


Reachability Analysis Plus Satisfiability Modulo Theories: An Adversary-Proof Control Method for Connected and Autonomous Vehicles

Qing Xu, Yicong Liu , Jian Pan, Jiawei Wang , *Graduate Student Member, IEEE*, Jianqiang Wang , and Keqiang Li 

Abstract—Connected and autonomous vehicles (CAVs) are expected to operate with safety guarantee in presence of adversaries from the Internet of Vehicles. This article proposes a control method named reachability analysis plus satisfiability modulo theories (RA-SMT) for CAVs against integrity attacks caused by bounded adversary. This method enables vehicles to possess the reach-avoid specification and strict control safety ensurance even in the worst case scenario. The introduction of state-feedback control decomposes the original complex problem into more manageable reachability analysis and adversary-free control strategy optimization. Precisely, zonotope sets are employed for reachability analysis, and the control strategy is optimally solved and verified simultaneously via SMT. This method is applicable to complex traffic scenarios with the help of SMT, which can describe various constraints flexibly and conveniently. Simulation results reveal the effectiveness and safety of the proposed method in the classical car-following scenario against bounded adversary under various conditions. Particularly, RA-SMT exhibits significantly improved control performance (around 16%) and computation efficiency (around 63%) compared with existing methods. Finally, the RA-SMT is implemented on an autonomous driving platform to validate its practicability.

Index Terms—Bounded adversary, connected and autonomous vehicle (CAV) controller synthesis, reachability analysis, satisfiability modulo theories (SMT).

I. INTRODUCTION

THE emergence of connected and autonomous vehicles (CAVs) promises sharing of sensor networks,

computation resources, and control platforms with the development of the Internet of Vehicles (IoV) technology, e.g., 5G-V2X [1], [2], contributing to better road transportation systems in the near future. On the other hand, the IoV environment poses crucial challenges to the system cybersecurity, which is widely recognized as a critical issue for networked control systems (NCSs) [3]. Precisely, the CAVs are likely to operate in the presence of adversaries and attacks, including attacks against physical objects, denial-of-service attacks, and integrity attacks [4]. In particular, the integrity attacks are among the most addressed cyberattacks in NCS, which refers to the cyberattacks orchestrated by modifying the information contained in the transmitted data packets and compromising the system integrity. Compared to other types of attacks, the integrity attacks can cause more severe consequences because of the intentional modification of the transmitted data [5]. To realize the advantages of the networking environment in transportation and CAVs, it is significant to develop methods of ensuring vehicle safety under integrity attacks. Insecurities and vulnerabilities can be widely present in CAVs, such as in-vehicle systems (e.g., sensors and bus networks), V2X communications, and intelligent and autonomous algorithms corresponding to the perception, cognition, and execution layers [6]. This article falls within the scope of designing trajectory planning and tracking control methods for CAVs in the execution layer to address the specific integrity attacks in V2X communication.

In recent years, the security issues regarding IoV have garnered researchers' attention. Several security methods have been proposed to mitigate the risk of integrity attacks, which can be used for reference to improve the security level for CAVs. One natural idea is to develop more sophisticated mechanisms to detect and block such attacks [7], [8]. However, the attacks may not be identified in the data layer, and thus, can subtly deteriorate the system performance [5], [9]. Learning-based techniques offer alternative options for solving these security problems. The adversarial reinforcement learning (ARL) provides a cutting-edge framework, which has been used to build a learning framework of CAVs that includes specific scenarios, adversarial information, and system outputs as a whole. Behzadan and Munir [10] established the framework of ARL for the collision avoidance mechanism of CAVs with adversary. In [11], ARL was used to improve the control robustness and

Manuscript received 16 November 2021; revised 23 February 2022; accepted 22 March 2022. Date of publication 12 April 2022; date of current version 16 November 2022. This work was supported by the National Key Research and Development Program of China under Grant 2019YFB1600804, in part by the National Natural Science Foundation of China under Grant 52072212, in part by Tsinghua-Toyota Joint Research Institute Cross-discipline Program, and in part by Dongfeng Automobile Company, Ltd. (Corresponding author: Yicong Liu.)

The authors are with the School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China (e-mail: qingxu@tsinghua.edu.cn; liuyicon20@mails.tsinghua.edu.cn; pja17@mails.tsinghua.edu.cn; wang-jw18@mails.tsinghua.edu.cn; wjqlws@tsinghua.edu.cn; likq@tsinghua.edu.cn).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TIE.2022.3165293>.

Digital Object Identifier 10.1109/TIE.2022.3165293

safety of CAVs in lane-changing scenarios. The learning-based methods have advantages of simplifying the CAV control system analysis and perhaps better adapting to a variety of attacks, with the disadvantages of scenario-based limitations and lack of strict guarantee of control safety.

From the perspective of controller synthesis, there is a considerable amount of the literature focused on optimal control methods of adapting to adversaries. These studies aim to ensure control safety, although the system is attacked inevitably. Researchers attempted to model the controller and the adversary as two sides of a differential game with the reachability analysis, and have applied it to robot-tracking control recently [12]–[14]. This method is systematic and applicable to linear and nonlinear systems, but it suffers from the problem of “curse of dimensionality,” and the assumption of information dependency is required between the gaming sides [14]. Another approach involves combining reachability analysis with optimal control. Schürmann and Althoff [15] and Schürmann *et al.* [16] proposed formal methods for the control problem with disturbances that use the zonotope to approximate the reachable sets and realized the optimal control employing the model predictive control (MPC), which is similar to the tube-based robust control method for additional perturbation inputs [17], [18] or parametric uncertainties [19]. These studies have improved the computation efficiency and have been applied to the disturbed vehicle control or mobile robot control. However, the solving technique is not effective with complex constraints and probably resulting in the solution obtained being a local optimum.

To further improve the flexibility of solving control problems, certain researchers adopted the method of linear temporal logic (LTL) for controller synthesis [20]–[22]. Huang *et al.* [20] introduced a way of implementing the LTL method using satisfiability modulo theories (SMT), which refers to determining whether a first-order formula is satisfiable with respect to background theories [23]. In a subsequent study, a controller with adversary was synthesized for linear dynamical systems [21], which first proposed decoupling of the adversary from the system, and thereafter solving the adversary-free control strategy. Consequently, Fan *et al.* [22] combined the reachability analysis with a tracking controller, and realized the close-loop SMT control for disturbed linear systems. Besides the flexibility, SMT also exhibits its advantages in satisfying certain requirements with risky attacks. Shoukry *et al.* [24] proposed an algorithm exploiting SMT to securely estimate the system state with adversarially corrupted measurements, so that the controllers are not affected critically. Dutta *et al.* [25] presented a hybrid framework, which combines SMT and reinforcement learning to strike a better balance between the optimization of response planning with adversaries and satisfying all various requirements as constraints. Although applying the LTL or SMT to CAV control methods results in enhanced ability to handle complex constraints, the control performance with adversary still requires further advancement. As to CAVs application, safety-critical feature and certain control performance are required in various traffic scenarios. Methods, such as the reachability analysis, satisfiability theories, and optimal control, should be integrated to

TABLE I
ABBREVIATIONS USED THROUGHOUT THIS ARTICLE

Abbreviation	Meaning
ARL	adversarial reinforcement learning
CAVs	connected and autonomous vehicles
IoV	internet of vehicles
LTL	linear temporal logic
(r)MPC	(robust) model predictive control
NCS	networked control systems
RA-SMT	reachability analysis plus satisfiability modulo theories
RHC	receding horizon control
RSU	road side units
SMT	satisfiability modulo theories

synthesize the controller for CAVs against the potential integrity attacks.

This article focuses on designing a CAV control method under integrity attacks, with the aim of ensuring vehicle safety with arbitrary bounded adversary. Specifically, the contributions of this article are as follows:

- 1) For the potential integrity attacks in IoV, an adversary-proof control method named reachability analysis plus satisfiability modulo theories (RA-SMT) is proposed to enable CAVs to possess the reach-avoid specification, thereby aiming to ensure strict vehicle control safety.
- 2) Efficient reachability analysis with zonotope and optimal controller synthesis via SMT improve the control feasibility and optimality for CAVs with bounded adversaries, as well as adaptability to more complex traffic scenarios.
- 3) Compared with [26], the proposed CAV control method demonstrates significantly improved control performance and efficiency in the simulations, and the experiments validate its practicability in the car-following scenario.

The rest of this article is organized as follows. Section II lists the significance of various abbreviations throughout this article. The system modeling and problem statement are presented in Section III. Section IV presents the proposed CAV control method with bounded adversary. Section V shows the car-following simulation results with our autonomous driving platform. Finally, Section VI concludes this article.

II. LIST OF ABBREVIATIONS

The significance of various abbreviations used throughout this article is summarized and listed alphabetically in Table I.

III. DYNAMICAL MODELING AND PROBLEM STATEMENT

In this section, we introduce the dynamical modeling framework of CAV control under adversarial input, and then present the reach-avoid problem under investigation in this article.

A. CAV Control Model Under Adversarial Input

CAVs deployed in the IoV environment can be regarded as remote control systems, which share central computation

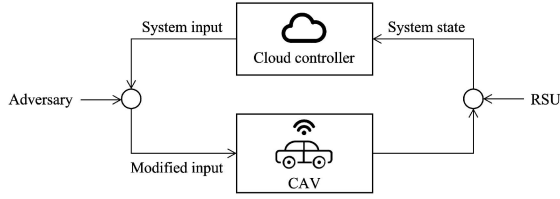


Fig. 1. Model of CAVs with the integrity attack.

resources and control platforms [27], [28]. The vehicles will be controlled by receiving wireless control signals through communication channels. The cloud controller observes the system state via the CAV and the road-side units, and calculates the system control input. Motivated by [21] and [29], the integrity attack is modeled as the modification or injection of the control signals in the communication channels by unknown adversary, see Fig. 1 for illustration.

Consider the following discrete-time linear time variant system with adversary:

$$x(k+1) = A(k)x(k) + B(k)u(k) + C(k)w(k) \quad (1)$$

where $x(k) \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ is the system state, $u(k) \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ is the system input, and $w(k) \in \mathcal{W} \subseteq \mathbb{R}^{n_w}$ is the adversarial input, with \mathcal{X} , \mathcal{U} , and \mathcal{W} denoting the corresponding constraint sets. The matrices $A(k)$, $B(k)$, and $C(k)$ represent the state, input, and adversarial matrices, respectively, which are allowed to vary over time step k within the horizon N , $0 \leq k \leq N$.

Furthermore, the adversarial input is assumed to be bounded. Note that the no control method can protect the CAV from being arbitrarily attacked by the unbounded adversary. Motivated by the existing research [21], [22], we assume that the adversarial input is norm bounded, given by

$$\|w(k)\|_\infty \leq \bar{w} \quad \forall 0 \leq k \leq N \quad (2)$$

with the upper bound denoted as \bar{w} . Note that this assumption does not require prior information regarding the adversarial input, except that its infinite norm bound is known. In addition, the proposed control method is also applicable to other norms given the fact that they can be transformed linearly.

B. Problem Statement

According to (1), the arbitrary adversary causes a change of the system state. Regarding the objectives of the CAV control, the vehicle controller must ensure vehicle safety with the adversary at every moment. Moreover, there exist specific control goals of the vehicle that must be achieved within a certain time, as shown in Fig. 2. The aim of the controller design coincides with the reach-avoid specification [13], [30], [31], implying that with which the controller is effective with the adversary.

Definition 1 (Reachable Set): Considering the control system described by (1), the reachable set of the vehicle state at any given time is defined as follows:

$$\mathcal{R}_{k,\mathcal{U},\mathcal{W}}(\mathcal{S}) = \{x(k) \in \mathbb{R}^{n_x} \mid \exists x(0) \in \mathcal{S}, u(\cdot) \in \mathcal{U}, w(\cdot) \in \mathcal{W} : x(k) = \xi(x(0), u(\cdot), w(\cdot), k)\} \quad (3)$$

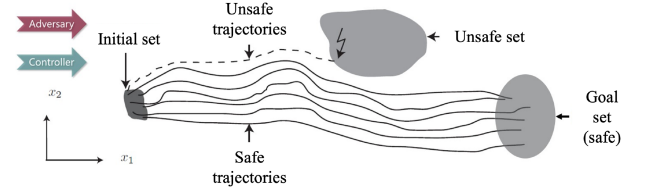


Fig. 2. Schematic diagram of the reach-avoid specification with adversary.

where $\mathcal{R}_{k,\mathcal{U},\mathcal{W}}(\mathcal{S})$ denotes the reachable set at time step k with the initial set \mathcal{S} , $u(\cdot)$ and $w(\cdot)$ represent the admissible control input and adversarial input at any time, respectively, and $\xi(\cdot)$ is the system state dynamical function, based on (1).

Definition 2 (Reach-Avoid Specification): At any particular time period $k \in [k_1, k_2] \subseteq [0, N]$, there exists a control strategy (input sequence) $u(\cdot) \in \mathcal{U}$, such that for all the possible adversarial strategies, i.e., $\forall w(\cdot) \in \mathcal{W}$, the reach-avoid specification is equivalent to the following conditions that should be satisfied:

$$\mathcal{R}_{[k_1, k_2], \mathcal{U}, \mathcal{W}}(\mathcal{S}) = \bigcup_{k \in [k_1, k_2]} \mathcal{R}_{k, \mathcal{U}, \mathcal{W}}(\mathcal{S}) \quad (4)$$

$$\mathcal{R}_{[k_1, k_2], \mathcal{U}, \mathcal{W}}(\mathcal{S}) \cap \mathcal{O}_{[k_1, k_2]} = \emptyset \quad (5)$$

$$\mathcal{R}_{k_2, \mathcal{U}, \mathcal{W}}(\mathcal{S}) \subseteq \mathcal{G} \quad (6)$$

where $\mathcal{O}_{[k_1, k_2]}$ denotes the union of all the unsafe sets within the time $k \in [k_1, k_2]$ as (4) and \mathcal{G} denotes the goal set, that is, the vehicle state should ultimately be in the set.

Based on the definitions, once the controller has the reach-avoid specification, it can ensure control safety and goal regardless of the arbitrary adversarial inputs. The key issue is how to achieve this with arbitrarily uncertain adversary.

IV. CAV CONTROLLER DESIGN WITH BOUNDED ADVERSARY

This section introduces the proposed control method with adversary, including the reachability analysis, optimal controller synthesis, as well as the specific implementation algorithm.

A. Reachability Analysis

We focus on the controller design at the worst case. Hence, the computation of reachable sets of the system state is indispensable. Precisely, the computation is based on state-feedback control, which narrows down the reachable sets and appears promising for controlling performance [32]. Accordingly, a linear state-feedback control law is utilized to conduct the computation recursively.

The following design is considered. First, a virtual reference trajectory is set. Subsequently, a tracking controller is designed to track the trajectory. Upon determining the tracking controller, the computation of the reachable sets is decomposed from the solving of the control inputs. Precisely, the reference trajectory and the control input are described as follows:

$$x_{\text{ref}}(k+1) = A(k)x_{\text{ref}}(k) + B(k)u_{\text{ref}}(k) \quad (7)$$

$$u(k) = u_{\text{ref}}(k) + K[x(k) - x_{\text{ref}}(k)] \quad (8)$$

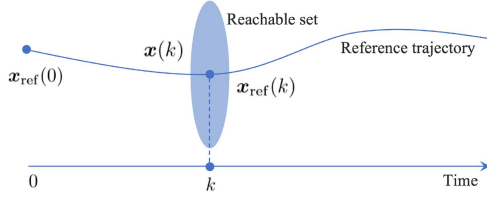


Fig. 3. Reachable set centered on the reference trajectory.

where $x_{\text{ref}}(k) \in \mathbb{R}^{n_x}$ is the state of the reference trajectory (reference state) at the time step k , $u_{\text{ref}}(k) \in \mathbb{R}^{n_u}$ is the control input for the reference trajectory, which enables all states of the reference trajectory to be iteratively calculated, and K is the linear-feedback matrix for the control law.

The tracking error can be calculated by combining (1), (7), and (8), and described by

$$x_d(k+1) = [A(k) + B(k)K] x_d(k) + Cw(k) \quad (9)$$

where $x_d(k) = x(k) - x_{\text{ref}}(k)$ represents the deviation between the actual state and the reference state. More intuitively, the center of the reachable set at any particular time is the corresponding reference state, see Fig. 3 for illustration. If the center point of the region is shifted to the origin, the region constitutes the *difference reachable set* (i.e., all the possible tracking errors) corresponding to that time.

It can be deduced that the difference reachable set is independent of the selection of the reference trajectory, thereby facilitating the subsequent computation and verification. Furthermore, with the same definition as (3), the difference reachable set at time step k is denoted by $\mathcal{R}_{k,\mathcal{U},\mathcal{W}}^d$.

Definition 3 (Zonotope Set): Zonotope is a manner of representing the centrally symmetric sets with one center vector and multiple generator vectors, which is defined by

$$\mathcal{Z} := \left\{ x \in \mathbb{R}^{n_x} \mid x = c_z + \sum_{i=1}^e \beta_i g^{(i)}, \beta_i \in [-1, 1] \right\} \quad (10)$$

where \mathcal{Z} is the zonotope set, $c_z \in \mathbb{R}^{n_x}$ is its center vector, $g^{(1)}, \dots, g^{(e)} \in \mathbb{R}^{n_x}$ are the generator vectors, and β_i is the variable to expand the generators. It can also be briefly denoted as $\mathcal{Z} := (c_z, \langle g^{(1)}, \dots, g^{(e)} \rangle)$.

The computation of the difference reachable set is achieved via zonotope. Employing the zonotope to represent the set is more efficient in computing the recursive set [33]. Zonotope is closed in linear transformation and Minkowski addition, according to its definition. Moreover, it is sufficiently flexible to convert to polytope in an exact and efficient manner.

The difference reachable set can be computed recursively and efficiently using the following formula:

$$\begin{aligned} \mathcal{R}_{k+1,\mathcal{U},\mathcal{W}}^d &= [(A(k) + B(k)K) \cdot \mathcal{R}_{k,\mathcal{U},\mathcal{W}}^d] \oplus (C(k) \cdot D_k) \\ &= (c_{k+1}, \langle g^{(1)}, \dots, g^{(e_{k+1})} \rangle) \end{aligned} \quad (11)$$

where $\mathcal{R}_{k,\mathcal{U},\mathcal{W}}^d$ denotes the difference reachable set at time step k , D_k denotes the set of the bounded adversarial inputs, which can be described by the zonotope set according to (2) and (10), the

operator \cdot represents the linear transformation, and the operator \oplus represents the Minkowski addition.

Till now, the difference reachable sets at any particular time can be determined provided the initial set is available. The introduction of the state-feedback control decouples the reachability analysis from further control strategy design. This provides the basis for the solving and verification of the adversary-proof control strategy.

Remark 1: The reachability analysis is also applicable to the control problem with multiple bounded adversarial inputs defined by (1) and (2). The difference reachable set can still be computed with zonotope by performing multiple Minkowski additions to the linearly transformed zonotope sets of the multiple bounded adversarial inputs.

B. Control Strategy Design

Following the reachability analysis, a new problem of synthesizing the adversary-free control strategy is to be converted to eliminate system uncertainty. Herein, the problem conversion is first illustrated by strengthening the constraints. Subsequently, a satisfiability problem is built, and SMT is used to solve and verify the optimal control strategy accordingly.

1) Constraints Strengthening: According to Fig. 3, the satisfiability problem that arises is that given the constraint and difference reachable sets, does there exist a sequence of feasible and solvable control inputs for the reference trajectory such that the system states can always be within the constraints? To better handle this problem, the uncertainty in the control system must be eliminated. Given the decoupled reachable sets in (7), a natural idea is to solve the control inputs for the reference trajectory with alternative constraints. This allows the problem to become determined and solvable. However, one crucial point is to calculate the corresponding constraints such that the problem conversion is equivalent.

The abovementioned equivalent conversion is embodied in the constraint and the difference reachable sets. The following relation should be satisfied to ensure the constraints are always satisfied:

$$\mathcal{X}_{\text{ref}}(k) \oplus \mathcal{R}_{k,\mathcal{U},\mathcal{W}}^d \subseteq \mathcal{X}_{\text{cstr}}(k) \quad \forall 0 \leq k \leq N \quad (12)$$

where k is the time step within the horizon N , \mathcal{X}_{ref} denotes the constraint sets for the reference trajectory to be computed, and $\mathcal{X}_{\text{cstr}}$ denotes the known constraint sets for the actual trajectory of the system state, which is also the complementary set of the unsafe set or the goal set in the reach-avoid specification.

In fact, the constraint sets for the reference trajectory should be as large as possible to render the control inputs for the reference trajectory more feasible and solvable. Thus, the constraint sets can be computed as expressed by

$$\mathcal{X}_{\text{ref}}(k) = \mathcal{X}_{\text{cstr}}(k) \oplus (-\mathcal{R}_{k,\mathcal{U},\mathcal{W}}^d). \quad (13)$$

Such computation can be implemented as mentioned above for zonotope, polytope, or a nonconvex union of polytopes, see detailed discussions in [34]. The equivalent problem for the reference trajectory is now determined; the iteration relation is deterministic as expressed by (7) and the states of the reference

trajectory should be within the constraint sets described by (13). Essentially, this is a satisfiability problem. SMT is adopted to build and solve the problem.

2) Optimal Control With SMT: Regarding the constraint sets for the reference trajectory presented as polytope, they can be formulated by the half-space description as $\mathcal{X}_{\text{ref}}(k) = \{x \in \mathbb{R}^{n_x} \mid H(k)x \leq b(k)\}$. At any particular time, the reference state should be in the corresponding constraint set, implying that (14) should hold. Linear inequalities can be conveniently transformed to formula in first-order logic, such that they form the inputs of the satisfiability problem, given by

$$\bigwedge_{1 \leq r \leq R_k} h_r^{(k)} x_{\text{ref}}(k) \leq b_r^{(k)} \quad (14)$$

where the matrix H and the vector b are the components for a half-space description of a polytope, R_k is the number of the half-space at time step k , $h_r^{(k)}$ and $b_r^{(k)}$ are the r th row vector and element of the matrix $H(k)$ and the vector $b(k)$, respectively, and the operator \wedge represents the logical AND in SMT. However, the actual variables are not $x_{\text{ref}}(k)$, but rather the sequential control inputs according to (7).

The constraint satisfiability problem at any particular time is subsequently formulated in SMT using

$$\Phi^{(k)}(x_{\text{ref}}(k)) := P_1^{(k)}(u_{\text{ref}}(\cdot)) \wedge \cdots \wedge P_{R_k}^{(k)}(u_{\text{ref}}(\cdot)) \quad (15)$$

$$P_r^{(k)}(u_{\text{ref}}(\cdot)) := \left(h_r^{(k)} x_{\text{ref}}(k) \leq b_r^{(k)} \right) \quad (16)$$

where $\Phi^{(k)}$ denotes the satisfiability problem at time step k , $P_r^{(k)}$ denotes the r th predicate of the problem, and the sequential inputs $u_{\text{ref}}(\cdot)$ are the actual variables, which refer to $u_{\text{ref}}(0), \dots, u_{\text{ref}}(k-1)$ in this case.

Furthermore, the complete satisfiability problem is for every time step. The reference state should always be in the constraint sets. Therefore, the complete satisfiability problem is presented as

$$\Phi := \bigwedge_{1 \leq k \leq N} \Phi^{(k)}(x_{\text{ref}}(k)) \quad (17)$$

where Φ denotes the complete satisfiability problem, from current time to ended time step N . The logical AND represents that all should be satisfied simultaneously.

We proceed to present the optimization problem to obtain the optimal control input for the reference trajectory within certain time horizon. The objective function is designed to minimize the deviation between the reference trajectory and the desired trajectory, implying that the vehicle should be controlled to the desired state constantly. The optimization problem is formally presented as follows:

$$\begin{aligned} \min_{u_{\text{ref}}(\cdot)} \quad & \mathcal{L}(u_{\text{ref}}(\cdot)) = \mathcal{D}(x_{\text{ref}}(\cdot), x_{\text{des}}(\cdot)) \\ \text{subject to} \quad & (7), (8), \text{ and } \Phi \text{ is satisfiable} \end{aligned} \quad (18)$$

where $u_{\text{ref}}(\cdot)$ denotes the sequential control inputs for the reference trajectory $x_{\text{ref}}(\cdot)$, $x_{\text{des}}(\cdot)$ is the desired trajectory to be tracked, \mathcal{L} is the objective function, and \mathcal{D} is the corresponding

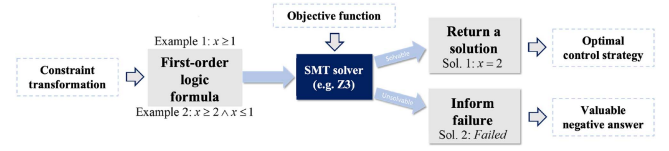


Fig. 4. Solving optimal control strategy with the SMT solver.

function representing the distance between the two trajectories, such as the 1-norm function.

3) Calculation and Verification: The solving and verification techniques are based on the SMT solver, such as Z3 [23]. The process is illustrated in Fig. 4. First, the satisfiability problem, as (17), is expressed by the first-order logic formula, which is the input of the SMT solver. Thereafter, the SMT solver either find a feasible solution to a set of constraints or to prove that no such solution exists by its theory solver and SAT reasoning. Consequently, once the solver outputs a solution, it must be feasible for the satisfiability problem, i.e., the solving and the verification are completed simultaneously. In the vehicle controller design, the control strategy should conform to constraints and also ensures control performance to a certain extent, which requires optimization in the solving process. Technically, optimization with SMT solver is viable. For linear objective functions, the vZ optimizer as an extended functionality of Z3 solver can be adopted conveniently [35]. For nonlinear objective functions and even nonconvex problems, a self-designed loop around the SMT solver can be written to find the global optimal solution [36].

Remark 2: When the constraint sets for the reference trajectory are more complex (e.g., nonconvex), the solving and verification method based on SMT can still be applied. Following the constraint sets being exactly described or approximated by finite multiple polytopes, the logical operators in SMT can express the relations among them. For example, the constraint sets composed of a union of polytopes can always be divided into finite mutually exclusive polytopes, the logical operator OR can be used among them flexibly.

C. Implementation Algorithm of CAVs

Based on discussion till now, Algorithm IV-C is proposed for solving the sequential control inputs of the CAV controller. The subroutine functions are summarized as follows:

- 1) *SetCompute*: It computes the difference reachable set $\mathcal{R}_{k,U,W}^d$, given the initial set.
- 2) *SetStrengthen*: It aims to eliminate the uncertainty in the control system caused by the adversary by strengthening the actual constraints $\mathcal{X}_{\text{cstr}}(k)$ to obtain the adversary-free constraints $\mathcal{X}_{\text{ref}}(k)$ for the reference trajectory.
- 3) *LogicFormulate*: It transforms the constraints into the first-order logic formulae, and constructs the satisfiability problem, which can be handled by SMT.
- 4) *LogicConnect*: It combines the satisfiability problem for every time step with operators in the first-order logic.
- 5) *SolveOptSMT*: It is used to solve and verify the optimal sequential control inputs for the reference trajectory $u_{\text{ref}}(\cdot)$ with the SMT solver and an objective function.

Algorithm 1: CAV Implementation Algorithm with Adversary.

Input: Initial reference state $x_{\text{ref}}(0)$, constraint sets $\mathcal{X}_{\text{cstr}}(\cdot)$, initial difference reachable set $\mathcal{R}_{0,\mathcal{U},\mathcal{W}}^d$, adversarial bound \bar{w} , linear-feedback matrix K , time horizon N ;
Output: Existence of a feasible control input sequence $u(\cdot)$, objective function \mathcal{L} ;

- 1: $\Phi \leftarrow \emptyset$;
- 2: **for** $k \in [1, N]$ **do**
- 3: $\mathcal{R}_{k,\mathcal{U},\mathcal{W}}^d \leftarrow \text{SETCOMPUTE}\mathcal{R}_{k-1,\mathcal{U},\mathcal{W}}^d, \bar{w}, K$;
- 4: $\mathcal{X}_{\text{ref}}(k) \leftarrow \text{SETSTRENGTHEN}\mathcal{X}_{\text{cstr}}(k), \mathcal{R}_{k,\mathcal{U},\mathcal{W}}^d$;
- 5: $\Phi^{(k)} \leftarrow \text{LOGICFORMULATE}\mathcal{X}_{\text{ref}}(k)$;
- 6: $\Phi \leftarrow \text{LOGICCONNECT}\Phi, \Phi^{(k)}$;
- 7: **end for**
- 8: $u_{\text{ref}}(\cdot) \leftarrow \text{SOLVEOPTSMT}\Phi, \mathcal{L}$;
- 9: **if** $u_{\text{ref}}(\cdot) = \emptyset$ **then**
- 10: **return** *Failed*;
- 11: **else**
- 12: $u(\cdot) \leftarrow \text{FEEDBACKCONTROL}u_{\text{ref}}(\cdot), x_{\text{ref}}(0), K$;
- 13: **return** $u(\cdot)$;
- 14: **end if**

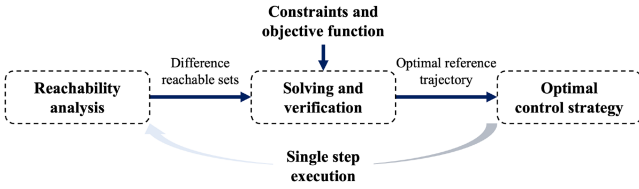


Fig. 5. CAV control algorithm under the RHC framework.

6) *FeedbackControl*: It transforms the control inputs from the reference trajectory to the actual one using (8).

The CAV control process typically lasts for a long time. Owing to the accumulation of adversarial uncertainty, a large duration (i.e., a large N) of the abovementioned algorithm could result in infeasibility of Φ . Therefore, to better adapt the algorithm to the traffic, the framework of receding horizon control (RHC) enables the efficacy of the algorithm in the long-duration scenario. Herein, the core algorithm remains unchanged within the framework, but it is executed iteratively (see Fig. 5). Every time the SMT solver outputs the effective optimal solution, only the first element of the sequential control inputs is adopted and executed. When the updated system state is observed, the algorithm is triggered again. The RHC framework ensures that the CAV control algorithm continues to run against the bounded adversary.

V. SIMULATION EXPERIMENTS AND RESULTS

In this section, we first introduce the experiment setup, including the car-following scenario, our autonomous driving platform, and vehicle dynamics model. Simulations and experiments under various conditions are then conducted to validate the proposed CAV control method in Section IV.

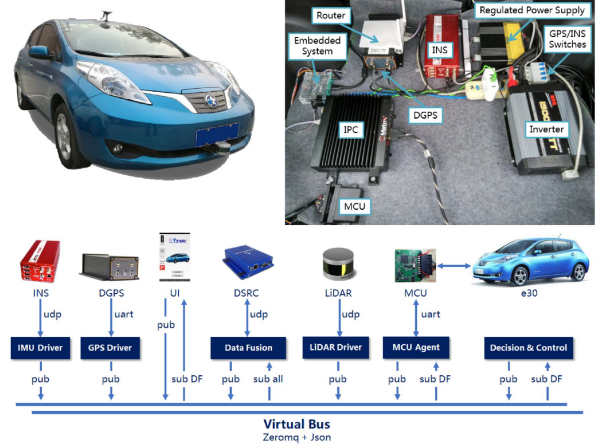


Fig. 6. Venucia E30 autonomous driving platform with the intra-vehicle communication network.

A. Experiment Setup

The simulation experiments were conducted considering the typical car-following scenario with adversary, where the ego-vehicle was controlled by the proposed method. In addition, we assumed that there were two vehicles in front and behind with the identical speed profile. The bounded adversary would potentially induce the ego-vehicle to collide with the two vehicles. Therefore, CAVs without adversary-proof controllers could be at high risk.

To improve the authenticity of the experiment, we first utilized a reliable autonomous driving platform (Venucia E30), as shown in Fig. 6, to obtain the recorded speed profile for the preceding vehicle. The vehicle platform, equipped with the environmental sensors of camera, LiDAR, inertial navigation system (INS), and differential global positioning system (DGPS), was following a preceding vehicle to record the speed and acceleration data of the preceding vehicle on a slightly congested road. A pub-sub socket-based communication network was designed to facilitate the data exchange among the sensors and the industrial personal computer (IPC). Hence, the kinematic data of the preceding vehicle collected by the sensors on the platform can be transmitted to the IPC and stored there. The recorded speed profile used in simulation is a segment intercepted from the actual measured data.

We proceed to introduce the model for the ego-vehicle in the car-following scenario. The linear vehicle dynamics model is presented as

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1/T_L \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ K_L/T_L \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ K_L/T_L \end{bmatrix} w$$

$$x = \begin{bmatrix} s & v & a \end{bmatrix}^T, u = a_{\text{des}}, w = a_{\text{adv}} \quad (19)$$

where s , v , and a are the position, speed, and acceleration, respectively, which form the ego-vehicle state variable x , a_{des} is the desired acceleration of the vehicle or the control input of the system with the bound $a_{\text{des}} \in [-5, 5] \text{ m/s}^2$, a_{adv} is the

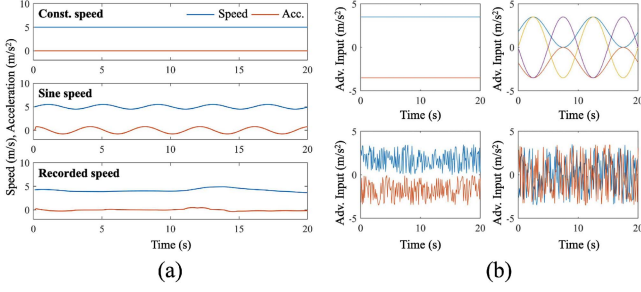


Fig. 7. Various conditions. (a) Speed profiles of the preceding vehicle. (b) Generated input sequences of the bounded adversary.

adversarial input with the bound $\bar{w} = 3.5 \text{ m/s}^2$, T_L is the parameter considering the time-delay characteristics of vehicle control and is set to 0.45, and K_L is the steady-state gain of the closed-loop system and is set to 1.0. The continuous system in (19) is discretized via the zero-order holder method with the sampling time of 0.1 s.

To comprehensively demonstrate the vehicle control performance, various conditions in the simulation were created, including the speed profiles of the preceding vehicle, as shown in Fig. 7(a), and the potential bounded adversarial inputs, as shown in Fig. 7(b). The preceding vehicle is assumed to be running in three ways: 1) at constant speed; 2) sine speed; and 3) recorded speed. Note that the proposed algorithm is applicable to arbitrary bounded adversary, and we consider these specific scenarios for simulation. Thus, ten adversarial input sequences were generated, which may have a purpose, such as leading to the ego-vehicle collision with the preceding vehicle. The simulations were carried out in MATLAB R2019b in a laptop with Intel Core i7-8565 U CPU.

To demonstrate the practicability of the proposed algorithm, we implemented it on the E30 platform, which followed another similar autonomous driving platform (Chang-an CS55) on a straight road. The preceding vehicle CS55 was self-driving, according to the recorded speed profile. The ego-vehicle E30 embedded and controlled with the RA-SMT was following the preceding vehicle. The two vehicles were connected by the LTE-V2X network, with which the real-time positions and speeds obtained by their DGPS and INS could be shared. The RA-SMT was running on the MXE-5401 IPC.

B. Reachability Analysis and Optimal Control

The reachability analysis is conducted considering the state-feedback control. The feedback matrix is calculated with the linear-quadratic regulator method. However, first, the weight matrices reflecting the optimization objective need to be determined

$$Q = \text{diag}(0.5, 0.2, 0.2), R = 0.1 \quad (20)$$

where Q denotes the deviation cost between the actual state and the desired state and R denotes the cost of the control inputs; their values originate from the common consideration of both the vehicle tracking performance and the energy saving

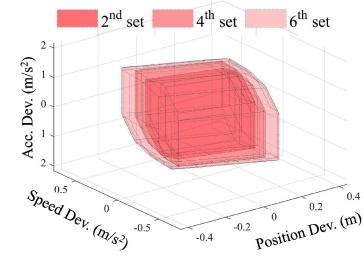


Fig. 8. Difference reachable sets in 3-D space.

performance. Consequently, the Riccati equation can be used to solve the feedback matrix $K = [-2.2361, -3.6329, -1.5039]^T$ in MATLAB.

In this scenario, the deviation between the actual state and the reference state is specified as

$$x_d(k) = \begin{bmatrix} \Delta s(k) & \Delta v(k) & \Delta a(k) \end{bmatrix}^T \quad (21)$$

where Δs , Δv , and Δa denote the deviation between the actual trajectory and the reference trajectory in terms of position, speed, and acceleration, respectively.

The initial difference reachable set should also be determined before the set computation as follows:

$$\mathcal{R}_{0,\mathcal{U},\mathcal{W}}^d = \mathcal{S}^d = (0_{3 \times 1}, \langle I_{3 \times 3} \rangle) \quad (22)$$

where $\mathcal{R}_{0,\mathcal{U},\mathcal{W}}^d$ and \mathcal{S}^d denote the initial sets, represented by zonotope, its center is the zero vector and its generators are the identity matrix I .

Subsequently, the set computation is executed considering the prediction horizon equal to 6 or 0.6 s. MPT3 and CORA2018 are used as auxiliary tools in set analysis, computation, and presentation [34], [37]. Following the iterative computations, the exact results obtained are shown in Fig. 8, which presents the 3-D difference reachable sets. The set expansion over time indicates that the deviation of the vehicle actual state from the reference increases owing to the adversary.

The constraint set $\mathcal{X}_{\text{ctr}}(k)$ for the actual trajectory at every time k in the car-following scenario are known and can be derived in polytope form by the following inequalities:

$$\mathcal{X}_{\text{ctr}}(k) = \begin{cases} |[s_p(k) - s(k)] - d_{\text{des}}| \leq d_{\text{safe}} \\ |v_p(k) - v(k)| \leq \Delta v_{\text{safe}} \\ |a(k)| \leq a_{\text{max}} \end{cases} \quad (23)$$

where s_p and v_p denote the position and speed of the preceding vehicle, respectively, d_{des} is the desired tracking distance from the preceding vehicle set to 4 m, d_{safe} and Δv_{safe} are the limit values of the tracking error in distance and speed set to 2 m and 1 m/s, respectively, and a_{max} is the limit of the acceleration set to 5 m/s² in the simulation.

We can now embody the optimal control problem with the objective function expressed as

$$\mathcal{L} = \omega^T \sum_{k=1}^N \|x_{\text{ref}}(k) - x_{\text{des}}(k)\|_2 \quad (24)$$

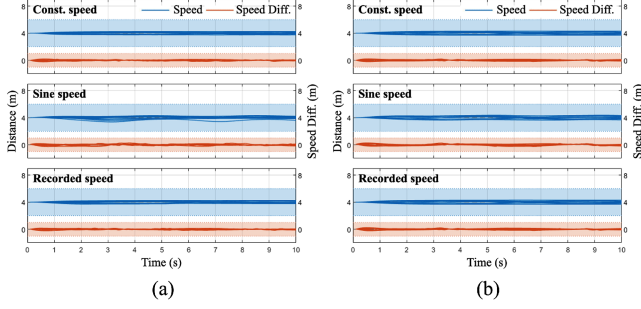


Fig. 9. Simulation results. (a) RA-SMT. (b) Tube-based rMPC.

where x_{des} denotes the desired state, which we assume to mimic the speed profile of the preceding vehicle and maintain the desired tracking distance, and ω is the weight vector set to $[200, 20, 1]^T$. The objective function in (24) implying that the vehicle trajectory should exhibit minimal deviation from its desired trajectory. The corresponding constraints can be obtained by set computing between the constraint set $\mathcal{X}_{\text{ctr}}(k)$ and the difference reachable set $\mathcal{R}_{k,\mathcal{U},\mathcal{W}}^d$, and by the first-order logic conversion such that SMT can be applied.

Thus, the reach-avoid specification in the car-following scenario can be achieved. The corresponding safe sets are the constraint sets derived from (23); the corresponding goal set is made to be the constraint set at the end of the planning period, implying that the vehicle should continuously maintain good tracking capability.

C. CAV Control Performance Under Various Conditions

The RHC framework is adopted and the algorithm is executed iteratively with the prediction horizon set to 4. Thereafter, various simulations are conducted with the different conditions. In addition, the bounded adversary can be considered as an uncertain input. Classical MPC methods experience loss in efficacy in terms of maintaining state safety and control optimization. In contrast, the robust MPC (rMPC) (e.g., tube-based methods) can deal with such uncertainty in this car-following scenario [38]. To demonstrate the control performance of the proposed RA-SMT algorithm, the tube-based rMPC with the same objective function and parameters is implemented as the benchmark for the contrastive analysis. Specifically, the MPT was used to conduct reachability analysis and controller synthesis for the benchmark, as reported in [26]. Both methods are within the framework of RHC, mainly including reachability analysis and optimal control. Nevertheless, RA-SMT exploits efficient zonotope for recursive computation of reachable sets and leverages flexible SMT solver for optimization of control strategies. The benchmark method uses vanilla polytope for reachability analysis and quadratic programming solver for optimal control.

The simulation results of the two algorithms with all the generated adversarial input sequences are shown in Fig. 9. Each subgraph, with the shaded parts highlighting the areas within the constraints, illustrates the tracking performance (of all the ten adversarial input sequences) with one speed profile of the preceding vehicle. The results show that both algorithms can

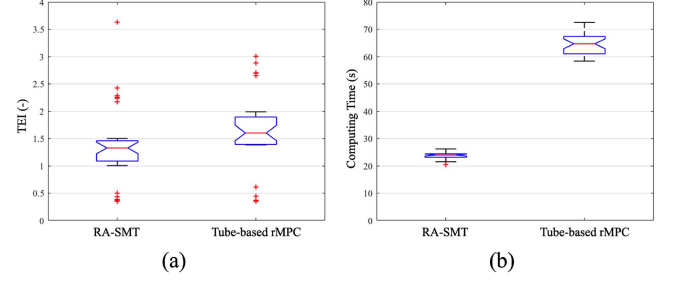


Fig. 10. RA-SMT and tube-based rMPC comparison. (a) Control performance. (b) Algorithm efficiency.

ensure safety (the state of the controlled vehicle is always within the predetermined boundary), and can maintain certain tracking performance with various types of the adversarial inputs and the speed profiles of the preceding vehicle.

In addition to ensuring vehicle control safety, control performance and algorithm efficiency are also important aspects. Precisely, we utilize the tracking error index (TEI) [39] as the reverse evaluation metric to indicate the control performance in the car-following scenario, defined as

$$\text{TEI} = \frac{1}{N_{\text{simu}}} \sum_{k=0}^{N_{\text{simu}}} \left(\omega_d |\Delta d(k)|^2 + \omega_v |\Delta v(k)|^2 \right) \quad (25)$$

where N_{simu} is the total simulation steps, Δd denotes the tracking distance error deviating from the desired distance (i.e., $\Delta d = s_p - s - d_{\text{des}}$), Δv denotes the tracking speed error deviating from the speed of the preceding vehicle (i.e., $\Delta v = v_p - v$), and ω_d and ω_v denote the weight scalars of the distance error and the speed error with values of 1.0 and 0.1, respectively. In terms of calculation, the 2-norm TEI is designed to penalize larger errors in tracking speed and distance while ignoring smaller ones, which coincides with drivers' behavior during a car-following process. To better calibrate the tracking performance, the cumulative errors are averaged over the tracking time. Moreover, for each simulation round, the computing time was recorded as the indicator of the algorithm efficiency for comparison.

All the original results are summarized and presented in boxplot graphs. Fig. 10 shows the control performance and algorithm efficiency of the two algorithms, respectively. The results indicate that the control performance of the RA-SMT is better than that of the tube-based rMPC, with an average TEI decrease of approximately 17%. Regarding the algorithm efficiency, the RA-SMT significantly outperforms the tube-based rMPC, with a considerable reduction in computing time of approximately 63% in almost all the simulation rounds. Furthermore, the computing time of the RA-SMT exhibits minimal to no fluctuations, implying that the RA-SMT is much more efficient in terms of stability. The better control performance and algorithm efficiency of the RA-SMT are attributed to the efficient reachability analysis performed with zonotope sets and the optimal controller synthesis with bounded adversary.

For better benchmarking, the medium phase of the world-wide harmonized light vehicles test procedure (WLTP-M) is introduced, as given [40]. The preceding vehicle mimics the

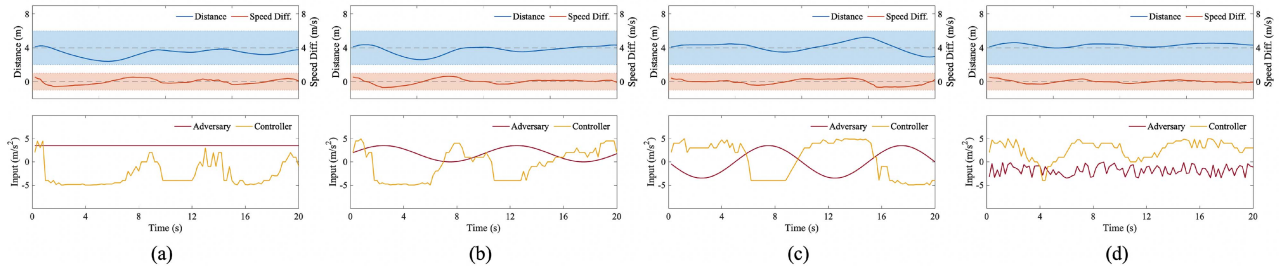


Fig. 11. Experiment results of a vehicle with RA-SMT following a preceding vehicle with the recorded speed profile with representative adversarial inputs. (a) Step signal. (b) Small sinusoidal signal. (c) Large sinusoidal signal. (d) Random signal.

TABLE II
ALGORITHM PERFORMANCE WITH WLTP-M DRIVE CYCLE

Metric	RA-SMT	Tube-based rMPC	Improvement
Avg. TEI (-)	1.40	1.68	16.7%
Avg. computing time (s)	771.9	2221.2	65.3%

speed profile of WLTP-M for comparison. We have obtained the similar simulation results, as shown in Fig. 9, where the vehicle state is optimally controlled within the predetermined region for all the generated adversarial inputs. The average TEI and computing time are calculated and summarized in Table II, which coincides with the simulation results described previously.

D. Experimental Validation

During the experiments, RA-SMT calculates the controller input (i.e., the desired acceleration), which is immediately integrated to the target speed based on the current speed at each control period. The target speed is then transmitted to the MCU and CAN bus to complete the speed control loop. The control frequency is set to 5 Hz, which is enough for the RA-SMT to finish once calculation (around 0.10–0.15 s). Four representative adversarial inputs from Fig. 7(b) are modeled in the IPC to conduct the experiments. Other parameters remain unchanged compared with the previous simulation.

The experiment results of RA-SMT with four representative adversaries are shown in Fig. 11. It is shown that the RA-SMT does keep the state of the ego-vehicle within a predetermined region even with bounded adversaries, which reflects the same reach-avoid specification with the previous simulation. Moreover, the RA-SMT algorithm demonstrates its applicability to the real-vehicle in the car-following scenario.

VI. CONCLUSION

This article proposed a CAV control method, named RA-SMT, to deal with bounded integrity attacks from the IoV environment. This method efficiently addresses the uncertainty caused by the adversary over zonotope sets, and realizes the verification and optimal control simultaneously via the innovative incorporation of SMT. Particularly, it enables CAVs to possess the reach-avoid

specification, which strictly ensures vehicle control safety with adversary and is feasible in the worst case in theory. The incorporation of SMT in optimal controller synthesis inherently guarantees global optimality. Furthermore, the method is flexible and convenient for describing various constraints with the aid of SMT, and thus, has the potential to better adapt to complex traffic scenarios, which is beyond the ability of the benchmark tube-based rMPC [26].

The simulation results of the car-following scenario revealed its effectiveness against the bounded adversary under various conditions, and demonstrated its improved control performance and algorithm efficiency compared with the benchmark. The experiments conducted with the RA-SMT algorithm on an autonomous driving platform exhibited its practicability. Interesting future work includes exploiting learning-based method to generate malicious adversar, and applying the proposed approach to more complex traffic scenarios, such as lane changing of CAVs with surrounding vehicles.

REFERENCES

- [1] K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo, "Development of autonomous car—Part I: Distributed system architecture and development process," *IEEE Trans. Ind. Electron.*, vol. 61, no. 12, pp. 7131–7140, Dec. 2014.
- [2] K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo, "Development of autonomous car—Part II: A case study on the implementation of an autonomous driving system based on distributed architecture," *IEEE Trans. Ind. Electron.*, vol. 62, no. 8, pp. 5119–5132, Aug. 2015.
- [3] R. A. Gupta and M.-Y. Chow, "Networked control system: Overview and research trends," *IEEE Trans. Ind. Electron.*, vol. 57, no. 7, pp. 2527–2535, Jul. 2010.
- [4] Y. Yuan, H. Yang, L. Guo, and F. Sun, *Analysis and Design of Networked Control Systems Under Attacks*. Boca Raton, FL, USA: CRC, 2018.
- [5] Y. Li, D. Shi, and T. Chen, "False data injection attacks on networked control systems: A Stackelberg game analysis," *IEEE Trans. Autom. Control*, vol. 63, no. 10, pp. 3503–3509, Oct. 2018.
- [6] Z. Fang, W. Zhang, Z. Li, H. Tang, H. Han, and F. Xu, "Revisiting attacks and defenses in connected and autonomous vehicles," in *Proc. Int. Conf. Secur., Privacy Anonymity Comput., Commun. Storage*, 2020, pp. 104–117.
- [7] Y. Mo and B. Sinopoli, "Integrity attacks on cyber-physical systems," in *Proc. 1st Int. Conf. High Confidence Netw. Syst.*, 2012, pp. 47–54.
- [8] S. N. Narayanan, S. Mittal, and A. Joshi, "OBD_SecureAlert: An anomaly detection system for vehicles," in *Proc. IEEE Int. Conf. Smart Comput.*, 2016, pp. 1–6.
- [9] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, pp. 1–33, 2011.

- [10] V. Behzadan and A. Munir, "Adversarial reinforcement learning framework for benchmarking collision avoidance mechanisms in autonomous vehicles," *IEEE Intell. Transp. Syst. Mag.*, vol. 13, no. 2, pp. 236–241, 2021.
- [11] X. Ma, K. Driggs-Campbell, and M. J. Kochenderfer, "Improved robustness and safety for autonomous vehicle control with adversarial reinforcement learning," in *Proc. IEEE Intell. Veh. Symp.*, 2018, pp. 1665–1671.
- [12] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent Hamilton–Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. Autom. Control*, vol. 50, no. 7, pp. 947–957, Jul. 2005.
- [13] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry, "Reach-avoid problems with time-varying dynamics, targets and constraints," in *Proc. 18th Int. Conf. Hybrid Syst.: Comput. Control*, 2015, pp. 11–20.
- [14] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton–Jacobi reachability: A brief overview and recent advances," in *Proc. IEEE 56th Annu. Conf. Decis. Control*, 2017, pp. 2242–2253.
- [15] B. Schürmann and M. Althoff, "Guaranteeing constraints of disturbed nonlinear systems using set-based optimal control in generator space," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11 515–11 522, 2017.
- [16] B. Schürmann, D. Heß, J. Eilbrecht, O. Stursberg, F. Köster, and M. Althoff, "Ensuring drivability of planned motions using formal methods," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.*, 2017, pp. 1–8.
- [17] F. Ke, Z. Li, and C. Yang, "Robust tube-based predictive control for visual servoing of constrained differential-drive mobile robots," *IEEE Trans. Ind. Electron.*, vol. 65, no. 4, pp. 3437–3446, Apr. 2018.
- [18] L. Dai, Y. Lu, H. Xie, Z. Sun, and Y. Xia, "Robust tracking model predictive control with quadratic robustness constraint for mobile robots with incremental input constraints," *IEEE Trans. Ind. Electron.*, vol. 68, no. 10, pp. 9789–9799, Oct. 2021.
- [19] S. Cheng, L. Li, X. Chen, J. Wu, and H.-d. Wang, "Model-predictive-control-based path tracking controller of autonomous vehicle considering parametric uncertainties and velocity-varying," *IEEE Trans. Ind. Electron.*, vol. 68, no. 9, pp. 8698–8707, Sep. 2021.
- [20] Z. Huang, Y. Wang, S. Mitra, G. E. Dullerud, and S. Chaudhuri, "Controller synthesis with inductive proofs for piecewise linear systems: An SMT-based algorithm," in *Proc. IEEE Conf. 54th Decis. Control*, 2015, pp. 7434–7439.
- [21] Z. Huang, Y. Wang, S. Mitra, and G. Dullerud, "Controller synthesis for linear dynamical systems with adversaries," in *Proc. Symp. Bootcamp Sci. Secur.*, 2016, pp. 53–62.
- [22] C. Fan, U. Mathur, S. Mitra, and M. Viswanathan, "Controller synthesis made real: Reach-avoid specifications and linear dynamics," in *Int. Conf. Comput. Aided Verification*, 2018, pp. 347–366.
- [23] L. De Moura and N. Bjørner, "Satisfiability modulo theories: Introduction and applications," *Commun. ACM*, vol. 54, no. 9, pp. 69–77, 2011.
- [24] Y. Shoukry, P. Nuzzo, A. Puggelli, A. L. Sangiovanni-Vincentelli, S. A. Seshia, and P. Tabuada, "Secure state estimation for cyber-physical systems under sensor attacks: A satisfiability modulo theory approach," *IEEE Trans. Autom. Control*, vol. 62, no. 10, pp. 4917–4932, Oct. 2017.
- [25] A. Dutta, E. Al-Shaer, and S. Chatterjee, "Constraints satisfiability driven reinforcement learning for autonomous cyber defense," 2021, *arXiv:2104.08994*.
- [26] M. Kvasnica, B. Takács, J. Holaza, and D. Ingole, "Reachability analysis and control synthesis for uncertain linear systems in MPT," *IFAC-PapersOnLine*, vol. 48, no. 14, pp. 302–307, 2015.
- [27] M. Cai et al., "Formation control for connected and automated vehicles on multi-lane roads: Relative motion planning and conflict resolution," 2021, *arXiv:2103.10287*.
- [28] C. Chen et al., "Conflict-free cooperation method for connected and automated vehicles at unsignalized intersections: Graph-based modeling and optimality analysis," 2021, *arXiv:2107.07179*.
- [29] X. Jin, W. M. Haddad, Z.-P. Jiang, and K. G. Vamvoudakis, "Adaptive control for mitigating sensor and actuator attacks in connected autonomous vehicle platoons," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 2810–2815.
- [30] Z. Zhou, R. Takei, H. Huang, and C. J. Tomlin, "A general, open-loop formulation for reach-avoid games," in *Proc. IEEE 51st Conf. Decis. Control*, 2012, pp. 6501–6506.
- [31] P. M. Esfahani, D. Chatterjee, and J. Lygeros, "The stochastic reach-avoid problem and set characterization for diffusions," *Automatica*, vol. 70, pp. 43–56, 2016.
- [32] M. B. Saltik, L. Özkan, J. H. Ludlage, S. Weiland, and P. M. van den Hof, "An outlook on robust model predictive control algorithms: Reflections on performance and computational aspects," *J. Process Control*, vol. 61, pp. 77–102, 2018.
- [33] I. B. Makhlof, J. Gan, and S. Kowalewski, "A study on solving guard and invariant set intersection in zonotope-based reachability of linear hybrid systems," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 13–20, 2015.
- [34] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari, "Multi-parametric toolbox 3.0," in *Proc. Eur. Control Conf.*, 2013, pp. 502–510.
- [35] N. Bjørner, A.-D. Phan, and L. Fleckenstein, "vZ-AN optimizing SMT solver," in *Proc. Int. Conf. Tools Algorithms Construction Anal. Syst.*, 2015, pp. 194–199.
- [36] R. Araújo, I. Bessa, L. C. Cordeiro, and J. E. C. Filho, "SMT-based verification applied to non-convex optimization problems," in *Proc. VI Braz. Symp. Comput. Syst. Eng.*, 2016, pp. 1–8.
- [37] M. Althoff, D. Grebenyuk, and N. Kochdumper, "Implementation of taylor models in CORA 2018," in *Proc. 5th Int. Workshop Appl. Verification Continuous Hybrid Syst.*, 2018, pp. 145–173.
- [38] D. Q. Mayne, E. C. Kerrigan, and P. Falugi, "Robust model predictive control: Advantages and disadvantages of tube-based methods," *IFAC Proc. Vol.*, vol. 44, no. 1, pp. 191–196, 2011.
- [39] J. Pan, Q. Xu, K. Li, and J. Wang, "Controller design for V2X application under unreliable feedback channel," in *Proc. IEEE Conf. Intell. Transp. Syst.*, 2019, pp. 2496–2502.
- [40] G. Frezza and S. A. Evangelou, "Ecological adaptive cruise controller for a parallel hybrid electric vehicle," in *Proc. Eur. Control Conf.*, 2020, pp. 491–498.



Qing Xu received the B.S., M.S., and Ph.D. degrees in automotive engineering from Beihang University, Beijing, China, in 2006, 2008, and 2014, respectively.

During his Ph.D. research, he was a Visiting Scholar with the Department of Mechanical Science and Engineering, UIUC. From 2014 to 2016, he was a Postdoctoral Researcher with Tsinghua University, Beijing, China, where he is currently an Assistant Research Professor with the Department of Automotive Engineering. His research interests include decision and control of intelligent vehicles.



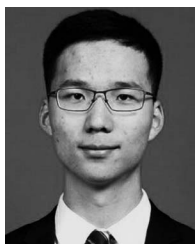
Yicong Liu received the B.E. degree in automotive engineering and the M.S. degree in mechanical engineering, in 2017 and 2020, respectively, from Tsinghua University, Beijing, China, where he is currently working toward the Ph.D. degree in mechanical engineering with the School of Vehicle and Mobility. From 2017 to 2018, he received the dual master's degree in mechanical engineering from the Faculty of Mechanical Engineering, RWTH Aachen University, Aachen, Germany.

His research focuses on intelligent control of connected and autonomous vehicles.



Jian Pan received the B.E. degree in automotive engineering in 2017 from Tsinghua University, Beijing, China, where he is currently working toward the Ph.D. degree in mechanical engineering with the School of Vehicle and Mobility.

His research focuses on control of connected vehicle under unreliable communication.



Jiawei Wang (Graduate Student Member, IEEE) received the B.E. degree in automotive engineering in 2018 from Tsinghua University, Beijing, China, where he is currently working toward the Ph.D. degree in mechanical engineering with the School of Vehicle and Mobility.

His research interests include distributed control and optimization.

Mr. Wang was the recipient of the National Scholarship in Tsinghua University and the Best Paper Award at 18th COTA International Conference.



Keqiang Li received the B.Tech. degree in automotive engineering from Tsinghua University, Beijing, China, in 1985, and the M.S. and Ph.D. degrees in mechanical engineering from Chongqing University, Chongqing, China, in 1988 and 1995, respectively.

He is currently an Academician with the Chinese Academy of Engineering, Beijing, China, and a Professor with the School of Vehicle and Mobility, Tsinghua University. He has authored more than 200 papers and is a co-inventor over

80 patents in China and Japan. His research interests include automotive control system and networked dynamics and control.



Jianqiang Wang received the B.Tech. and M.S. degrees in automotive application engineering from the Jilin University of Technology, Changchun, China, in 1994 and 1997, respectively, and the Ph.D. degree in vehicle operation engineering from Jilin University, Changchun, China, in 2002.

He is currently a Professor with School of Vehicle and Mobility, Tsinghua University, Beijing, China. He has authored more than 80 journal papers and is currently a co-holder of more than

80 patent applications. His research interests include intelligent vehicles, driving assistance systems, and driver behavior.