considered in Ody (2017), where the authors show that formula satisfaction can be mapped to feasibility of formulas in the first-order theory of real-closed fields. Runtime monitoring techniques based on predictions of future behaviors of traffic participants are developed for safety specifications in Chou et al. (2020) and Du et al. (2020). The authors of Du et al. (2020) present a pedestrian intent estimation framework that can predict future pedestrian trajectories, and integrate it into a reachability-based online monitoring and decision making scheme. A predictive runtime monitoring method for estimating future vehicle positions and the probability of collisions with obstacles is presented in Chou et al. (2020). Their approach combines Bayesian inference techniques and set-valued reachability analysis to approximate future positions of a vehicle.

LTL. Similar to verification and synthesis, most monitoring techniques against LTL formulas require converting the LTL formula to an automaton. Depending on the structure of the formula, this automaton can be a finite state automaton, a Büchi automaton, or a Rabin automaton. Monitoring against ROTRs expressed as LTL formulas is performed in Esterle et al. (2019, 2020) and Rizaldi et al. (2017). The authors of Rizaldi et al. (2017) focus on overtaking and safe distance keeping. In Esterle et al. (2020), ROTRs are modeled as objects called RuleMonitors, which are then used to monitor rule compliance through simulation and comparison against a public dataset. In Esterle et al. (2019), the authors develop an LTL offline monitoring method that does not require the computation of a complete automaton from the specification and the partition of the ego vehicle's continuous environment, but rather constructs a smaller automaton corresponding to a specific traversal of the quotient graph.

STL and MTL. As mentioned previously, STL and MTL are particularly fit for monitoring due to their quantitative semantics (i.e., robustness functions that quantify the degree of satisfaction or violation with respect to a formal specification). In Hekmatnejad et al. (2019), the authors encode the RSS model in STL, and perform monitoring of two RSS specifications (i.e., keeping a safe distance to front and side vehicles) on traffic scenarios from CommonRoad (Althoff, Koschi, & Manzinger, 2017) using S-TALIRO (Annpureddy, Liu, Fainekos, & Sankaranarayanan, 2011). STL specifications for vehicle following are encoded using a special, block-sparse Mixed Integer Programming (MIP) problem structure in Sahin et al. (2020), which is exploited to increase the efficiency of the computation involved in monitoring.

The authors of Aasi et al. (2021) propose STL monitoring to compute corrections in a two-level AV control architecture. At the top level, simple representations of the environment and vehicle dynamics are used to derive controllers using an MPC approach. At the bottom level, STL runtime monitoring techniques, together with detailed representations of the environment and vehicle dynamics, are used to compensate for the mismatch between the simple models used in the MPC and the real complex models.

A runtime monitoring algorithm that checks for violations of properties written in a future-bounded, propositional MTL by an experimental AV is presented in Kane et al. (2015). The algorithm incrementally takes as input a system state, which maps propositions to either true or false, and a MTL formula, and eagerly checks the state trace for violations. It uses an iteration based on dynamic programming to reduce the input formula as soon as possible using history—summarizing structures and formula-rewriting-based simplifications.

5. Control synthesis

The control synthesis problem is to find controllers for AVs that minimize a cost, while satisfying physical constraints and formal rules. This section presents commonly used approaches for formal control synthesis for AVs. We first review papers that use automata-based techniques for control from specifications given in LTL or fragments of LTL. We then focus on optimization-based approaches that exploit the quantitative semantics (robustness) of concrete-time temporal logics such as STL (see Belta & Sadraddini, 2019, for a review and comparison of automata-based and optimization-based approaches to formal synthesis) and on papers that use Control Barrier Functions (CBF) and Control Lyapunov Functions (CLF). The most popular formal synthesis techniques for AV control involve reachability analysis and invariant sets, and most of this section reviews such papers. Finally, we review papers using falsification techniques and machine learning. A summary of the reviewed papers is listed in Table 4.

Automata-based synthesis. For formal rules written in LTL and fragments of LTL, the formal synthesis problem can be mapped to solving an automaton game. In short, this method is based on translating the specification to an automaton, such as a Finite State Automaton (FSA), Büchi automaton, or Rabin automaton, and then combining this with a finite abstraction of the dynamics of the system. The control strategy is generated by graph analysis, or by solving an automaton (Büchi or Rabin) game (Belta et al., 2017).

The authors of Vasile et al. (2017) propose a receding-horizon approach to synthesize controllers in static environments without any other traffic participants by solving a minimum-violation motion planning problem. This problem is formulated given a conflicting set of customer demands (e.g., pick up or drop off a customer at certain locations within desired deadlines) and ROTRs specified in scLTL. A delay penalty is associated with meeting customer demands and is minimized in the global long-term routing, while a lower-level RRT* (see Karaman, Walter, Perez, Frazzoli, & Teller, 2011; LaValle & Kuffner, 2001) planner is used to compromise between delay penalty and violating the ROTRs, while guaranteeing safety. The scLTL specifications are converted to deterministic automata with weighted transitions that are used to capture the level of violation based on the priorities assigned to the ROTRs. The related work in Wongpiromsarn et al. (2021) develops an incremental sampling-based approach to solve minimum-violation planning problems for static environments considering multiple, potentially conflicting, ROTRs specified in si-FLTLGX that have different priorities.

Optimization-based synthesis. For formal rules written in logics with real-time and real-valued specifications such as STL and MTL, the control synthesis problem can be formulated as an optimal control problem, where the cost captures traditional objectives, such as energy spent and/or distance traveled. Vehicle limitations, such as acceleration and turning radius, are modeled as constraints. Boolean rule satisfactions can also be imposed as constraints. Alternatively, rule satisfactions can be maximized by adding weighted aggregations of their robustness values to the cost. An example of this approach can be found in Sahin et al. (2020), where the authors translate selected ROTRs formulated as STL specifications into a set of mixed-integer and linear constraints and solve the synthesis problem for a simplified vehicle motion model with bounded additive uncertainty using MIP techniques.

Table 4
Papers organized by control approaches and application areas.

	Automata	Optimization	CBFs& CLFs	Reachability[a]	Machine learning	Falsification
Vehicle following			Ames et al. (2014)	Alam et al. (2014), Althoff et al. (2021), Ligthart et al. (2018), Magdici and Althoff (2017), Nilsson et al. (2016), Park and Özgüner (2012), Sadraddini et al. (2017)		
Lane keeping/ changing				Berntorp et al. (2017), Hoehener et al. (2016)		
Other	Vasile et al. (2017), Wongpiromsarn et al. (2021)	Aasi et al. (2021), Jha et al. (2018), Sahin et al. (2020)	Xiao et al. (2021), Xu et al. (2018)	Berntorp et al. (2020), Bouraine et al. (2012), Brüdigam et al. (2021), Danielson et al. (2020), DeCastro et al. (2020), Falcone et al. (2011), Gerdts and Xausa (2013), Herbert et al. (2017), Jalalmaab et al. (2017), Kianfar et al. (2013), Macek et al. (2009), Magdici and Althoff (2016), Parthasarathi and Fraichard (2007), Pek et al. (2020), Schäfer, Manzinger, and Althoff (2021), Smith et al. (2016), Soloperto et al. (2019), Vanholme et al. (2013), Wu and How (2012)	Cho et al. (2019), Li, Rosman, Gilitschenski, DeCastro et al. (2021), Mirchevska et al. (2018), Rong and Luan (2020), Sadigh et al. (2017), Shao et al. (2021)	Corso and Kochenderfer (2020), DeCastro et al. (2020), O'Kelly et al. (2017), Tuncali et al. (2018, 2020)

[a]Reachability refers to works on reachability analysis, control and positive invariant sets and ICS.

A two-level control architecture for a fully autonomous system in a deterministic environment with real-time performance is proposed in Aasi et al. (2021). At the top level, ROTRs formulated as STL specifications are translated into MIP constraints and imposed in a linear MPC problem defined over a simple representation of the environment and vehicle dynamics. At the bottom level, specification-based run-time monitoring techniques, together with detailed representations of the environment and vehicle dynamics, are used to compensate for the mismatch between the simple models used in the MPC and the real complex models. The authors of Jha et al. (2018) propose a correct-by-construction algorithm to control AVs under perception uncertainty with probabilistic correctness guarantees specified as C2TL formulas. By approximating C2TL constraints with a set of mixed-integer constraints, the synthesis problem is formulated as a scalable second-order cone program that can be solved using off-the-shelf optimization tools.

Synthesis through CBFs and CLFs. The control synthesis problem has also been formulated as an optimal control problem in which the satisfaction of the rule(s) and the vehicle's state limitations are enforced by CBFs, and convergence to desired states (e.g., vehicle following) is achieved through CLFs. These ideas are proposed in Ames et al. (2014) and used for ACC, in which CBFs are associated with safe sets, and the inequality constraints that ensure forward invariance of the set are imposed over the control strategies as optimization constraints. The unified optimal control problem with simultaneous safety (CBFs) and ACC objectives (CLFs) is solved through a sequence of computationally efficient Quadratic Programs (QPs). Building on this approach, the authors of Xu et al. (2018) develop controllers with probabilistic correctness guarantees for simultaneous lane keeping and ACC obtained using fast QPs. The work in Xiao et al. (2021) uses high-order CBFs (i.e., CBFs that can accommodate constraints with high relative degree) to guarantee satisfaction of a set of prioritized formal rules including lane keeping, following speed limits, and maintaining clearance with other traffic participants.

Reachability techniques, invariant sets, and ICSs. Many recent works combine reachability analysis and control techniques in hierarchical planning architectures. These are usually composed of a high-level route planner and a low-level controller used to follow the constructed trajectory, while guaranteeing the satisfaction of ROTRs and physical constraints of AVs. As already stated, the reachability analysis of most papers reviewed here is discussed in Section 3. Here, we focus on the planning and control aspects.

Reachability techniques have been extensively investigated for vehicle platooning (Alam et al., 2014; Park & Özgüner, 2012), ACC (Kianfar et al., 2013), and path planning (Gerdts & Xausa, 2013) for AVs in complex, safety-critical situations. In Wu and How (2012), the authors model obstacles as single speed, maximum turn-rate unicycle robots and define velocity obstacle occupancy sets as unions of the sets of all reachable points by the obstacles. This study proves that subject to certain initial conditions, an infinite-horizon iterative planner guarantees collision avoidance for all times with respect to moving obstacles that have constrained dynamics. The work in Liebenwein et al. (2020) uses reachability analysis to concurrently solve compositional verification for the local road model and synthesizes assume-guarantee contracts to certify the safety of the given controllers. To improve the computation time for online applications, multiple works study offline pre-computations of reachable sets (Herbert et al., 2017; Schürmann et al., 2017).

Control invariant sets are mainly used to guarantee the indefinite feasibility of MPC frameworks despite limited prediction horizons (Jalalmaab et al., 2017). Safety constraints normally make the admissible domain of the MPC optimization problem non-convex. Convexification of the safety constraint presented in Jalalmaab et al. (2017) makes the computation of the control invariant sets fast for real-time applications. However, it reduces the set of feasible solutions. The work in Schäfer et al. (2021) identifies collision-free driving corridors that represent spatio-temporal constraints for motion planning using set-based reachability analysis. In Jalalmaab et al. (2017), look-up tables are generated offline to determine the control invariant sets in real-time (Jalalmaab et al., 2017). The work in Nilsson et al. (2016) formulates ACC specifications as LTL formulas assuming varying velocities of the leading vehicle, and designs two synthesis methods: one based on control invariant set computations on the continuous state space domain, and one using set computations on a non-deterministic finite state abstraction of the system. In Soloperto et al. (2019), the authors use smooth over-approximations of the collision avoidance constraints and present a tube-based robust MPC framework with formal guarantees on recursive feasibility and satisfaction of the constraints. Synthesis of correct-by-construction centralized and distributed ACC policies for vehicle platooning with infinite-time collision avoidance guarantees in the presence of bounded additive disturbances are investigated in Sadraddini et al. (2017) using robust control invariant sets and QP optimization. In Smith et al. (2016), the authors use control invariant sets for synthesizing controllers for an AV with linear parameter-varying dynamics, ACC, and lane keeping subsystems, which are robust against additive parametric uncertainties. The work in Hoehener et al. (2016) designs a safety supervisor for lane departure assist systems to keep a semi-autonomous vehicle in a lane using control invariance techniques.

Positive invariant sets are augmented with state-feedback control in Berntorp et al. (2017) to guarantee collision-free closed-loop trajectory tracking under modeling errors in overtaking and lane-change maneuvers. The framework proposed in this study relies on graph search to find invariant sets over a finite set of lateral displacements on the road. A similar idea is proposed in Berntorp et al. (2020), which integrates motion planning and state-feedback control. The authors of Danielson et al. (2020) investigate robust positive invariant set motion planners for systems with persistently varying disturbances and parametric model uncertainty. The invariant sets are parameterized using a pre-computed input-to-state Lyapunov function.

ICS are employed in safe motion planning where a safety checker determines whether a system motion could lead to an ICS (Fraichard & Asama, 2004). To efficiently build a conservative approximation of the ICS set rather than checking for collisions for all possible future trajectories of infinite duration, the authors of Parthasarathi and Fraichard (2007) propose a principle to select a finite subset of the possible future trajectories through *imitating maneuvers*, in which the AV tries to duplicate the object's behavior. The works in Bouraine et al. (2012) and Macek et al. (2009) propose a less conservative version of ICS, called braking ICS, which is used to guarantee that a collision could occur only when the AV was at rest.

Fail-safe maneuvers have been proposed to reach time-invariant safe states such that safety for an infinite time horizon can be ensured. The works in Magdici and Althoff (2016, 2017) consider the most likely trajectory of other traffic participants for ACC control design, and maintain an emergency maneuver based on an over-approximation of the predicted occupancy set. Cooperative ACC is investigated in Ligthart et al. (2018), in which a predefined gradual braking strategy overrides the nominal controller to guarantee collision avoidance. In a more recent study, the authors of Althoff et al. (2021) investigate fail-safe controllers for ACC in various driving conditions studied in the literature, such as full braking of the lead vehicle as well as more complex cut-in scenarios while taking into account uncertainties.

Considering all dynamically feasible behaviors of other traffic participants may over-conservatively limit the maneuverability of the AV. The work in Vanholme et al. (2013) proposes a nominal control framework for highly automated driving on highways (e.g., with ACC and lane-changing functionalities), which considers legal and reasonably foreseeable nonlegal behavior of other traffic participants, and designs failure functioning trajectories for critical situations. Legal safety is guaranteed in Pek et al. (2020), which finds legal and dynamically feasible behaviors of traffic participants using online reachability, and proposes a fail-safe trajectory to a standstill state in designated safe areas. In Brüdigam et al. (2021), the authors use Stochastic Model Predictive Control (SMPC) to reformulate hard constraints (e.g., for lane change and collision avoidance) in uncertain environments into probabilistic chance constraints. A fail-safe trajectory is planned using reachability analysis, which overrides the nominal SMPC controller.

Falsification. Falsification can be used to validate safety requirements (Corso & Kochenderfer, 2020) and to provide guidance on control design (Tuncali et al., 2018). A simulation-based adversarial test generation framework for AVs to check closed-loop properties of autonomous driving systems has been studied in Tuncali et al. (2020). The authors of O'Kelly et al. (2017) propose a hierarchical control stack (including an ACC planner, trajectory planner and trajectory tracker), to reach a goal within a fixed time and meet selected ROTRs formulated in MTL. In that work, S-TALIRO (Annpureddy et al., 2011) and dReach (Kong, Gao, Chen, & Clarke, 2015) are used to evaluate the existence of a falsifying trajectory under different types of uncertainty, including uncertainty in AV's perception and non-determinism in the dynamics of other vehicles. In DeCastro et al. (2020), safety contracts are constructed by alternatively using falsification to create counterexamples for collision-free specifications, and employing them as obstacles in a reach-avoid problem solved through reachability analysis.

Machine learning and control. The quantitative semantics (robustness function) of formal rules has enabled a growing interest in combining machine learning techniques with TL-guided control for autonomous driving. The authors of Cho et al. (2019) formulate urban driving ROTRs in STL and combine the benefits of deep learning and MPC to propose controllers that can reason about the future behavior of nearby vehicles and behave close to human experts. Allowing relaxation of the rule constraints up to the predicted margin to satisfaction of each rule provides insight on the importance of rules as described in Section 2.3. The related work Li, Rosman, Gilitschenski, DeCastro et al. (2021) integrates a set of parametric STL rules into a neural network for trajectory prediction. A differentiable STL robustness of the rules is optimized using gradient techniques.

An increasing number of papers propose formal methods for reinforcement learning in AV control. The work in Rong and Luan (2020) proposes a hierarchical structure with a high-level deep reinforcement learning model and a low-level (adapted RRT*) motion planner. The reinforcement learning reward function and the motion planner cost function are formulated using quantitative robustness of LTL specifications that represent ROTRs. Learning the suitable reward function based on human's preferences is studied in Sadigh et al. (2017). Safeguarded reinforcement learning for lane change AV control is proposed in Mirchevska et al. (2018).

6. Discussion and remaining challenges

In this section, we discuss remaining challenges and directions for future work.

Formal specifications. Even though, as shown above, formally specifying ROTRs has received a lot of attention recently, it is still one of the main challenges facing the AV community. Ideally, we would like to have a computational framework allowing to automatically map sets of traffic laws written in plain English, such as state driving laws in the US, or the Vienna Convention on Road Traffic, or the German Road Traffic Regulation, to sets of formal rules, such as the TLs reviewed in Section 2. This is a very daunting task.

First, a specification language should be chosen. As already noted, most existing works use LTL-type languages, as opposed to branching time logics such as CTL, which are popular in the formal methods community. The explanation for this is most probably that translation from natural language to LTL is easier and less prone to error than CTL (Memon, 2003). However, the few existing works that propose frameworks to translate natural language to formulas of logics such as LTL (Lignos, Raman, Finucane, Marcus, & Kress-Gazit, 2014; Nikora & Balcom, 2009) (see also Brunello, Montanari, & Reynolds, 2019, for a review) in related fields are restrictive and difficult to automate. Second, off-the-shelf TLs might be unnecessarily expressive, and as a result, the corresponding verification and synthesis algorithms too expensive for AVs. Defining languages that are specifically tailored to autonomous driving is a current direction of research. Third, based on our own experience and of others, most probably there exists a rather small set of formal rules (primitives) in a properly chosen logic, such that any traffic law can be written as a (temporal, Boolean) combination of these primitives. Choosing the primitives, the composition rules, and the logic, is a challenging and open problem.

As reviewed in Section 2, recent works propose (pre-, partial, total) orders and/or weights to model rule relative importance, or priorities. The problems of trajectory selection and control synthesis while satisfying rule priority structures are not well understood. In particular, dealing with priorities under uncertainty is a widely open problem. Consider, for example, the problem of trajectory selection under classification uncertainty. Assume there are two rules: pedestrian and parked vehicle clearance, with the first being more important. If pedestrian classification is less reliable than parked vehicle classification, it is not clear how to perform trajectory selection when both pedestrians and parked vehicles are detected.

In most traffic scenarios, parameterizing formal specifications is challenging. For example, in pedestrian clearance, it is not obvious what combination of vehicle distance to pedestrian and approach speed would not look dangerous to the pedestrian. Some of the reviewed papers show that, when the specification is safety, such parameters can be learned from data. When the specifications are formulas of LTL, STL, or MTL, the problem is more complicated. Works from the controls and formal methods communities suggest that STL robustness can be used to find parameters in rules with given structures by solving optimization problems (Asarin, Donzé, Maler, & Nickovic, 2011). More recent works (Bombara & Belta, 2021) show that the formula structures can be learnt from data as well, which can prove useful for AV applications. For example, safely engaging a curve might require a TL combination of positions, speeds, and accelerations that is not easy to formulate, but can be learnt from good driving behavior. Finally, even with known rules, specifying their relative importance (priorities) is a challenging problem. An encouraging direction, supported by very recent preliminary results (Helou et al., 2021), is to learn them from data.

Sets of ROTR formalized by translating traffic rules or by learning from data can be inconsistent and/or incomplete. The first can be dealt with using priority structures, as reviewed in Section 2.3, and corresponding iterative control schemes. An interesting alternative is to generate rules that are consistent and complete by construction. Very recently, the authors of Phan-Minh, Cai, and Murray (2019) addressed this problem using a distributed assume-guarantee structure. However, characterizing consistency and completeness of sets of ROTRs is an open and challenging problem.

Verification. Most of the verification approaches reviewed here focus on well-defined use cases for autonomous driving, such as safe vehicle following. In such a well-defined scenario, one can verify the system offline by making certain assumptions; e.g., the leading vehicle is not allowed to perform a lane change. However, when removing this assumption, a standing vehicle could suddenly be revealed so that the ego vehicle is in an inevitable collision state. It is necessary to be able to continuously monitor such situations and assume that obstacles can be present in occluded spaces. This requires to develop online verification methods that can react to each situation appropriately—offline verification methods are infeasible for fully autonomous driving at the system level due to the large amount of test cases required to obtain a meaningful coverage. Online verification methods have to be able to consider all possible legal behaviors of surrounding traffic participants to verify that the planned action is not causing an accident. This is one of the main remaining challenges in this area. In order to avoid requiring that online approaches are real-time capable and that a new safe solution can always be found, most current approaches use fail-safe maneuvers that are executed in case no safe maneuver can be computed on time. Another remaining challenge is that verification methods currently focus on safety specifications, while methods for verifying more complicated rules are in their infancy. This is because in contrast to monitoring approaches, verification methods have to verify a system given all uncertainties. The combined challenge of verifying complicated specifications for all possible executions of the system and its environment is still an unsolved problem.

Synthesis. There is an increasing number of works that propose temporal logics as formal specification languages for autonomous driving. The methods for synthesis of control strategies from temporal logic specifications can be roughly grouped into two categories: automata-based and optimization-based approaches. The first group is mostly used for LTL-type specifications, and synthesis maps to solving an automaton game (e.g., Büchi or Rabin games). This is usually expensive, and, as a result, not suited to real-time control of autonomous vehicles. Most of the reviewed papers in formal synthesis belong to the second category, for which the specifications are given in concrete-time TLs, such as STL and MTL, which have quantitative semantics. Optimization methods can be subdivided into MIP-based and gradient-based approaches. Computational complexity is still a limitation for both methods. Current research is aimed at defining meaningful and smooth convex robustness functions that can be efficiently used in optimization. Most of existing approaches are based on MILPs, which only apply to linear dynamics. A current research direction is their extension to realistic vehicle dynamics. CBF-based methods are fast but myopic, and the corresponding QPs can easily become infeasible, which is one of the main challenges in this approach. Another challenge and direction of future work is designing frameworks for automatic construction of barrier functions for a given formal rule or state constraint.

Reachability analysis remains the most used technique in AV control. As shown in this review, various tools are available for computing the reachable sets on arbitrary road networks, which allows such techniques to be used for online control. However, reducing conservativeness caused by over-approximation of reachable states, while maintaining guarantees on safety, remains a challenge.

Uncertainty, misclassification, and sensor noise. While formal methods for discrete systems often do not require to consider uncertainties, this is of paramount importance for physical systems, such as AVs. Undoubtedly, the main uncertainty arises due to the unknown future behavior of surrounding traffic participants—even if a traffic participant performs full braking, the ego vehicle has to ensure safety. A second major source of uncertainty originates from the sensing of surrounding traffic participants whose positions, velocities, and headings are subject to measurement uncertainties. This also applies to the proprioceptive sensors of the ego vehicle, whose noise has to be considered when predicting the maximum deviation from a planned trajectory. Besides uncertainty in physical measurements, another challenge is dealing with misclassifications of traffic participants.

Formal verification and synthesis methods have to consider the above-mentioned uncertainties in their entirety. In contrast, monitoring approaches only have to evaluate a concrete evolution of a traffic scene. For instance, when the classification of a traffic participant is uncertain, formal approaches require to compute with all remaining classification hypotheses. As a consequence, most formal verification and synthesis approaches compute with sets to ensure that disturbances and sensor noise are appropriately considered. The challenge here is to consider all sources of uncertainties. While some papers focus on the uncertain future movements of other traffic participants, others only focus on the tracking error of the ego vehicle when following a planned trajectory, yet others only focus on the unknown number and states of occluded traffic participants. Obviously, formal verification can only be accomplished for the real system—and not just its mathematical model—if all sources of uncertainty of the real-world are considered. To address the potential model mismatch, several works have developed conformance checking techniques for autonomous vehicles (see Roehm, Oehlerking, Woehrle, & Althoff, 2019, for a recent review). Nevertheless, this area of research is underrepresented in our point of view.

7. Conclusion

In this paper, we reviewed recent works that use formal methods for autonomous driving. We covered formal specifications for rules of the road, with particular emphasis on temporal logics. Verification, monitoring, and control synthesis techniques from such specifications were reviewed. We restricted our attention to ego-centric approaches and system-level methods that focus on the behavior of an autonomous vehicle in its entirety, rather than specific software code within the vehicle. In addition, we included a critical discussion on the field and discussed remaining challenges and directions for future research.

We believe this paper will be of interest to a large audience, which includes academia and the rapidly growing AV industry. Control theorists will learn how control techniques and basic stability concepts are used in autonomous driving. They will also get exposure to formal methods techniques and their connection to dynamical systems. Computer scientists working in formal methods will see how the expressivity of temporal logic formulas can be exploited to formalize traffic laws. Last but not least, this paper will be of interest to engineers working on developing autonomous cars. We also hope that this paper will help form a community of researchers and educators interested in using tools and concepts from formal methods in the rapidly increasing area of autonomous driving.

Acknowledgment

The authors gratefully acknowledge financial support by the European Commission project justITSELF under grant number 817629.

References

Aasi, E., Vasile, C. I., & Belta, C. (2021). A control architecture for provably-correct autonomous driving. In *Proc. of the American control conference* (pp. 2913–2918).

Ahn, H., Berntorp, K., Inani, P., Ram, A. J., & Di Cairano, S. (2021). Reachability-based decision-making for autonomous driving: Theory and experiments. *IEEE Transactions on Control Systems Technology*, *29*(5), 1907–1921.

Alam, A., Gattami, A., Johansson, K. H., & Tomlin, C. J. (2014). Guaranteeing safety for heavy duty vehicle platooning: Safe set computations and experimental evaluations. *Control Engineering Practice*, *24*, 33–41.

Althoff, M., Althoff, D., Wollherr, D., & Buss, M. (2010). Safety verification of autonomous vehicles for coordinated evasive maneuvers. In *Proc. of the IEEE intelligent vehicles symposium* (pp. 1078–1083).

Althoff, M., & Dolan, J. M. (2011). Set-based computation of vehicle behaviors for the online verification of autonomous vehicles. In *Proc. of the 14th IEEE conference on intelligent transportation systems* (pp. 1162–1167).

Althoff, M., & Dolan, J. M. (2012). Reachability computation of low-order models for the safety verification of high-order road vehicle models. In *Proc. of the American control conference* (pp. 3559–3566).

Althoff, M., & Dolan, J. M. (2014). Online verification of automated road vehicles using reachability analysis. *IEEE Transactions on Robotics*, *30*(4), 903–918.

Althoff, M., Koschi, M., & Manzinger, S. (2017). CommonRoad: Composable benchmarks for motion planning on roads. In *Proc. of the IEEE intelligent vehicles symposium* (pp. 719–726).

Althoff, M., & Magdici, S. (2016). Set-based prediction of traffic participants on arbitrary road networks. *IEEE Transactions on Intelligent Vehicles*, *1*(2), 187–202.

Althoff, M., Maierhofer, S., & Pek, C. (2021). Provably-correct and comfortable adaptive cruise control. *IEEE Transactions on Intelligent Vehicles*, *6*(1), 159–174.

Alvarez, L., & Horowitz, R. (1999). Safe platooning in automated highway systems part I: Safety regions design. *Vehicle System Dynamics*, *32*(1), 23–55.

Ames, A. D., Grizzle, J. W., & Tabuada, P. (2014). Control barrier function based quadratic programs with application to adaptive cruise control. In *Proc. of the 53rd IEEE conference on decision and control* (pp. 6271–6278).

Angeli, D., & Sontag, E. D. (2003). Monotone control systems. *IEEE Transactions on Automatic Control*, *48*(10), 1684–1698.

Annpureddy, Y., Liu, C., Fainekos, G., & Sankaranarayanan, S. (2011). S-TALIRO: A tool for temporal logic falsification for hybrid systems. In *International conference on tools and algorithms for the construction and analysis of systems* (pp. 254–257).

Aréchiga, N. (2019). Specifying safety of autonomous vehicles in signal temporal logic. In *Proc. of the IEEE intelligent vehicles symposium* (pp. 58–63).

Asarin, E., Donzé, A., Maler, O., & Nickovic, D. (2011). Parametric identification of temporal properties. *Runtime Verification*, *7186*, 147–160.

Baier, C., & Katoen, J. P. (2008). *Principles of model checking*. The MIT Press.

Belta, C., & Sadraddini, S. (2019). Formal methods for control synthesis: An optimization perspective. *Annual Review of Control, Robotics, and Autonomous Systems*, *2*, 115–140.

Belta, C., Yordanov, B., & Gol, E. A. (2017). *Formal methods for discrete-time dynamical systems*. Springer.

Berntorp, K., Bai, R., Erliksson, K. F., Danielson, C., Weiss, A., & Cairano, S. D. (2020). Positive invariant sets for safe integrated vehicle motion planning and control. *IEEE Transactions on Intelligent Vehicles*, *5*(1), 112–126.

Berntorp, K., Weiss, A., Danielson, C., Kolmanovsky, I. V., & Di Cairano, S. (2017). Automated driving: Safe motion planning using positively invariant sets. In *Proc. of the IEEE 20th international conference on intelligent transportation systems* (pp. 2247–2252).

Bombara, G., & Belta, C. (2021). Offline and online learning of signal temporal logic formulae using decision trees. *ACM Transactions on Cyber-Physical Systems*, [ISSN: 2378-962X] *5*(3), 1–23.

Bouraine, S., Fraichard, T., & Salhi, H. (2012). Provably safe navigation for mobile robots with limited field-of-views in dynamic environments. *Autonomous Robots*, *32*(3), 267–283.

Brüdigam, T., Olbrich, M., Wollherr, D., & Leibold, M. (2021). Stochastic model predictive control with a safety guarantee for automated driving. *IEEE Transactions on Intelligent Vehicles*.

Brunello, A., Montanari, A., & Reynolds, M. (2019). Synthesis of LTL formulas from natural language texts: State of the art and research directions. *147*, In *26th international symposium on temporal representation and reasoning* (pp. 17:1–17:19).

Censi, A., Slutsky, K., Wongpiromsarn, T., Yershov, D., Pendleton, S., Fu, J., et al. (2019). Liability, ethics, and culture-aware behavior specification using rulebooks. In *International conference on robotics and automation* (pp. 8536–8542).

Cho, K., Ha, T., Lee, G.-M., & Oh, S. (2019). Deep predictive autonomous driving using multi-agent joint trajectory prediction and traffic rules. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2076–2081.

Chou, Y., Yoon, H., & Sankaranarayanan, S. (2020). Predictive runtime monitoring of vehicle models using Bayesian estimation and reachability analysis. In *Proc. of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 2111–2118).

Clarke, E. M., Grumberg, O., & Peled, D. A. (1999). *Model checking*. MIT Press.

Corso, A., & Kochenderfer, M. J. (2020). Interpretable safety validation for autonomous vehicles. In *Proc. of the 23rd IEEE international conference on intelligent transportation systems* (pp. 1–6).

Dahl, J., de Campos, G. R., Olsson, C., & Fredriksson, J. (2019). Collision avoidance: A literature review on threat-assessment techniques. *IEEE Transactions on Intelligent Vehicles*, *4*(1), 101–113.

Dai, S., & Koutsoukos, X. (2016). Safety analysis of automotive control systems using multi-modal port-Hamiltonian systems. In *Proc. of the 19th international conference on hybrid systems: Computation and control* (pp. 105–114).

Danielson, C., Berntorp, K., Weiss, A., & Cairano, S. D. (2020). Robust motion planning for uncertain systems with disturbances using the invariant-set motion planner. *IEEE Transactions on Automatic Control*, *65*(10), 4456–4463.

DeCastro, J., Liebenwein, L., Vasile, C. -I., Tedrake, R., Karaman, S., & Rus, D. (2020). Counterexample-guided safety contracts for autonomous driving. In *Algorithmic foundations of robotics XIII* (pp. 939–955). Springer.

Dokhanchi, A., Amor, H. B., Deshmukh, J. V., & Fainekos, G. (2018). Evaluating perception systems for autonomous vehicles using quality temporal logic. *11237*, In *Runtime verification* (pp. 409–416). Springer.

Dolginova, E., & Lynch, N. (1997). Safety verification for automated platoon maneuvers: A case study. In *Proc. of the international workshop on hybrid and real-time systems* (pp. 154–170).

Du, P., Huang, Z., Liu, T., Ji, T., Xu, K., Gao, Q., et al. (2020). Online monitoring for safe pedestrian-vehicle interactions. In *Proc. of the 23rd IEEE international conference on intelligent transportation systems* (pp. 1–8).

Ersal, T., Kolmanovsky, I., Masoud, N., Ozay, N., Scruggs, J., Vasudevan, R., et al. (2020). Connected and automated road vehicles: State of the art and future challenges. *Vehicle System Dynamics*, *58*(5), 672–704.

Esterle, K., Aravantinos, V., & Knoll, A. (2019). From specifications to behavior: Maneuver verification in a semantic state space. In *Proc. of the IEEE intelligent vehicles symposium* (pp. 2140–2147).

Esterle, K., Gressenbuch, L., & Knoll, A. (2020). Formalizing traffic rules for machine interpretability. In *Proc. of the 3rd IEEE connected and automated vehicles symposium* (pp. 1–7).

Falcone, P., Ali, M., & Sjöberg, J. (2011). Predictive threat assessment via reachability analysis and set invariance theory. *IEEE Transactions on Intelligent Transportation Systems*, *12*(4), 1352–1361.

Fan, C. (2019). *Formal methods for safe autonomy: data-driven verification, synthesis, and applications* (Ph.D. thesis), University of Illinois at Urbana-Champaign, URL http://hdl.handle.net/2142/106202.

Fan, C., Qi, B., & Mitra, S. (2018). Data-driven formal reasoning and their applications in safety analysis of vehicle autonomy features. *IEEE Design & Test*, *35*(3), 31–38.

Fraichard, T., & Asama, H. (2004). Inevitable collision states—A step towards safer robots? *Advanced Robotics*, *18*(10), 1001–1024.

Gerdts, M., & Xausa, I. (2013). Avoidance trajectories using reachable sets and parametric sensitivity analysis. In *System modeling and optimization* (pp. 491–500).

Guanetti, J., Kim, Y., & Borrelli, F. (2018). Control of connected and automated vehicles: State of the art and future challenges. *Annual Reviews in Control*, [ISSN: 1367-5788] *45*, 18–40.

Hekmatnejad, M., Yaghoubi, S., Dokhanchi, A., Amor, H. B., Shrivastava, A., Karam, L., et al. (2019). Encoding and monitoring responsibility sensitive safety rules for automated vehicles in signal temporal logic. In *Proc. of the 17th ACM-IEEE international conference on formal methods and models for system design* (pp. 1–11). Association for Computing Machinery.

Helou, B., Dusi, A., Collin, A., Mehdipour, N., Chen, Z., Lizarazo, C., et al. (2021). The reasonable crowd: Towards evidence-based and interpretable models of driving behavior. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 6708–6715).

Herbert, S. L., Chen, M., Han, S., Bansal, S., Fisac, J. F., & Tomlin, C. J. (2017). FaSTrack: A modular framework for fast and guaranteed safe motion planning. In *Proc. of the 56th IEEE conference on decision and control* (pp. 1517–1522).

Hilscher, M., Linker, S., & Olderog, E.-R.

Hoehener, D., Huang, G., & Del Vecchio, D. (2016). Design of a lane departure driver-assist system under safety specifications. In *Proc. of the IEEE 55th conference on decision and control* (pp. 2468–2474).

de Iaco, R., Smith, S. L., & Czarnecki, K. (2020). Safe swerve maneuvers for autonomous driving. In *Proc. of the IEEE intelligent vehicles symposium* (pp. 1941–1948).

Jalalmaab, M., Fidan, B., Jeon, S., & Falcone, P. (2017). Guaranteeing persistent feasibility of model predictive motion planning for autonomous vehicles. In *Proc. of the IEEE intelligent vehicles symposium* (pp. 843–848).

Jha, S., Raman, V., Sadigh, D., & Seshia, S. (2018). Safe autonomy under perception uncertainty using chance-constrained temporal logic. *Journal of Automated Reasoning*, *60*, 43–62.

Jula, H., Kosmatopoulos, E. B., & Ioannou, P. A. (2000). Collision avoidance analysis for lane changing and merging. *IEEE Transactions on Vehicular Technology*, *49*, 2295–2308.

Kane, A., Chowdhury, O., Datta, A., & Koopman, P. (2015). A case study on runtime monitoring of an autonomous research vehicle (ARV) system. In *Runtime verification* (pp. 102–117). Springer.

Karaman, S., Walter, M. R., Perez, A., Frazzoli, E., & Teller, S. (2011). Anytime motion planning using the RRT. In *IEEE international conference on robotics and automation* (pp. 1478–1483).

Karimi, A., & Duggirala, P. S. (2020). Formalizing traffic rules for uncontrolled intersections. In *ACM/IEEE 11th international conference on cyber-physical systems* (pp. 41–50).

Kianfar, R., Falcone, P., & Fredriksson, J. (2013). Safety verification of automated driving systems. *IEEE Intelligent Transportation Systems Magazine*, *5*, 73–86.

Kojchev, S., Klintberg, E., & Fredriksson, J. (2020). A safety monitoring concept for fully automated driving. In *Proc. of the 23rd IEEE international conference on intelligent transportation systems* (pp. 1–7).

Kong, S., Gao, S., Chen, W., & Clarke, E. (2015). dReach: δ-reachability analysis for hybrid systems. In *International conference on TOOLS and algorithms for the construction and analysis of systems* (pp. 200–205).

Koschi, M., & Althoff, M. (2017). SPOT: A tool for set-based prediction of traffic participants. In *Proc. of the IEEE intelligent vehicles symposium* (pp. 1686–1693).

Koschi, M., & Althoff, M. (2017). Interaction-aware occupancy prediction of road vehicles. In *IEEE 20th international conference on intelligent transportation systems* (pp. 1–8).

Koschi, M., & Althoff, M. (2021). Set-based prediction of traffic participants considering occlusions and traffic rules. *IEEE Transactions on Intelligent Vehicles*, *6*(2), 249–265.

Koschi, M., Pek, C., & Althoff, M. (2018). Set-based prediction of pedestrians in urban environments considering formalized traffic rules. In *Proc. of the 21st IEEE international conference on intelligent transportation systems* (pp. 2704–2711).

Koschi, M., Söntges, S., & Althoff, M. (2018). Worst-case analysis of the time-to-react using reachable sets. In *Proc. of the IEEE intelligent vehicles symposium* (pp. 1891–1897).

Kousik, S., Vaskov, S., Johnson-Roberson, M., & Vasudevan, R. (2017). Safe trajectory synthesis for autonomous driving in unforeseen environments. *58271*, In *Dynamic systems and control conference*. American Society of Mechanical Engineers.

Koymans, R. (1990). Specifying real-time properties with metric temporal logic. *Real-Time Systems*, *2*(4), 255–299.

Kupferman, O., & Vardi, M. Y. (2001). Model checking of safety properties. *Formal Methods in System Design*, *19*(3), 291–314.

LaValle, S. M., & Kuffner, J. J., Jr. (2001). Randomized kinodynamic planning. *International Journal of Robotics Research*, *20*(5), 378–400.

Lawitzky, A., Nicklas, A., Wollherr, D., & Buss, M. (2014). Determining states of inevitable collision using reachability analysis. In *Proc. of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 4142–4147).

Li, X., Rosman, G., Gilitschenski, I., DeCastro, J., Vasile, C. -I., Karaman, S., et al. (2021). Differentiable logic layer for rule guided trajectory prediction. In *Proceedings of Machine Learning Research*: *155*, *Proc. of the 2020 conference on robot learning* (pp. 2178–2194).

Li, X., Rosman, G., Gilitschenski, I., Vasile, C. -I., DeCastro, J. A., Karaman, S., et al. (2021). Vehicle trajectory prediction using generative adversarial network with temporal logic syntax tree features. *IEEE Robotics and Automation Letters*, *6*(2), 3459–3466.

Liebenwein, L., Schwarting, W., Vasile, C. -I., DeCastro, J., Alonso-Mora, J., Karaman, S., et al. (2020). Compositional and contract-based verification for autonomous driving on road networks. In *Robotics research* (pp. 163–181). Springer International Publishing.

Lignos, C., Raman, V., Finucane, C., Marcus, M., & Kress-Gazit, H. (2014). Provably correct reactive control from natural language. *Autonomous Robots*, *38*, 89–105.

Ligthart, J., Semsar-Kazerooni, E., Ploeg, J., Alirezaei, M., & Nijmeijer, H. (2018). Controller design for cooperative driving with guaranteed safe behavior. In *Proc. of the IEEE conference on control technology and applications* (pp. 1460–1465).

Lin, Q., Chen, X., Khurana, A., & Dolan, J. M. (2020). ReachFlow: An online safety assurance framework for waypoint-following of self-driving cars. In *Proc. of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 6627–6632).

Linker, S., & Hilscher, M. (2013). Proof theory of a multi-lane spatial logic. In *Proc. of international colloquium on theoretical aspects of computing* (pp. 231–248).

Loos, S. M., Platzer, A., & Nistor, L. (2011). Adaptive cruise control: Hybrid, distributed, and now formally verified. In *LNCS 6664*, *Proc. of the 17th international symposium on formal methods* (pp. 42–56). Springer.

Loos, S. M., Witmer, D., Steenkiste, P., & Platzer, A. (2013). Efficiency analysis of formally verified adaptive cruise controllers. In *Proc. of the 16th international IEEE conference on intelligent transportation systems* (pp. 1565–1570).

Luckcuck, M., Farrell, M., Dennis, L. A., Dixon, C., & Fisher, M. (2019). Formal specification and verification of autonomous robotic systems: A survey. *ACM Computing Surveys*, *52*(5), 1–41.

Lygeros, J., Godbole, D. N., & Sastry, S. (1996). A verified hybrid controller for automated vehicles. In *Proc. of the 35th IEEE conference on decision and control* (pp. 2289–2294).

Lygeros, J., Godbole, D. N., & Sastry, S. (1998). Verified hybrid controllers for automated vehicles. *IEEE Transactions on Automatic Control*, *43*(4), 522–539.

Macek, K., Vasquez, D., Fraichard, T., & Siegwart, R. (2009). Towards safe vehicle navigation in dynamic urban scenarios. *Automatika*, *50*(3–4), 184–194.

Magdici, S., & Althoff, M. (2016). Fail-safe motion planning of autonomous vehicles. In *Proc. of the 19th international IEEE conference on intelligent transportation systems* (pp. 452–458).

Magdici, S., & Althoff, M. (2017). Adaptive cruise control with safety guarantees for autonomous vehicles. In *Proc. of the 20th world congress of the international federation of automatic control* (pp. 5774–5781).

Maierhofer, S., Rettinger, A.-K., Mayer, E., & Althoff, M. (2020). Formalization of Interstate Traffic Rules in Temporal Logic. In *Proc. of the IEEE Intelligent Vehicles Symposium* (pp. 752–759).

Maler, O., & Nickovic, D. (2004). Monitoring temporal properties of continuous signals. In *Proc. of international conference on FORMATS-FTRTFT* (pp. 152–166).

Mehra, A., Ma, W., Berg, F., Tabuada, P., Grizzle, J. W., & Ames, A. D. (2015). Adaptive cruise control: Experimental validation of advanced controllers on scale-model cars. In *Proc. of the American control conference* (pp. 1411–1418).

Memon, M. A. (2003). *Computational logic: Linear-time vs. Branching-time logics*. Graduate Course, Simon Fraser University, Burnaby, Canada.

Mirchevska, B., Pek, C., Werling, M., Althoff, M., & Boedecker, J. (2018). High-level decision making for safe and reasonable autonomous lane-changing with reinforcement learning. In *Proc. of the 21st IEEE international conference on intelligent transportation systems* (pp. 2156–2162).

Mitra, S. (2021). *Verifying cyber-physical systems: A path to safe autonomy*. MIT Press.

Nager, Y., Censi, A., & Frazzoli, E. (2019). What lies in the shadows? Safe and computation-aware motion planning for autonomous vehicles using intent-aware dynamic shadow regions. In *Proc. of the international conference on robotics and automation* (pp. 5800–5806).

Naumann, M., Konigshof, H., Lauer, M., & Stiller, C. (2019). Safe but not overcautious motion planning under occlusions and limited sensor range. In *Proc. of the IEEE intelligent vehicles symposium* (pp. 140–145).

Naumann, M., Königshof, H., & Stiller, C. (2019). Provably safe and smooth lane changes in mixed traffic. In *IEEE intelligent transportation systems conference* (pp. 1832–1837).

Neel, G., & Saripalli, S. (2020). Improving bounds on occluded vehicle states for use in safe motion planning. In *Proc. of the IEEE international symposium on safety, security, and rescue robotics* (pp. 268–275).

Nikora, A. P., & Balcom, G. (2009). Automated identification of LTL patterns in natural language requirements. In *20th international symposium on software reliability engineering* (pp. 185–194).

Nilsson, P., Hussien, O., Balkan, A., Chen, Y., Ames, A. D., Grizzle, J. W., et al. (2016). Correct-by-construction adaptive cruise control: Two approaches. *IEEE Transactions on Control Systems Technology*, *24*(4), 1294–1307.

Ody, H. (2017). Monitoring of traffic manoeuvres with imprecise information. In *First workshop on formal verification of autonomous vehicles* (pp. 43–58).

O'Kelly, M., Abbas, H., & Mangharam, R. (2017). Computer-aided design for safe autonomous vehicles. In *Proc. of resilience week* (pp. 90–96).

Orzechowski, P. F., Li, K., & Lauer, M. (2019). Towards responsibility-sensitive safety of automated vehicles with reachable set analysis. In *Proc. of the IEEE international conference on connected vehicles and expo* (pp. 1–6).

Orzechowski, P. F., Meyer, A., & Lauer, M. (2018). Tackling occlusions limited sensor range with set-based safety verification. In *21st international conference on intelligent transportation systems* (pp. 1729–1736).

Park, J., & Özgüner, Ü. (2012). Model Based Controller Synthesis using Reachability Analysis that Guarantees the Safety of Autonomous Vehicles in a Convoy. In *Proc. of the IEEE International Conference on Vehicular Electronics and Safety* (pp. 134–139).

Parthasarathi, R., & Fraichard, T. (2007). An inevitable collision state-checker for a car-like vehicle. In *Proc. of the IEEE international conference on robotics and automation* (pp. 3068–3073).

Pek, C., & Althoff, M. (2018). Efficient computation of invariably safe states for motion planning of self-driving vehicles. In *Proc. of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 3523–3530).

Pek, C., Koschi, M., Werling, M., & Althoff, M. (2017). Enhancing motion safety by identifying safety-critical passageways. In *Proc. of the 56th IEEE conference on decision and control* (pp. 320–326).

Pek, C., Manzinger, S., Koschi, M., & Althoff, M. (2020). Using online verification to prevent autonomous vehicles from causing accidents. *Nature Machine Intelligence*, *2*, 518–528.

Pek, C., Zahn, P., & Althoff, M. (2017). Verifying the safety of lane change maneuvers of self-driving vehicles based on formalized traffic rules. In *Proc. of the IEEE intelligent vehicles symposium* (pp. 1477–1483).

Phan-Minh, T., Cai, K. X., & Murray, R. M. (2019). Towards assume-guarantee profiles for autonomous vehicles. In *Proc. of the 58th IEEE conference on decision and control* (pp. 2788–2795).

Plaku, E., & Karaman, S. (2016). Motion planning with temporal-logic specifications: Progress and challenges. *AI Communications*, *29*(1), 151–162.

Rajabli, N., Flammini, F., Nardone, R., & Vittorini, V. (2021). Software verification and validation of safe autonomous cars: A systematic literature review. *IEEE Access*, *9*, 4797–4819.

Reyes Castro, L. I., Chaudhari, P., Tumová, J., Karaman, S., Frazzoli, E., & Rus, D. (2013). Incremental sampling-based algorithm for minimum-violation motion planning. In *Proc. of the 52nd IEEE conference on decision and control* (pp. 3217–3224).

Riedmaier, S., Ponn, T., Ludwig, D., Schick, B., & Diermeyer, F. (2020). Survey on scenario-based safety assessment of automated vehicles. *IEEE Access*, *8*, 87456–87477.

Rizaldi, A., & Althoff, M. (2015). Formalising traffic rules for accountability of autonomous vehicles. In *Proc. of the 18th IEEE international conference on intelligent transportation systems* (pp. 1658–1665).

Rizaldi, A., Immler, F., & Althoff, M. (2016). A formally verified checker of the safe distance traffic rules for autonomous vehicles. In *8th NASA formal methods symposium* (pp. 175–190).

Rizaldi, A., Immler, F., Schürmann, B., & Althoff, M. (2018). A formally verified motion planner for autonomous vehicles. In *Proc. of the international symposium on automated technology for verification and analysis* (pp. 75–90).

Rizaldi, A., Keinholz, J., Huber, M., Feldle, J., Immler, F., Althoff, M., et al. (2017). Formalising and -onitoring traffic rules for autonomous vehicles in Isabelle/HOL. In *Proc. of the 13th international conference on integrated formal methods* (pp. 50–66).

Rodionova, A., Alvarez, I., Elli, M. S., Oboril, F., Quast, J., & Mangharam, R. (2020). How safe is safe enough? Automatic safety constraints boundary estimation for decision-making in automated vehicles. In *Proc. of the IEEE intelligent vehicles symposium* (pp. 1457–1464).

Roehm, H., Oehlerking, J., Woehrle, M., & Althoff, M. (2019). Model conformance for cyber-physical systems: A survey. *Association for Computing Machinery*, [ISSN: 2378-962X] *3*(3).

Rong, J., & Luan, N. (2020). Safe reinforcement learning with policy-guided planning for autonomous driving. In *Proc. of the IEEE international conference on mechatronics and automation* (pp. 320–326).

Sadigh, D., Dragan, A., Sastry, S., & Seshia, S. (2017). Active preference-based learning of reward functions. In *Robotics: Science and systems*.

Sadraddini, S., Sivaranjani, S., Gupta, V., & Belta, C. (2017). Provably safe cruise control of vehicular platoons. *IEEE Control Systems Letters*, *1*(2), 262–267.

Sahin, Y. E., Quirynen, R., & Cairano, S. D. (2020). Autonomous vehicle decision-making and monitoring based on signal temporal logic and mixed-integer programming. In *Proc. of the American control conference* (pp. 454–459).

Schäfer, L., Manzinger, S., & Althoff, M. (2021). Computation of solution spaces for optimization-based trajectory planning. *IEEE Transactions on Intelligent Vehicles*, 1–1 (early access).

Schmidt, C., Oechsle, F., & Branz, W. (2006). Research on trajectory planning in emergency situations with multiple objects. In *Proc. of the IEEE intelligent transportation systems conference* (pp. 988–992).

Schürmann, B., Eilbrecht, D. H. J., Stursberg, O., Köster, F., & Althoff, M. (2017). Ensuring drivability of planned motions using formal methods. In *Proc. of the 20th IEEE international conference on intelligent transportation systems* (pp. 1–8).

Schwarting, W., Alonso-Mora, J., & Rus, D. (2018). Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, *1*(1), 187–210.

Seshia, S. A., Sadigh, D., & Sastry, S. S. (2015). Formal methods for semi-autonomous driving. In *52nd ACM/EDAC/IEEE design automation conference* (pp. 1–5).

Shalev-Shwartz, S., Shammah, S., & Shashua, A. (2017). On a formal model of safe and scalable self-driving cars. [Online]. Available: arXiv:1708.06374,

Shao, Y. S., Chen, C., Kousik, S., & Vasudevan, R. (2021). Reachability-based trajectory safeguard (RTS): A safe and fast reinforcement learning safety layer for continuous control. *IEEE Robotics and Automation Letters*, *6*(2), 3663–3670.

Smith, S. W., Nilsson, P., & Ozay, N. (2016). Interdependence quantification for compositional control synthesis with an application in vehicle safety systems. In *Proc. of the 55th IEEE conference on decision and control* (pp. 5700–5707).

Soloperto, R., Köhler, J., Allgöwer, F., & Müller, M. A. (2019). Collision avoidance for uncertain nonlinear systems with moving obstacles using robust model predictive control. In *Proc. of the 18th European control conference* (pp. 811–817).

Söntges, S., & Althoff, M. (2015). Determining the nonexistence of evasive trajectories for collision avoidance systems. In *Proc. of the 18th IEEE international conference on intelligent transportation systems* (pp. 956–961).

Söntges, S., & Althoff, M. (2018). Computing the drivable area of autonomous road vehicles in dynamic road scenes. *IEEE Transactions on Intelligent Transportation Systems*, *19*(6), 1855–1866.

Stahl, T., & Diermeyer, F. (2021). Online verification enabling approval of driving functions—Implementation for a planner of an autonomous race vehicle. *IEEE Open Journal of Intelligent Transportation Systems*, *2*, 97–110.

Stursberg, O. A. N. D., Fehnker, A. A. N. D., Han, Z. A. N. D., & Krogh, B. H. (2004). Verification of a cruise control system using counterexample-guided search. *Control Engineering Practice*, *12*(10), 1269–1278.

Tabuada, P. (2009). *Verification and control of hybrid systems - a symbolic approach*. Springer.

Tuncali, C. E., Fainekos, G., Ito, H., & Kapinski, J. (2018). Simulation-based adversarial test generation for autonomous vehicles with machine learning components. In *Proc. of the IEEE intelligent vehicles symposium* (pp. 1555–1562).

Tuncali, C. E., Fainekos, G., Prokhorov, D., Ito, H., & Kapinski, J. (2020). Requirements-driven test generation for autonomous vehicles with machine learning components. *IEEE Transactions on Intelligent Vehicles*, *5*(2), 265–280.

Vanholme, B., Gruyer, D., Lusetti, B., Glaser, S., & Mammar, S. (2013). Highly automated driving on highways based on legal safety. *IEEE Transactions on Intelligent Transportation Systems*, *14*(1), 333–347.

Vasile, C. -I., Tumova, J., Karaman, S., Belta, C., & Rus, D. (2017). Minimum-violation scLTL motion planning for mobility-on-demand. In *IEEE international conference on robotics and automation* (pp. 1481–1488).

Vaskov, S., Kousik, S., Larson, H., Bu, F., Ward, J. R., Worrall, S., et al. (2019). Towards Provably Not-at-Fault Control of Autonomous Robots in Arbitrary Dynamic Environments. In *Proc. of Robotics: Science and Systems*.

Völker, M., Kloock, M., Rabanus, L., Alrifaee, B., & Kowalewski, S. (2019). Verification of cooperative vehicle behavior using temporal logic. *IFAC-PapersOnLine*, *52*(8), 99–104.

Wang, Q., Li, D., & Sifakis, J. (2020). Safe and efficient collision avoidance control for autonomous vehicles. In *Proc. of the 18th ACM-IEEE international conference on formal methods and models for system design* (pp. 1–6).

Wongpiromsarn, T., Mitra, S., Murray, R., & Lamperski, A. (2012). Verification of periodically controlled hybrid systems: Application to an autonomous vehicle. *ACM Transactions on Embedded Computing Systems*, *11*(S2), 1–24.

Wongpiromsarn, T., Slutsky, K., Frazzoli, E., & Topcu, U. (2021). Minimum-violation planning for autonomous systems: Theoretical and practical considerations. In *Proc. of the American control conference* (pp. 4866–4872).

Wu, A., & How, J. P. (2012). Guaranteed infinite horizon avoidance of unpredictable, dynamically constrained obstacles. *Autonomous Robots*, *32*(3), 227–242.

Xiao, W., Mehdipour, N., Collin, A., Bin-Nun, A. Y., Frazzoli, E., Tebbens, R. D., et al. (2021). Rule-based optimal control for autonomous driving. In *Proc. of the ACM/IEEE 12th international conference on cyber-physical systems* (pp. 143–154).

Xu, X., Grizzle, J. W., Tabuada, P., & Ames, A. D. (2018). Correctness guarantees for the composition of lane keeping and adaptive cruise control. *IEEE Transactions on Automation Science and Engineering*, *15*(3), 1216–1229.

Yurtsever, E., Lambert, J., Carballo, A., & Takeda, K. (2020). A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, *8*, 58443–58469.

Noushin Mehdipour received a Ph.D. in Systems Engineering from Boston University, Massachusetts, USA in 2022. Her research interests include formal methods, control, machine learning and optimization. At Motional, Noushin worked on formal specification of safe and lawful autonomous driving using Rulebooks. She is currently working in the Motion Planning Research team at Motional where she is focused on evaluation of AVs driving behavior and contributing to the nuPlan competition, the world's first large-scale planning benchmark for autonomous driving.

Matthias Althoff is an associate professor in Computer Science at the Technical University of Munich, Germany. He received his diploma engineering degree in Mechanical Engineering in 2005, and his Ph.D. degree in Electrical Engineering in 2010, both from the Technical University of Munich, Germany. From 2010 to 2012 he was a postdoctoral researcher at Carnegie Mellon University, Pittsburgh, USA, and from 2012 to 2013 an assistant professor at Ilmenau University of Technology, Germany. His research interests include the formal verification of continuous and hybrid systems, reachability analysis, planning algorithms, nonlinear control, automated vehicles, and power systems.

Radboud Duintjer Tebbens professional interest is in mathematically modeling high-impact decisions that involve both non-linear dynamics and statistical uncertainty. His acclaimed work with Kid Risk, Inc. helped the Global Polio Eradication Initiative get closer to a world free from wild polioviruses. At nuTonomy, Aptiv, and Motional, Radboud helped develop frameworks for the statistical validation of the safety of fully autonomous vehicles and for formally specifying safe, lawful, and natural autonomous driving using Rulebooks. Radboud holds a Ph.D. in Applied Mathematics from Delft University of Technology in the Netherlands, conducted post-doctoral research at Harvard and MIT, and taught as an Assistant Professor of Risk and Decision Analysis at Delft University of Technology. He currently is a Senior Staff Safety Data Scientist at Aurora.

Calin Belta is a Professor of Mechanical Engineering, Electrical and Computer Engineering, and Systems Engineering at Boston University, where he holds the Tegan family Distinguished Faculty Fellowship. He is the Director of the BU Robotics Lab. His research focuses on dynamics and control theory, with particular emphasis on cyber–physical systems, formal methods, and applications to robotics and systems biology. Notable awards include the 2008 AFOSR Young Investigator Award, the 2005 National Science Foundation CAREER Award, and the 2017 IEEE TCNS Outstanding Paper Award. He is a Fellow of the IEEE and a Distinguished Lecturer of the IEEE CSS.