

Distributed Model Predictive Control for Connected and Automated Vehicles in the Presence of Uncertainty

Baisravan HomChaudhuri¹

Assistant Professor
Mem. ASME

Autonomous Control Systems Laboratory,
Mechanical and Aerospace
Engineering Department,
Illinois Institute of Technology,
Chicago, IL 60616
e-mail: bhomchaudhuri@iit.edu

Viranjan Bhattacharyya

Illinois Institute of Technology,
Chicago, IL 60616
e-mail: vbhattacharyya@hawk.iit.edu

This article focuses on the development of distributed robust model predictive control (MPC) methods for multiple connected and automated vehicles (CAVs) to ensure their safe operation in the presence of uncertainty. The proposed layered control framework includes reference trajectory generation, distributionally robust obstacle occupancy set computation, distributed state constraint set evaluation, data-driven linear model representation, and robust tube-based MPC design. To enable distributed operation among the CAVs, we present a method, which exploits sampling-based reference trajectory generation and distributed constraint set evaluation methods, that decouples the coupled collision avoidance constraint among the CAVs. This is followed by data-driven linear model representation of the nonlinear system to evaluate the convex equivalent of the nonlinear control problem. Finally, to ensure safe operation in the presence of uncertainty, this article employs a robust tube-based MPC method. For a multiple CAV lane change problem, simulation results show the efficacy of the proposed controller in terms of computational efficiency and the ability to generate safe and smooth CAV trajectories in a distributed fashion. [DOI: 10.1115/1.4054696]

Keywords: autonomous mobility and maneuver, controls and observation, intelligent decision making, vehicle autonomy

1 Introduction

Due to the high number of traffic crashes and fatalities [1], as highlighted in National Highway Traffic Safety Administration data [2], safety is one of the major concerns of the current transportation system. This has propelled research in controller design for the connected and automated vehicles (CAVs). Here, every CAV needs to evaluate its own control solution, while avoiding the “unconnected” (any vehicle that does not share information) and the other connected vehicles. Controller synthesis for CAVs is a challenging problem because of various uncertainties in the system, such as process noise in ego vehicle dynamics and uncertain motion of the vehicles surrounding the ego vehicle. In addition to that, nonlinear dynamics of the ego vehicle, nonconvex collision avoidance constraints, distributed nature of the problem, and the requirement of real-time applicability add to the challenges. In this article, we aim at developing a computationally tractable distributed robust MPC strategy for the CAVs, which ensures collision avoidance in the presence of system uncertainties. The proposed method is “distributed” because each CAV takes its own actions with local communication and information exchange. In this article, system uncertainty includes affine process noise in ego vehicle dynamics and uncertain motion of the vehicles surrounding the ego vehicle.

Connected and automated vehicles or autonomous vehicles are generally controlled in a receding horizon fashion, which allow them to utilize information of the surrounding environment. Generally, the overall control architecture involves a high-level sampling-based trajectory planner and a low-level controller that follows the high-level waypoints [3]. Among the various vehicle maneuvers, this article focuses on the lane change problem of the CAVs, which has gained a lot of attention in recent years. The lane

change maneuver is challenging since it involves lateral nonlinear vehicle dynamics. The nonlinear dynamics and the collision avoidance constraints make the receding horizon control problem non-convex. Uncertainty in the prediction of the unconnected vehicles, process noise in the CAVs, and distributed nature of operation add to the problem challenges.

The previous work has focused on the lane change problem for autonomous vehicles [4–15]. However, most of them consider knowledge of a reference trajectory, only focus on identifying if a lane change maneuver is possible [13], do not consider both lateral and longitudinal controller design, or separately solve the lateral and longitudinal control problem. For instance, Ref. [4] only focuses on longitudinal control and ignores lateral trajectory planning. On the other hand, [5] only considers lateral controller design, while assuming constant longitudinal velocity. The authors in Refs. [6,7] consider existence of a reference trajectory, and Ref. [7] uses motion primitives. The authors in Refs. [8–11] solve the lane change problem by separately developing longitudinal and lateral controllers. The authors in Ref. [15] developed a centralized lane change approach, while the authors in Ref. [14] focused on a negotiation-based approach. In a recent work, the authors in Ref. [16] have developed a Monte Carlo Tree Search based lane change algorithm. However, this approach relies on discretization of the control space and hence leads to suboptimal solutions. The authors in Ref. [17] provided limits on lateral acceleration that prevents vehicle roll over, which is critical for vehicle safety. Most of the previous work in the literature do not consider simultaneous lateral and longitudinal controller development and ignore the multivehicle lane change problem, where multiple vehicles perform the lane change maneuver. These research gaps are addressed in this article.

Recent works on multivehicle decision-making are available in Refs. [18–21], where authors in Refs. [19–21] specifically considered the lane change problem. The authors in Ref. [18] introduced the buffered input cell (BIC) to compute the inputs that ensure reciprocal collision avoidance. Its probabilistic counterpart

¹Corresponding author.

Manuscript received January 5, 2022; final manuscript received May 26, 2022; published online July 1, 2022. Assoc. Editor: Benjamin Duprey.

(probabilistic BIC) was introduced in Ref. [20]. The BIC and probabilistic BIC concepts were extended to multivehicle lane change problem in Refs. [19,20], respectively. However, the BIC or the probabilistic BIC-based approaches only use the current positions of the vehicles and not their predicted trajectory over the horizon. This highly restricts the state and control constraints of the vehicles, which can lead to highly conservative solutions. Also, the authors in Ref. [19] did not consider uncertainties (process noise or uncertain surrounding vehicle motion). The authors in Ref. [20] employed a sampling-based approach, which can provide incomplete probability distribution information, and thus can lead to collision. A learning-based approach is presented in Ref. [21], where vehicles (in simulation) learn the simultaneous lane change maneuver from the human experts. However, this does not provide any guarantees of collision avoidance and highly depends on the data set used for training.

The CAV control problem, especially the lane change problem, is a nonconvex optimal control problem, since it includes nonconvex state constraints (collision avoidance constraints) and nonlinear vehicle dynamics. Nonconvex problems are computationally expensive and can lead to highly suboptimal solutions. System uncertainty, including ego vehicle process noise and uncertain surrounding vehicles, adds to the problem complexity. Furthermore, the control solution is required to be generated in a distributed fashion, where each CAV needs to solve its own control problem while ensuring collision avoidance. However, distributed controller synthesis is challenging because of the coupled collision avoidance constraint, where action of one CAV influences the others. Recent methods [22] that try to address this issue for a drone racing problem uses an iterative approach at every time instant, which is not computationally tractable, and hence hinder real-time application. In this article, “real-time” applicability of an algorithm refers to its ability to obtain the solution within the sampling time.

Collision avoidance in the presence of dynamic and uncertain surrounding vehicles (obstacles) is also a challenging task. To avoid stochastic obstacles, a popular approach is characterizing the regions in state space that the obstacles could occupy [23–25] and then avoiding those regions in the vehicle’s trajectory planning. Computation of these future obstacle occupancy sets are generally done by assuming the true probability distribution of the disturbances to be known or by only using the bounds on the disturbances. Assumption of known true probability distribution can be a hard assumption for many real-world problems, and only using the bounds of the disturbances can lead to highly conservative solution [24]. In this article, we consider the first and second moment of the uncertainty associated with obstacle prediction error to lie within a known confidence interval, and its true probability distribution is unknown. We then develop a method to obtain the future probabilistic occupancy of these obstacles to systematically obtain the regions in state space the CAVs should avoid and facilitate convexification of their optimal control problem.

To address these challenges, in this article, an effort has been made to develop a computationally efficient distributed control strategy for each CAV in a multivehicle scenario in the presence of process noise in the ego CAV and uncertainty in obstacle motion. The proposed approach is “distributed,” since each vehicle takes its own action with local information exchange. The control strategy is developed in a layered framework, which includes development of distributionally robust future occupancy of the obstacles, development of distributed reference trajectory generation, convexification, distributed constraint set generation, dynamic mode decomposition (DMD)-based linear model representation, and robust tube-based MPC methods. We propose a new sampling-based distributed reference trajectory generation method that exploits the differential flatness property of the nonlinear vehicle dynamics [19] and cooperation among the CAVs. The proposed sampling-based approach only samples the terminal position of the trajectory and needs to sample only once to generate the full state trajectory (multiple such trajectories are generated). Hence, it is more computationally efficient than the sampling-based

approaches that sequentially grow a tree, such as rapidly exploring random tree (RRT) [26]. This is followed by state constraint convexification and distributed constraint set generation that evaluates the nonoverlapping convex state constraint sets for each CAV. To formulate the convex equivalent of the nonconvex optimal control problem, this article exploits a DMD-based approach [27,28] to develop a linear system model using the previous data on state and control trajectories. Finally, robust tube-based MPC method is employed to guarantee collision avoidance over the control horizon and successful lane change maneuver. This layered control strategy promotes computational tractability, avoids iterative decision-making in each time-step [22] or sequential decision-making (where only one system updates its solution at a particular time [29]), and ensures robust collision avoidance.

In our previous works [30,31], we have developed control strategies based on robust tube-based MPC method [30] and distributionally robust stochastic MPC method [31] for a single autonomous vehicle. However, the previous works [30,31] did not consider multiple CAV lane change problem and assumed the mean and covariance of the prediction error to be known. Moreover, Taylor series approximation-based linearization was used in Refs. [30,31], which requires small deviation from the trajectory around which the system is linearized. In this article, we extend those works by considering a distributed control framework, the mean and covariance of the prediction error to lie in a confidence interval, and data-driven linear system representation that increases the solution space for the controller.

The article contributions can be listed as follows: (i) development of a layered control strategy that involves evaluation of a sampling-based reference trajectory that exploits the differential flatness property of the bicycle model to allow nonconservative simultaneous lane change maneuvers; and robust tube-based MPC to guarantee collision avoidance in the presence of uncertainty, (ii) development of a distributed constraint set generation approach to facilitate distributed operation, (iii) exploitation of DMD-based linear model representation to formulate a convex equivalent problem, and (iv) development of method to obtain the distributionally robust future occupancy of the obstacles.

2 Problem Description

We consider a scenario including multiple cooperative and noncooperative (not adversarial) CAVs. Cooperative CAVs share information with each other (specifically, their future state and control trajectories over the horizon) and adjust their trajectories considering the surrounding CAVs. The noncooperative vehicles, which can be considered to share their information, are considered not to adjust their trajectories with the information they receive from the other CAVs. The “noncooperative” vehicles can be referred to as obstacles. This article only focuses on the control problem of the CAVs, and, hence, if a CAV does not receive any information from a vehicle, the CAV needs to predict its future trajectory. We consider the set I_V includes all the vehicles in the environment (surrounding the ego CAV), while I_{CV} and I_{UV} are the sets of cooperative and noncooperative vehicles, respectively (so that $I_{CV} \cup I_{UV} = I_V$).

2.1 System Model. In this article, we consider the nonlinear vehicle dynamics with affine disturbance (the process noise)

$$z_i(k+1) = f_i(z_i(k), v_i(k)) + B_{wi}w_i(k) \quad (1)$$

where $z_i(k) \in \mathbb{Z} \times \mathbb{R}^4$ is the vehicle state vector, $v_i(k) \in V_i$ is the control vector, $w_i(k) \in W_i \subset \mathbb{R}^{n_w}$ is the bounded disturbance, and $B_{wi} \in \mathbb{R}^{4 \times n_w}$. The nominal system dynamics (the dynamics without the disturbance) is given by

$$x_i(k+1) = f_i(x_i(k), v_i(k)) \quad (2)$$

where $x_i(k)$ is the nominal state vector. Following the previous research [32], we model the system dynamics of all the vehicles with the kinematic bicycle model. Based on Ref. [32], this model shows an acceptable performance in comparison with more detailed dynamic model. The discrete-time kinematic bicycle model of a vehicle i is described with state $x_i = [S_{ix}, S_{iy}, \psi_i, v_i]^T$ and sampling time Δt as follows

$$S_{ix}(k+1) = S_{ix}(k) + \Delta t v_i(k) \cos(\psi_i(k)) \quad (3a)$$

$$S_{iy}(k+1) = S_{iy}(k) + \Delta t v_i(k) \sin(\psi_i(k)) \quad (3b)$$

$$\psi_i(k+1) = \psi_i(k) + \Delta t \frac{v_i(k)}{L_{ir} + L_{if}} \tan(\delta_{if}(k)) \quad (3c)$$

$$v_i(k+1) = v_i(k) + \Delta t a_i(k) \quad (3d)$$

where S_{ix} and S_{iy} are the coordinates of the rear axle center points of the vehicle. The heading angle of the vehicle is given by ψ_i , and v_i is its velocity. L_{if} and L_{ir} are the distances from the center of mass of the vehicle to the front and rear axles, respectively. The control vector includes the acceleration a_i and steering angle δ_{if} , i.e., $v_i = [a_i, \delta_{if}]^T \in \mathbb{R}^2$.

2.2 Optimal Control Problem. In this article, we focus on the lateral and longitudinal control of the CAVs. The control problem for each coordinating CAV $i \in \mathcal{I}_{CV}$ is given by

$$\min_{v_i} \sum_{\tau=k}^{k+T-1} J_i(z_i(\tau), v_i(\tau)) \quad (4a)$$

$$\text{s.t. } x_{si}(\tau) \in X_s \setminus \bigcup_{l=1}^{|\mathcal{I}|} X_{sl}(\tau), \quad 1 \in \mathcal{I}_V, l \neq i \quad (4b)$$

$$0 \leq v_i(\tau) \leq v_{\max}; \quad v_i(\tau) \in V_i \quad (4c)$$

where $x_{si}(\tau) = [S_{xi}(\tau), S_{yi}(\tau)]^T \in \mathbb{R}^2$, the position vector of vehicle i , X_s is the set describing the feasible driving area (defined by the roads), and $X_{sl}(\tau)$ is the set of states occupied by vehicle l at time τ . Here, the CAV i is required to minimize its cost J_i while satisfying the collision avoidance constraint in Eq. (4b), and velocity ($v_{\max}$ is the road speed limit) and control bounds in Eq. (4c). The cost J_i in Eq. (4a) can take any form, such as deviation from a particular goal location, energy consumption, total distance traveled, or a combination of these. For practical purposes, the constraint Eq. (4b) needs to be satisfied considering only the set of

neighboring vehicles, given by N_i . The collision avoidance constraint in Eq. (4b) is a coupled constraint, since it involves states of multiple vehicles. The constraint in Eq. (4b) is also a nonconvex constraint, which, along with the nonlinear system dynamics in Eq. (3), make the problem in Eq. (4) a nonconvex optimal control problem. Nonconvex optimal control problems cannot be solved in a computationally tractable fashion, and they generally lead to locally optimal solutions. The nonconvexity of the problem, the requirement of satisfying the coupled constraint in a distributed fashion, and the presence of uncertainty are some of the major challenges in this problem.

3 Approach

To address the major challenges, a layered control architecture is developed in this article (as shown in Fig. 1), which includes occupancy set computation of noncooperative vehicles, reference trajectory generation, distributed constraint set evaluation, data-driven linear system modeling, and robust tube-based MPC methods.

3.1 Distributionally Robust Obstacle Occupancy. An ego CAV will be surrounded by both cooperative and noncooperative vehicles. If the noncooperative vehicles have communication capabilities, we consider them to share their future trajectory with the cooperative vehicles. If they do not, we consider the CAVs to be equipped with prediction algorithms [33] that can predict the future state trajectory of the surrounding vehicles. The prediction methods will be prone to errors, so that the positions of the noncommunicating noncooperative vehicles will given by

$$x_{sj}(\tau+1) = f_{pj}(x_{sj}(\tau)) + \xi_j \quad (5)$$

where f_{pj} is the prediction model that predicts the position of the obstacles in the next time-step based on its current position and $\xi_j \in \Xi_j \subset \mathbb{R}^2$ is the prediction error. For example, maneuver-based prediction models [34,35] will provide the vehicle trajectory associated with a particular predicted maneuver, while ξ_j will capture the deviation about the trajectory. Since the true probability distribution of this error is not fully known, we consider ξ_j to belong to a set of probability distributions $\mathcal{P}$ that include all probability distributions that satisfy [36]: $(E[\xi_j] - \hat{\mu}_j)^T \Sigma_j^{-1} (E[\xi_j] - \hat{\mu}_j) \leq \gamma_j$ and $E[(\xi_j - \hat{\mu}_j)(\xi_j - \hat{\mu}_j)^T] \in \gamma_j \Sigma_j$. This set of probability distributions considers the true mean of the random variable ξ_j to lie in an ellipsoid of size γ_j centered at the estimated mean $\hat{\mu}_j$ and the centered second moment matrix to lie in a positive semidefinite cone.

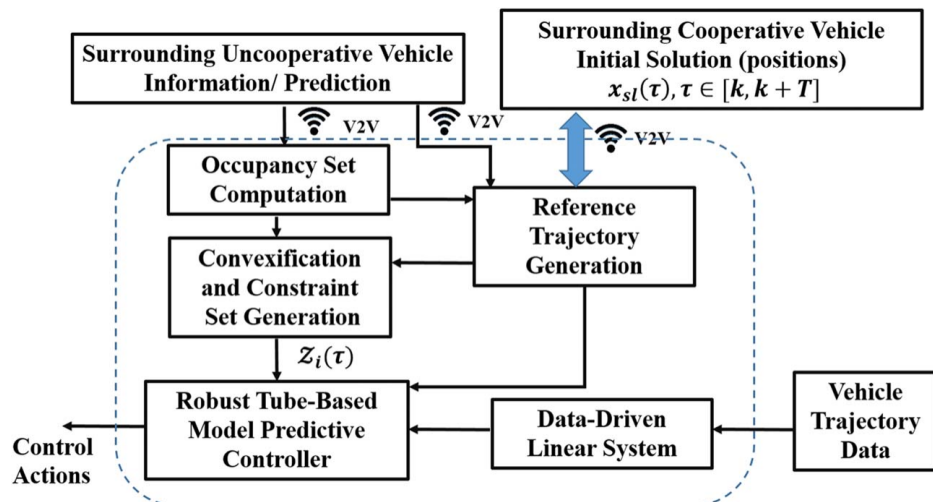


Fig. 1 The control framework

The constants ν_{j_1} and ν_{j_2} dictate the confidence on estimated mean and covariance and can be obtained from the prediction algorithm. For example, following finite bound assumption on ξ_j (will be satisfied by any real-world system) and the McDiarmid's theorem [36], the bounds can be obtained from the error samples. In this article, we consider $\hat{\mu}_j$, $\hat{\Sigma}_j$, ν_{j_1} , and ν_{j_2} to be known (obtained from the prediction algorithm).

To obtain the occupancy of the obstacles with a predefined threshold ϵ over the time horizon, we seek to find a tight bound on ξ_j , so that the probability with which the error stays within this bound $\ell_j = [\ell_{j1}, \ell_{j2}]^T$ is higher than ϵ for all probability distributions belonging to the set $\mathcal{P}$, i.e.,

$$\min_{\xi_j} P(|\xi_j| \leq \ell_j) \geq \epsilon, \quad \forall \mathcal{P} \Rightarrow \min_{\xi_j, \mathcal{P}} P(|\xi_j| \leq \ell_j) \geq \epsilon \quad (6)$$

This will give the distributionally robust future occupancy of the obstacles, since it is robust to the true distribution of the error. To ensure safety, ϵ should be chosen closer to 1, so that the obstacle occupancy over the horizon is identified with high probability (hence by avoiding those regions, the CAV will be safe with a high probability). Exploiting techniques from robust optimization [37], we can express the aforementioned probabilistic constraint as follows:

$$\min_{\xi_j, \mathcal{P}} P(|\xi_j| \leq \ell_j) \geq \epsilon \Rightarrow \min_{\xi_j} \max_{\mathcal{P}} \text{CVaR}_{1-\epsilon}(|\xi_j| - \ell_j) \leq 0 \quad (7a)$$

$$\text{CVaR}_{1-\epsilon}(|\xi_j| - \ell_j) = \min_{\beta} \beta + \frac{1}{1-\epsilon} E[(|\xi_j| - \ell_j - \beta)^+] \quad (7b)$$

where $\beta \in \mathbb{R}$, $\text{CVaR}_{1-\epsilon}$ is the conditional value at risk at level $(1 - \epsilon)$ [37], and for any function $g(x)$, $(g(x))^+ = \max(0, g(x))$. Hence, Eq. (6) can be rewritten as follows:

$$\min_{\xi_j, \beta} \beta + \max_{\mathcal{P}} \frac{1}{1-\epsilon} E[(|\xi_j| - \ell_j - \beta)^+] \leq 0 \quad (8)$$

The inner maximization problem can be expressed as a semidefinite programming problem [36], and we obtain ℓ_j , and hence the probabilistic bound on the error, by solving the problem in Eq. (8). Computation of the minimum bound ℓ_j^* is done offline. Finally, the distributionally robust occupancy sets of the obstacles is obtained by

$$X_{sj}(\tau + 1) = f_{pj}(X_{sj}(\tau)) \oplus \Xi_j \oplus X_{j, \text{shape}}(\psi_j(\tau)) \quad (9)$$

where $\Xi_j = \{\xi_j | \xi_j \leq \ell_j\}$. The set $X_{j, \text{shape}}(\psi_j(\tau))$ captures the shape of the obstacle with heading $\psi_j(\tau)$ at time τ , and $\oplus$ is Minkowski addition. For a noncooperative vehicle moving straight and not changing lanes with 18 m/s velocity, Fig. 2 shows an example of its distributionally robust probabilistic occupancy sets of different time instants over the horizon for $\nu_{j1} = 0.1$, $\nu_{j2} = 1.1$, and $\epsilon = 0.9$. Figure 2 also shows the occupancy sets for a lane change maneuver with uncertain velocity.

3.2 Reference Trajectory Generation. To solve the problem in Eq. (4), we first generate the feasible reference trajectories for the CAVs in a distributed fashion, which ensure avoidance of collision from other cooperative and noncooperative vehicles in the environment (satisfaction of the constraints in Eq. (4b)). Our reference trajectory generation approach includes the evaluation of a trajectory that avoids noncooperative vehicles, followed by modification of the trajectory (if needed) based on the cooperation among the cooperative CAVs to make sure it also avoids the reference trajectories of the other cooperative vehicles.

Given the set of positions occupied by the noncooperative vehicle $j \in I_{UC}$ to be $X_{sj}(\tau)$, the set $\bigcup_{j=1}^{|I_{UC}|} X_{sj}(\tau)$ captures the positions occupied by all the noncooperative vehicles at a given time τ . $|I_{UC}|$, which is the cardinality of the set I_{UC} , is the number of noncooperative vehicles. Then, the feasible set of states that can be occupied by the cooperative CAVs to avoid the noncooperative vehicles is given by

$$X_{s, cv}(\tau) = X_s \setminus \bigcup_{j=1}^{|I_{UC}|} X_{sj}(\tau) \quad (10)$$

To ensure safety, in addition to staying in the set $X_{s, cv}$ (ensuring avoidance of noncooperative vehicles), a CAV will have to avoid the other cooperating vehicle.

In this article, the reference trajectory of a CAV is generated by a sampling-based approach that exploits the differential flatness property of the bicycle model in Eq. (3) [19]. The differential flatness property allows us to generate the entire state trajectory from a low dimensional position trajectory. The position trajectory is expressed as follows:

$$x_{si}^{\text{ref}}(t) = \sum_{q=0}^3 \alpha_q t^q \quad (11)$$

whose coefficients α_q are obtained from the boundary conditions $[x_{si}(k), v_i(k) \cos(\psi_i(k)), v_i(k) \sin(\psi_i(k))] \in \mathbb{R}^4$ (positions and the

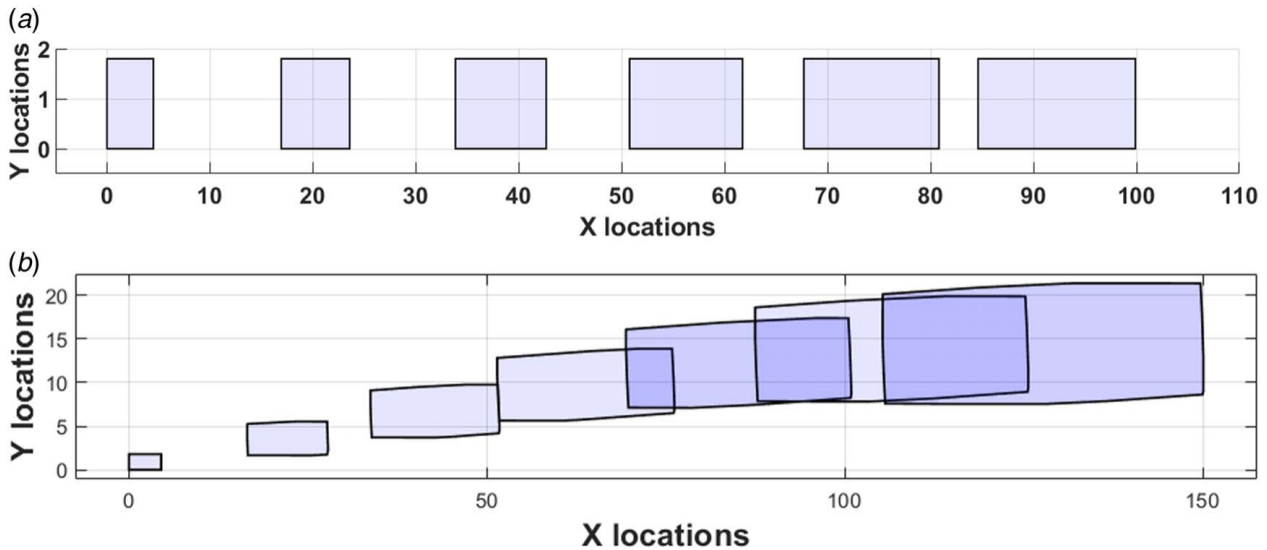


Fig. 2 Set of positions occupied by the stochastic obstacles with probability more than 0.9 for a lane following maneuver and a lane change maneuver

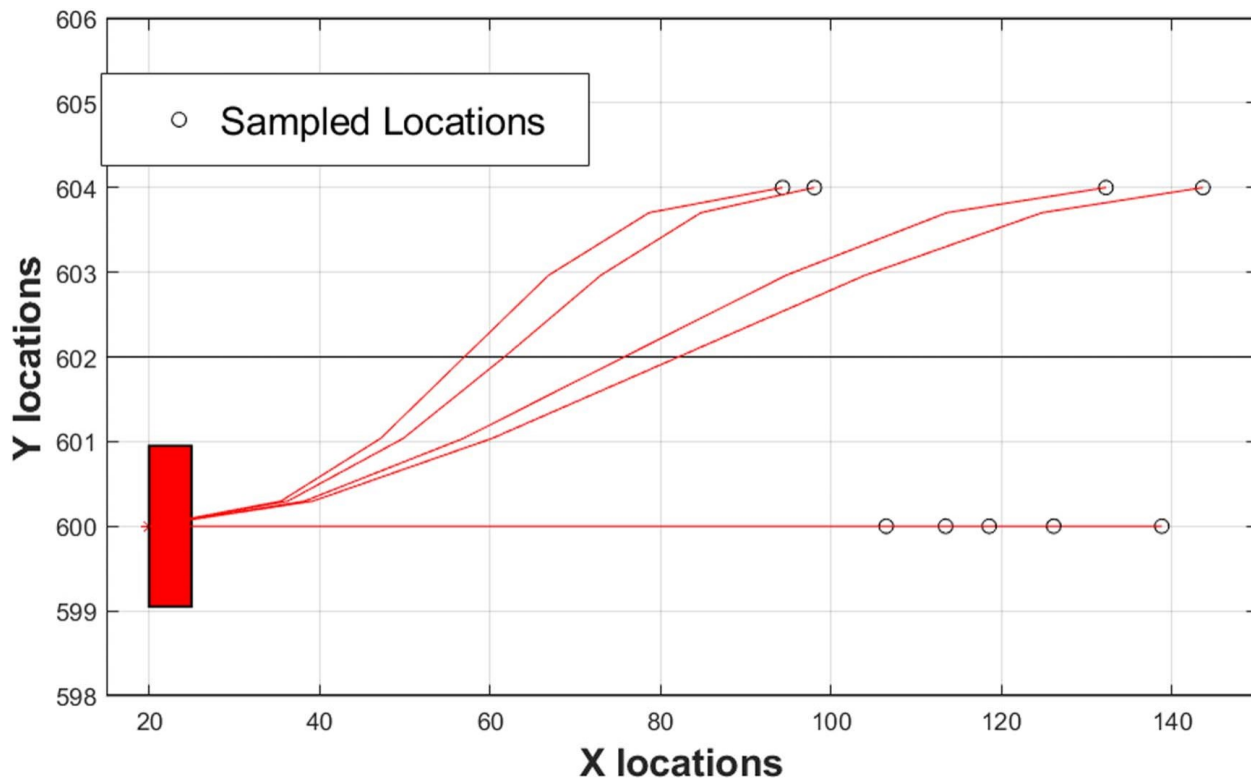


Fig. 3 Reference trajectory generation

velocities in the x and y directions at the initial time, respectively) and $[x_{si}(k+T), v_i(k+T)\cos(\psi_i(k+T)), v_i(k+T)\sin(\psi_i(k+T))]\in\mathbb{R}^4$ (positions and velocities in the x and y directions at the end of the horizon).

A number of position trajectories of the form in Eq. (11) are then obtained by sampling the center of the lanes that lie in the set $X_{s,cv}(k+T)$ and setting them as the terminal locations $x_{si}(k+T)$ at the end of the horizon. The terminal orientation is chosen to be zero $\psi_i(k+T)=0$, and the final velocity is randomly sampled from its feasible bound. The trajectories generated this way are then checked for collision avoidance from the noncooperative vehicles. Figure 3 shows multiple potential reference trajectories generated in this fashion. From these potential reference trajectories, the trajectory that maximizes the total distance travelled is chosen as the reference trajectory of the CAV.

Each cooperative CAV computes its reference position trajectory x_{si}^{ref} in this manner and shares it with its neighboring cooperative CAVs $1\leq N_i$. These position trajectories are then checked for collision. If a potential collision among the cooperative CAVs is detected (as in Fig. 4), the CAVs can choose a different reference trajectory. Since the CAVs sample the terminal location and generate multiple feasible reference trajectories, a separate reference trajectory can be readily obtained. The new reference trajectory will also require collision check with other CAVs.

We also pursue a separate approach of reference trajectory modification, where the position trajectories are updated by Δx_{si}^{ref} to obtain the new reference position trajectory $x_{si}^{ref} = x_{si}^{ref} + \Delta x_{si}^{ref}$. We adopt a heuristic approach to modify the conflicting reference trajectories, where the CAVs with conflicting trajectories choose their trajectory modification to have opposite signs (i.e., one increases and the other decreases) and satisfy the constraint $|\Delta x_{si}^{ref}| \geq L_{lf} + L_{lr}$ (to factor in the size of the vehicles). At the point of conflict in the reference trajectory, the CAV that is ahead chooses a positive Δx_{si}^{ref} , while the other chooses the negative sign. Figure 5 shows the reference trajectory after the reference trajectory modification. The CAVs can obtain their state reference trajectories x_i^{ref} from the corrected reference position trajectories x_{si}^{ref} by exploiting the differential flatness property of the bicycle model [19]. The reference trajectories generated this way significantly improves computation efficiency when compared to traditional motion planning methods [38]. This is because, rather than growing a tree, it only samples the center of the lanes and a smooth and feasible trajectory is obtained following Eq. (11). Computing multiple such reference trajectories helps the CAV, to some extent, optimize for any cost function (as in Eq. (4a)).

3.3 Constraint Set for Each Vehicle. In this section, we present how the CAVs generate their position constraint sets $X_{si}(\tau)$ based on the collision free reference trajectories x_{si}^{ref} . We

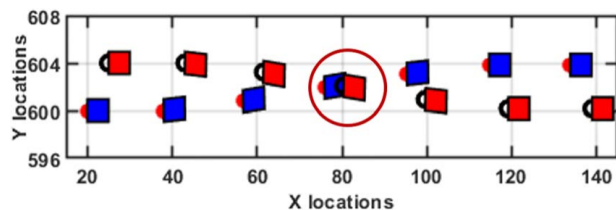


Fig. 4 Initial reference trajectories of the CAVs. Potential conflict is circled.

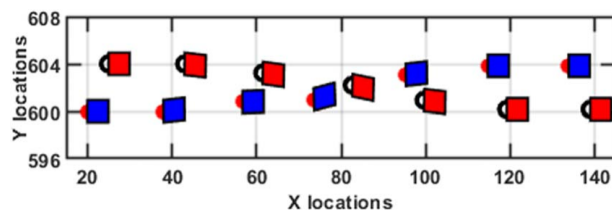


Fig. 5 Corrected reference trajectories of the CAVs

exploit the idea of distributed dynamic tessellation methods (Voronoi diagram) [39] to partition the set $X_{s,cv}(\tau)$ in a distributed manner among the cooperative CAVs to obtain their time-varying, nonoverlapping constraint sets $X_{si}(\tau)$, $X_{sj}(\tau)$, $X_{si}(\tau) \cap X_{sj}(\tau) = \emptyset$, $\forall i, j \in I_{CV}$, $i \neq j$, $\tau \in [k, k+T]$. This way, if every cooperative CAV $i \in I_{CV}$ satisfies its own position constraint $x_{si}(\tau) \in X_{si}(\tau)$, the coupling constraint in Eq. (4b) will be satisfied. Note, although the position constraint set generation method is motivated by the dynamic tessellation methods, it is different from it in terms of computation time and resultant constraint set.

Algorithm 1 describes this constraint set evaluation procedure for CAV i . Here, the CAV i receives/predicts the future positions that the noncooperative vehicles will occupy ($X_{sj}(\tau)$) and receives the nonconflicting reference trajectories $x_{sl}^{ref}(\tau)$ of cooperative CAV $l \in I_{CV}$, $l \neq i$. It then uses project and linearize method [40]—at each time instant τ over the horizon—to obtain the set of collision free positions $X_{sij}(\tau)$ from noncooperative vehicle j in its neighborhood $N_{i,uc}$. To avoid the neighboring cooperative vehicles $\forall l \in N_i$, it identifies the hyperplane (line for 2D cases) that separates $x(\tau)$ and $x_{sl}(\tau)$ and constructs the convex set $X_{si,l}(\tau)$ with the half-space (created by the hyperplane) that includes $x_{si}(\tau)$. To factor in the shape (described by the convex set $X_{i,shape}$) of the CAV, Pontryagin's difference ($\ominus$) is taken between the convex constraint set and set $X_{i,shape}$. This way, every cooperative CAV evaluates their own state constraints, which decouples the coupled constraint in Eq. (4b), and hence, every CAV can take their decision in a decentralized fashion without violating the coupling constraint.

Algorithm 1 Constraint set evaluate

Input : $x^{ref}(\tau)$, $\tau \in [k, k+T]$, $l \in N_i$ and $X_{sj}(\tau)$, $\forall j \in N_{i,uc}$
Output: $X_{si}(\tau)$, $\tau \in [k, k+T]$
Initialize $X_{si}(\tau) = X_s$;
While $\tau < k+T$ do
 While $j < N_{i,uc}$ do
 From $x_{sj}(\tau)$, use project and linearize method [40] to obtain $X_{sij}(\tau)$;
 Update $X_{si}(\tau) = X_{si}(\tau) \cap X_{sij}(\tau)$;
 $j = j + 1$;
 While $l < |N_i|$ do
 Find the hyperplane $h_{il}(\tau) = 0$ passing through the middle of $x_{si}(\tau)$ and $x_{sl}(\tau)$;
 Assign the convex set $X_{si,l}(\tau)$ as the half-space that contains $x_{si}(\tau)$;
 Update $X_{si}(\tau) = X_{si}(\tau) \cap X_{si,l}(\tau)$;
 $X_{si}(\tau) = X_{si}(\tau) \ominus X_{i,shape}$;
 $l = l + 1$;
 $\tau = \tau + 1$;

In this constraint set generation approach, the cooperative CAVs only communicate at the beginning of solving their control solution. This reduces high communication requirements associated with iterative methods. The state constraint set $Z_i(\tau) \in \mathbb{R}^4$ for each vehicle i can be obtained from the position constraint set $X_{si}(\tau)$ and the bounds on velocity and its heading direction.

3.4 Data-Driven Linear Dynamic Model. To generate the convex equivalent of the problem in Eq. (4), we utilize linear model representation of the system in Eq. (3) by using dynamic mode decomposition-based methods [27]. Here, we identify a linear discrete-time state space model of the nonlinear system (3) of the form

$$x_i(k+1) = A_i x_i(k) + B_i v_i(k) \quad (12)$$

using vehicle state and control trajectory data. We create snapshot matrices X'_i , X_i , and V_i with time series data samples from any time t to $t+N$, such that $X'_i = [x_i(t+1), x_i(t+2), \dots,$

$x_i(t+N)] \in \mathbb{R}^{4 \times N}$, $X'_i[x_i(t), x_i(t+1), \dots, x_i(t+N-1)] \in \mathbb{R}^{4 \times N}$, and $V_i = [v_i(t), v_i(t+1), \dots, v_i(t+N-1)] \in \mathbb{R}^{2 \times N}$. Given the data, A_i and B_i will need to satisfy $X'_i = A_i X_i + B_i V_i$. We use the method developed in Ref. [27] to compute A_i and B_i as follows:

$$X'_i = [A_i \ B_i] \begin{bmatrix} X_i \\ V_i \end{bmatrix} = G_i \Omega_i; \quad G_i = [A_i \ B_i] = X'_i \Omega_i^+ \quad (13)$$

where Ω_i^+ is the Moore–Penrose pseudo-inverse of Ω_i .

3.5 Robust Model Predictive Controller. The robust tube-based MPC method ensures the vehicle satisfies its convexified state constraints (as discussed in Sec. 3.3) in the presence of process noise (w_i in Eq. (1)). Here, the controller $v_i(k)$ is given by

$$v_i(k) = u_i(k) + K_i(z_i(k) - x_i(k)) \quad (14)$$

where $u_i(k)$ is the nominal controller and K_i is the feedback gain that stabilizes the closed-loop system ($A_i + B_i : K_i$). The feedback gain attenuates the effect of uncertainties and compensates for the difference between the actual and the nominal system. With the linear system dynamic model and the convexified state constraint sets, the problem in Eq. (4) can be reformulated as a convex optimal control problem:

$$\min_{\pi_i} \Delta x_i(k+T)^T P_i \Delta x_i(k+T) + \sum_{\tau=k}^{k+T-1} [u_i(\tau)^T R_{i1} u_i(\tau) \quad (15a)$$

$$+ (u_i(\tau) - u_i(\tau-1))^T R_{i2} (u_i(\tau) - u_i(\tau-1))]$$

$$\text{s.t. } x_i(\tau+1) = A_i x_i(\tau) + B_i u_i(\tau) \quad (15b)$$

$$x_i(\tau) \in Z_i(\tau) \cap S_i(\tau), \quad \forall \tau \in [k+1, k+T] \quad (15c)$$

$$u_i(\tau) \in V_i \cap K_i S_i(\tau), \quad \forall \tau \in [k+1, k+T-1] \quad (15d)$$

$$S_i(\tau+1) = (A_i + B_i K_i) S_i(\tau) + B_{wi} w_i \quad (15e)$$

where $\pi_i = [u_i(k)^T, u_i(k+1)^T, \dots, u_i(k+T-1)^T]^T$ and $\Delta x_i(k+T) = x_i(k+T) - x_i^{ref}(k+T)$. Here, the controller minimizes the control effort over the horizon (the second term), the rate of change of control (the last term), and a terminal cost (the first term) that penalizes the nominal terminal state's deviation from that of the reference trajectory. $S_i(\tau)$ in Eqs. (15c) and (15d) is the set in which the error between $z_i(\tau)$ and $x_i(\tau)$ lies at time τ [41]. Its update is given by Eq. (15e). To ensure constraint satisfaction in the presence of uncertainty w_i in Eq. (1), the nominal state and nominal control constraints are tightened, as shown in Eqs. (15c) and (15d), so that the actual state $z(\tau)$ and control $v_i(\tau)$ ensure the satisfaction of $z_i(\tau) \in Z_i$ and $v_i \in V_i$, respectively. By tightening the constraints, we are reducing the size of the feasible state and control constraint sets.

To solve the aforementioned problem in Eq. (15) in a computationally efficient fashion, we convert it into a quadratic program (QP)

$$\min \frac{1}{2} (y_i - y_i^j)^T H_i (y_i - y_i^j) \quad (16a)$$

$$\text{s.t. } M_i y_i \leq h_i; \quad C_i y_i = b_i \quad (16b)$$

$$y_i = [u_i(k), x_i(k+1), u_i(k+1), \dots, x_i(k+T-1), u_i(k+T-1), x_i(k+T)]^T \quad (16c)$$

where $y_i \in \mathbb{R}^{T(n_x+n_u)}$, $M_i \in \mathbb{R}^{(2T(n_x+n_u)-T, T(n_x+n_u))}$, $C_i \in \mathbb{R}^{(T n_x, T(n_x+n_u))}$, $b_i \in \mathbb{R}^{(T n_x)}$ and $h_i \in \mathbb{R}^{(2T(n_x+n_u)-T)}$, n_x and n_u are the state and control dimensions. The equality constraint in Eq. (16b) includes the constraint on state dynamics in Eq. (15b). H_i is a diagonal matrix and

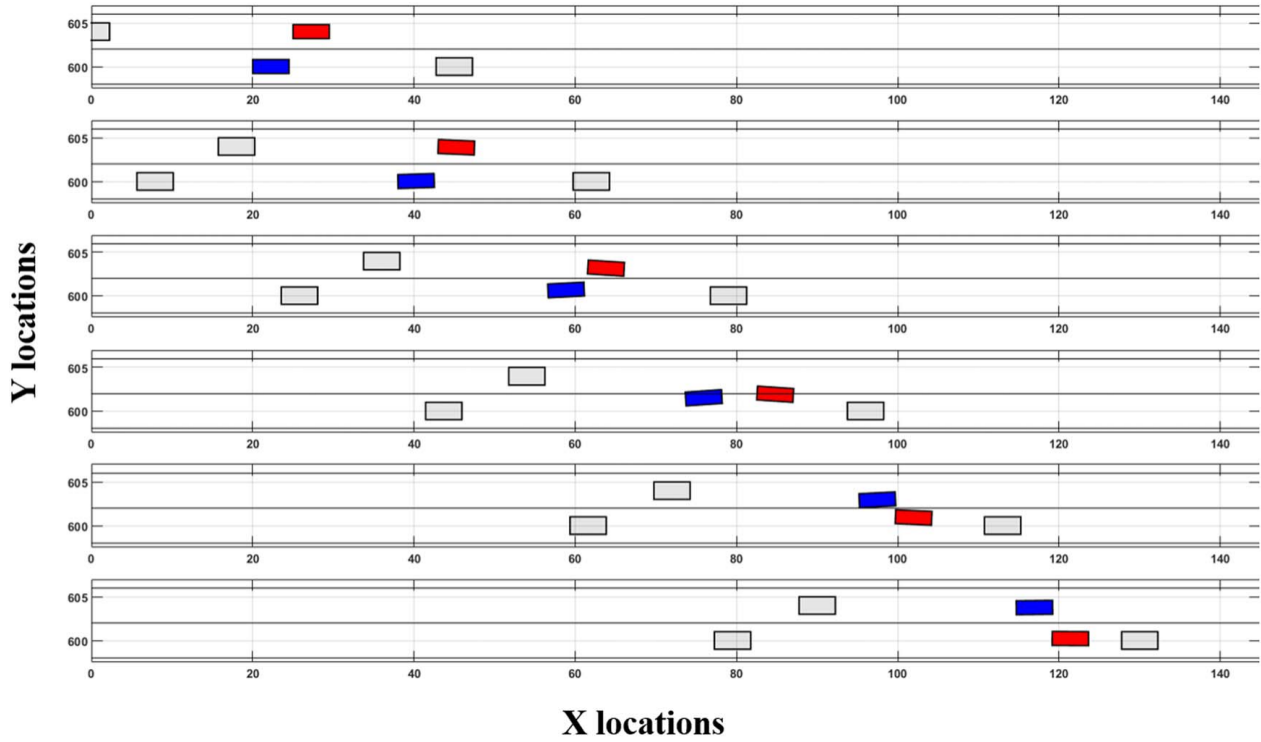


Fig. 6 Trajectories of the CAVs when robust tube-based MPC with DMD is used. The obstacles are shown in gray, while the CAVs are shown in upper lane and lower lane.

y_t^i is the target state, which includes the reference state trajectory x_1^{ref} . The Lagrangian of the problem in Eq. (16) is given by the following equation, where λ_{i1} and λ_{i1} are the Lagrange multipliers,

$$L_i(y_i, \lambda_{i2}, \lambda_{i1}) = \frac{1}{2}(y_i - y_t^i)^T H_i(y_i - y_t^i) + \lambda_{i1}(C_i y_i - b_i) + \lambda_{i2}(M_i y_i - h_i) \quad (17)$$

From the aforementioned equation, the first-order Karush–Kuhn–Tucker (KKT) optimality conditions can be obtained as follows [42]:

$$F_i(y_i, \lambda_{i1}, \lambda_{i2}) = \begin{bmatrix} H_i(y_i - y_t^i) + C_i^T \lambda_{i1} + M_i^T \lambda_{i2} \\ M_i y_i + s_{li} - h_i \\ C_i y_i - b_i \\ Y_i s_{li} \mathbf{1} \end{bmatrix} = 0 \quad (18)$$

where s_{li} are the slack variables given by $s_{li} = h_i - M_i y_i$, while Y_i and S_{li} are diagonal matrices given by $Y_i = \text{diag}(\lambda_{i2})$ and $S_{li} = \text{diag}(s_{li})$. $\mathbf{1}$ is a unity vector, and the last term in the aforementioned equation denotes the complementary slackness condition. The solution to the aforementioned problem in Eq. (18) provides the solution to the optimal control problem. To aid faster convergence, the problem in Eq. (18) is solved using the Newton step [42]. The corresponding update directions, Δy_i , $\Delta \lambda_{i1}$, and $\Delta \lambda_{i2}$, of the optimization variables are obtained from the following equation:

$$\begin{bmatrix} H_i & C_i^T & M_i^T & 0 \\ C_i & 0 & 0 & 0 \\ M_i & 0 & 0 & I \\ 0 & 0 & S_{li} & Y_i \end{bmatrix} \begin{bmatrix} \Delta y_i \\ \Delta \lambda_{i1} \\ \Delta \lambda_{i2} \\ \Delta s_{li} \end{bmatrix} = -F_i(y_i, \lambda_{i1}, \lambda_{i2}) \quad (19)$$

Finally, using a step size α_i , that keeps the Lagrange multipliers and the slack variables positive, the solution is updated at each iteration

by:

$$\begin{bmatrix} y(t+1) \\ \lambda_{i1}(t+1) \\ \lambda_{i2}(t+1) \\ s_{li}(t+1) \end{bmatrix} = \begin{bmatrix} y(t) \\ \lambda_{i1}(t) \\ \lambda_{i2}(t) \\ s_{li}(t) \end{bmatrix} + \alpha_i \begin{bmatrix} \Delta y(t) \\ \Delta \lambda_{i1}(t) \\ \Delta \lambda_{i2}(t) \\ \Delta s_{li}(t) \end{bmatrix} \quad (20)$$

Thus, at each time instant k , an iterative method is used, where the search direction is obtained from Eq. (19) and the solution is updated as in Eq. (20) till it converges, to find the optimal control solution for a time horizon T . To further expedite the process, similar to Refs. [43,44], a constant number of iterative steps are considered in this article, which provides suboptimal yet fast solutions. It was shown in Ref. [43] that the suboptimality of the solution for a very small number of iterative steps (even a single iterative step) was not high and hence the rational behind using a constant number of iterative steps.

4 Simulation Results

We present the simulation results for our proposed controller for two different scenarios: (i) a simultaneous lane change problem and (ii) a merging problem. We also compare our result with the lane change method proposed in Ref. [19].

For the simultaneous lane change problem, we consider two cooperative CAVs planning to change lane at the same time in an environment with three noncooperative vehicles (obstacles). The initial locations of the noncooperative vehicles and the cooperative CAVs are shown in the first subplot of Fig. 6. The initial locations of CAV 1 (in blue in the lower lane) and CAV 2 (in red in the upper lane) are given by $x_{s1} = [20, 600]$ and $x_{s2} = [45, 604]$, respectively. The uncertainty bounds in the CAVs are considered to be $W_i = [-1, -0.01, -0.01, -0.2] \times [1, 0.01, 0.01, 0.2]$, $i \in \{1, 2\}$. The minimum and maximum velocity of the CAVs are considered to be 0 m/s and 25 m/s, respectively. The obstacles' initial velocities are arbitrarily chosen as 18.95 m/s, 18.9 m/s, and 18 m/s, respectively, while the initial

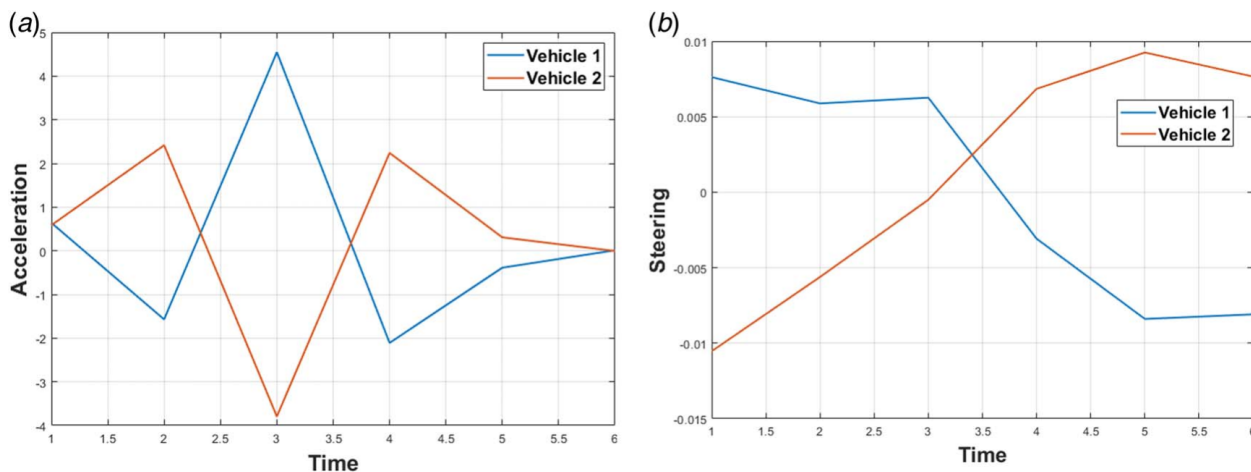


Fig. 7 Acceleration and steering of the vehicles in lane change scenario

velocities of the cooperative CAVs are chosen as 18 m/s. The CAV accelerations are constrained within -4.5 to 4.5 m/s², which ensures prevention of vehicle rollover [17]. The steering angle δ_f is constrained to be within -0.3 to 0.3 rad. The obstacles are considered to follow their lane with a velocity that lies within a finite bound of their initial velocity, but no change of lane occurs during the lane change maneuver of the CAVs. All the vehicles are considered to be of 4.5 m in length and 2 m in width (a standard sedan).

Following the layered control solution in Fig. 1, CAVs first use the reference position trajectory generation approach in Sec. 3.2 to generate their potential references, as shown in Fig. 4. Since both the CAVs plan to change lane, they only sample the center of their target lanes. The trajectories that maximize the distance traveled are chosen as their desired trajectories. However, as shown in Fig. 4, the initial reference trajectories, although avoid noncooperative vehicles, have a conflict point. The trajectories of both the CAVs are then modified by Δx_{si} , $i \in \{1, 2\}$, by the heuristic approach mentioned in Sec. 3.2 to obtain the reference position trajectories that are conflict free, as shown in Fig. 5. Although there is

no guarantee that conflict-free trajectories can be obtained in the first iteration of the modification method, in simulation, we experienced that the approach provided conflict-free trajectories within the first few iterations (mostly in the first step).

Given the reference trajectories, the state constraint sets are generated by the CAVs following Algorithm 1. We then use the data-driven DMD-based linear system representation to obtain the system and control matrices of the system, i.e., A_i and B_i . To consider the vehicle process noise, we exploit robust tube-based MPC method and solve the problem (15). The feedback gain K_i in Eq. (14) is computed as the linear quadratic regulator gain, which stabilizes the closed-loop system. A finite time horizon of 6 s with $\Delta t = 1$ s sampling time is considered, i.e., $T = 6$. In a computer with 3.5 GHz Intel^(R) Xeon^(R) processor and 32 GB RAM, the MPC problem required ≈ 0.11 s (including reference trajectory generation, constraint set evaluation, and solution of the tube-based MPC in a computationally efficient fashion), which is comparable to the time budget available for onboard implementation. It may be noted that no effort of optimizing the code to improve computation time has been done.

Figure 6 shows the successful simultaneous lane change maneuvers performed by the cooperative CAVs in the presence of obstacles. Figure 7 shows the control actions of the two CAVs. The solution obtained from the robust tube-based method with DMD-based linear model leads to smooth vehicle trajectories while being computationally efficient at the same time. Furthermore, these solutions ensure satisfaction of the system constraints in the presence of affine uncertainties.

We compare our work with the distributed lane change algorithm described in Ref. [19] that uses buffer Voronoi cells (BVCs) to constraint vehicle states and buffer input cells (BIC) to constraint the inputs to ensure collision avoidance. The reference trajectory in Ref. [19] is generated using Eq. (11). The approach in Ref. [19] uses Voronoi diagram to partition the feasible position space at a given time. It also constraints the vehicle input to the BIC (computed based on the current BVC) to ensure the vehicle stays

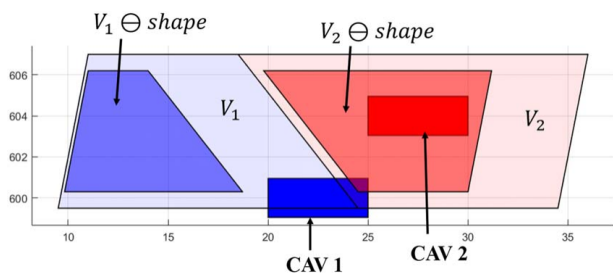


Fig. 8 Initial positions of CAV 1 and CAV 2. The light shaded regions are their Voronoi cells. After taking into account of the vehicle shapes, the constraint sets are given by the darker shaded regions.

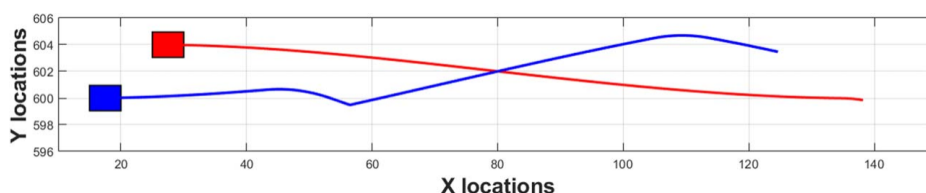


Fig. 9 Distributed lane change with BIC when the initial CAV 1 position is $x_{s1} = [15, 600]$

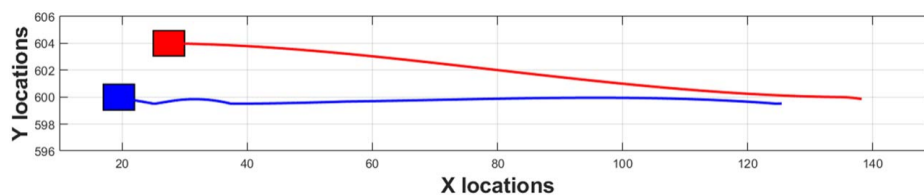


Fig. 10 Distributed lane change with BIC when the initial CAV 1 position is $x_{s1} = [17, 600]$

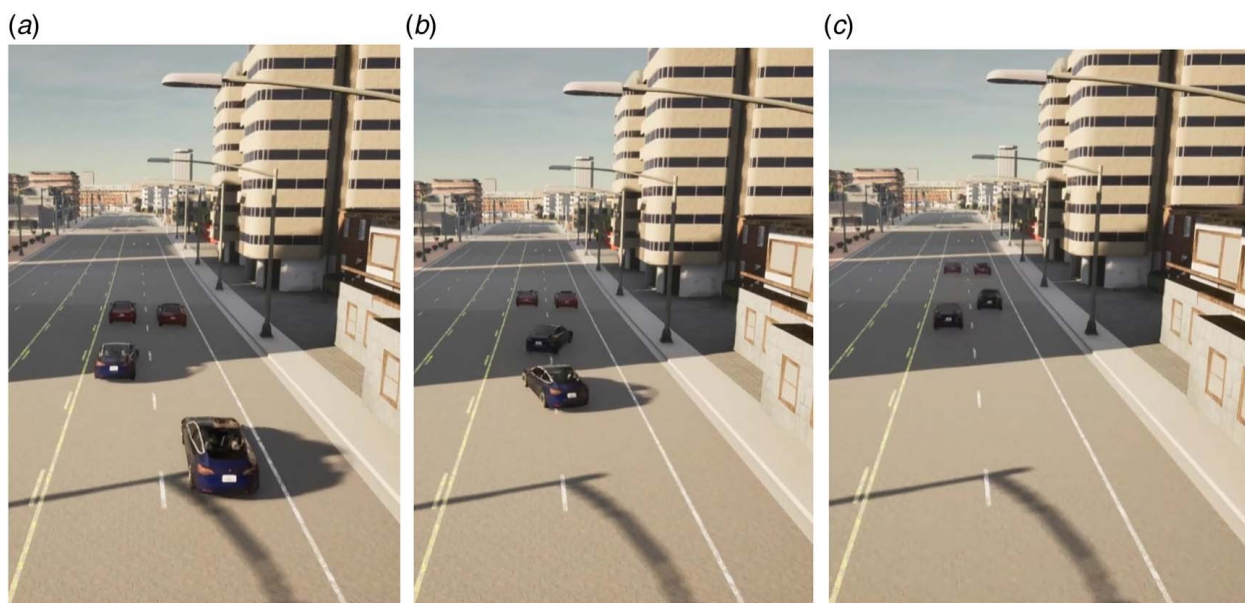


Fig. 11 Different instances in the lane change simulation in CARLA. The CAVs are shown in black. (a) initial locations; (b) intermediate stage; and (c) final locations.

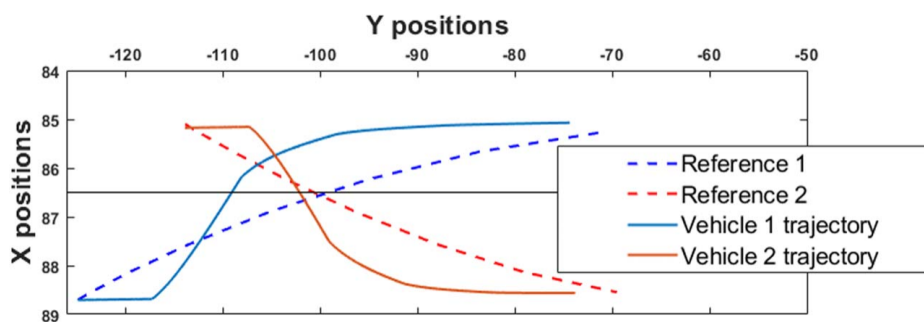


Fig. 12 Actual and reference trajectories in the lane changing scenario in Fig. 11

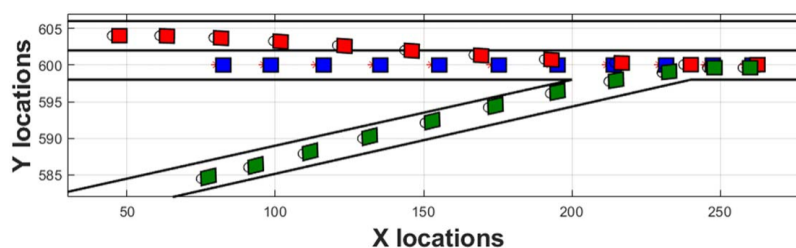


Fig. 13 Initial reference trajectories in a merging scenario

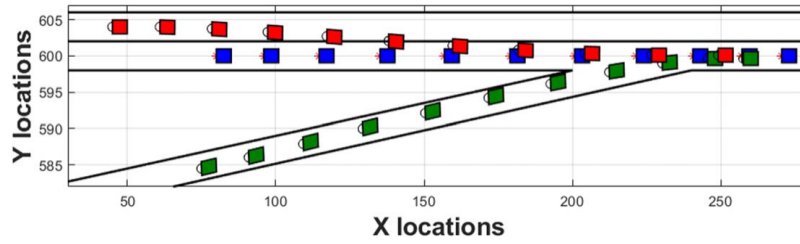


Fig. 14 Corrected reference trajectories in a merging scenario

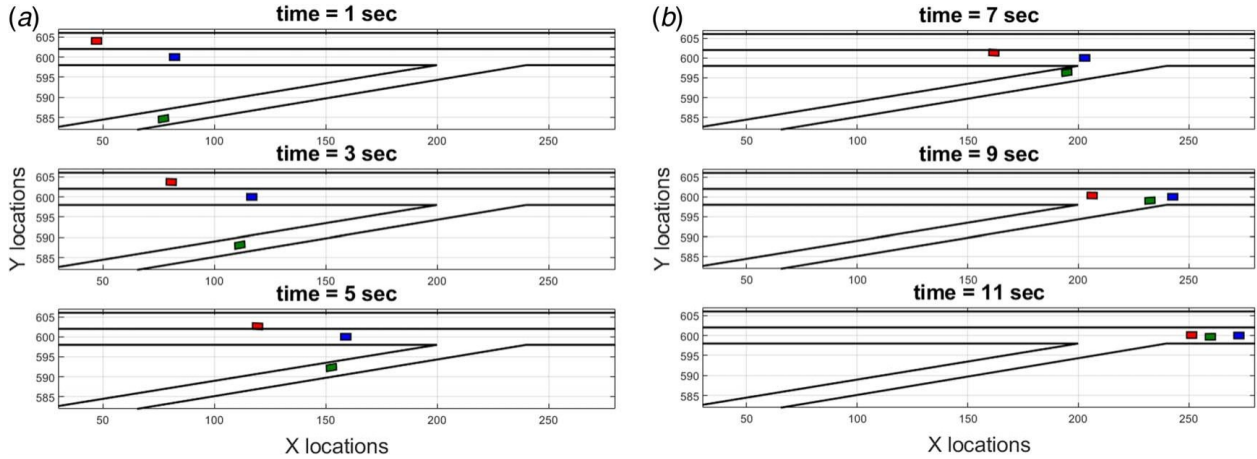


Fig. 15 Vehicle positions in a merging scenario: (a) times 1, 3, and 5 secs and (b) times 7, 9, and 11 secs

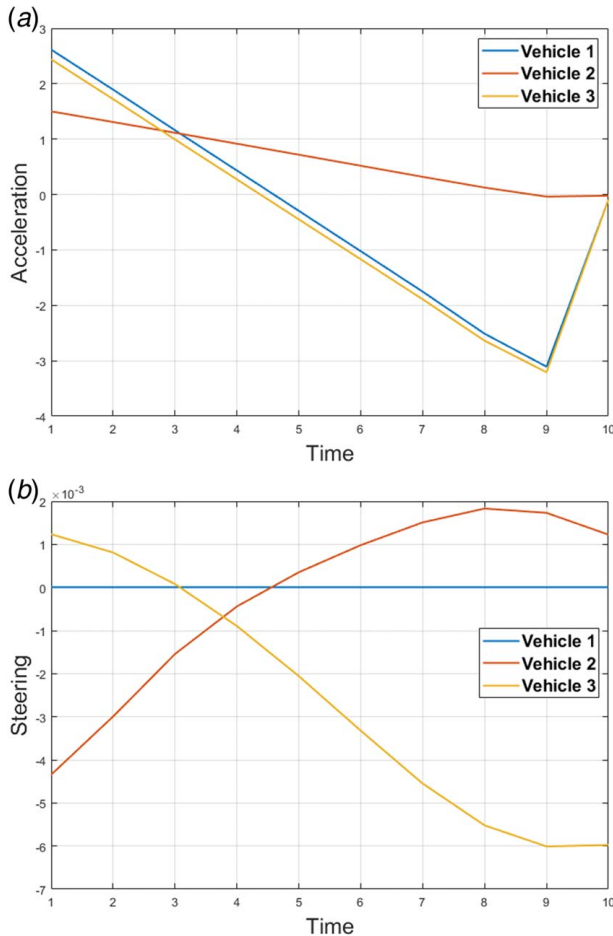


Fig. 16 Acceleration and steering of the vehicles: (a) times 1, 3, and 5 secs and (b) times 7, 9, and 11 secs

within the current BVC in the next time-step. To account for the vehicle shape, Pontryagin difference between the BVC and the set defining the vehicle shape is taken. This is one of the critical difference between [19] and our work, that we solve the problem over a horizon and generate time-varying state constraints; and choose our current control solution to keep the vehicle in the constraint set in the next time instant, which is different from the constraint set in the current time instant. The work in Ref. [19] also requires Taylor series approximation-based linearization, where the nonlinear dynamics is linearized around the current vehicle operating conditions.

Figure 8 shows initial positions of the CAVs (same as in Fig. 4), $x_{s1} = [20, 600]$ and $x_{s2} = [45, 604]$, the Voronoi cells (V_i) of each CAV (in light shade), and the Voronoi cell after taking into account the vehicle shape (in slightly darker shade). It is shown in Fig. 8 that the initial BVC of CAV 1 renders its position to be infeasible. The vehicle from that point cannot track its reference or even satisfy the system constraints, since the constraint set enforced by BIC keeps on choosing the lower bound of its control to bring the vehicle to the BVC, which is not possible considering vehicle's control constraints.

If we change the CAV 1 initial position to $x_{s1} = [15, 600]$, both the CAVs are able to change lanes as shown Fig. 9. However, as shown in Fig. 9, the CAV 1 tends to move back to its lane in the middle of lane change due to CAV 2. If the CAV 1 starts from position to $x_{s1} = [17, 600]$, Fig. 10 shows that it is unable to track the reference (and hence reach the other lane) since the control and state constraints due to CAV 2 positions prohibit it to do so. Apart from these, we also saw that the BIC-based control constraints can violate the vehicle's constraints—we saturated the control solution during the comparison. Also, there are cases of model mismatch between the linearized dynamics and the nonlinear dynamics. However, despite being conservative, the approach in Ref. [19] is highly computationally efficient and requires less computation time than our proposed method.

We also simulated our proposed method in CARLA [45], which is an open-source simulator for autonomous driving research. To simulate in CARLA, our solution was provided to

the CAVs to track. The road coordinates were defined using the inbuilt map in CARLA 0.9.X. Our control solutions were then passed on as reference trajectories to CARLA, where it was tracked using the CARLA object "VehiclePIDController()" for each vehicle. Figure 11 shows how the two CAVs (black vehicles) simultaneously change lanes. Figure 12 shows the trajectories from our proposed method that the vehicles should follow, and the vehicles' actual trajectories, which resulted in less than 1 m lateral error.

In the second scenario, we consider a vehicle merging case as shown in Figs. 13 and 14, where the green CAV (merging lane) intends to merge, while the red CAV (upper lane) intends to change lane as well. In this scenario, we consider three cooperative CAVs, and Fig. 13 shows their conflicting reference trajectories. Their modified feasible trajectories according to Sec. 3.2 are shown in Fig. 14. Here, only the red (upper lane) and blue (lower lane) CAVs modify their reference trajectories to help the green vehicle (in the merging lane) successfully merge in the highway. Similar to the simultaneous lane change case, the state constraints are obtained based on the reference trajectories and the process noise in CAVs are addressed by using a robust tube-based MPC approach. The resulting vehicle trajectories while following the reference trajectories are shown in Fig. 15, and the control actions are shown in Fig. 16.

The advantages of the proposed controller include computational tractability, ensured collision avoidance in the presence of process noise and obstacle prediction error, and simultaneous decision-making (lane change maneuvers) among the CAVs. The simulation results validate these. However, the proposed approach does not include multiplicative errors, uncertainty in communication, and faulty communications or cyber-attacks. These topics are out of scope of this article, but will be pursued in the future work.

5 Conclusion

In this article, we present a computationally efficient layered control framework for CAVs to ensure their safe operation in the presence of uncertainty. The distributed constraint set generation approach decouples the coupled constraints, which allows each CAV to take its own decisions. The distributed coordinated approach to generate their reference trajectories helps the CAVs to take nonconservative actions while ensuring collision avoidance. The robust tube-based MPC approach ensures constraint satisfaction in the presence of process noise. Future work includes incorporation of surrounding vehicle prediction methods, use of detailed vehicle dynamic models, optimization-based reference trajectory modification, consideration of multiplicative errors, and cyber-secure navigation.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

References

- [1] Najm, W. G., Koopmann, J., Smith, J. D., and Brewer, J., 2010, "Frequency of Target Crashes for Intellidrive Safety Systems," National Highway Traffic Safety Administration, WA, Technical Report DOT HS 811 381.
- [2] <https://www.fars.nhtsa.dot.gov/Main/index.aspx>
- [3] Guanetti, J., Kim, Y., and Borrelli, F., 2018, "Control of Connected and Automated Vehicles: State of the Art and Future Challenges," *Annu. Rev. Control*, 45, pp. 18–40.
- [4] Bageshwar, V. L., Garrard, W. L., and Rajamani, R., 2004, "Model Predictive Control of Transitional Maneuvers for Adaptive Cruise Control Vehicles," *IEEE Trans. Vehicular Technol.*, 53(5), pp. 1573–1585.
- [5] Anderson, S. J., Peters, S. C., Pilutti, T. E., and Iagnemma, K., 2010, "An Optimal-Control-Based Framework for Trajectory Planning, Threat Assessment, and Semi-Autonomous Control of Passenger Vehicles in Hazard Avoidance Scenarios," *Int. J. Vehicle Auton. Syst.*, 8(2), pp. 190–216.
- [6] Bevan, G. P., Gollee, H., and O'reilly, J., 2010, "Trajectory Generation for Road Vehicle Obstacle Avoidance Using Convex Optimization," *Proc. Inst. Mech. Eng., Part D: J. Automobile Eng.*, 224(4), pp. 455–473.
- [7] Gray, A., Gao, Y., Lin, T., Hedrick, J. K., Tseng, H. E., and Borrelli, F., 2012, "Predictive Control for Agile Semi-Autonomous Ground Vehicles Using Motion Primitives," American Control Conference, Montreal, Canada, June 27–29, pp. 4239–4244.
- [8] Liu, K., Gong, J., Kurt, A., Chen, H., and Ozguner, U., 2018, "Dynamic Modeling and Control of High-Speed Automated Vehicles for Lane Change Maneuver," *IEEE Trans. Intell. Vehicles*, 3(3), pp. 329–339.
- [9] Nilsson, J., Brännström, M., Coelingh, E., and Fredriksson, J., 2016, "Lane Change Maneuvers for Automated Vehicles," *IEEE Trans. Intell. Transpor. Syst.*, 18(5), pp. 1087–1096.
- [10] Nilsson, J., Brännström, M., Fredriksson, J., and Coelingh, E., 2016, "Longitudinal and Lateral Control for Automated Yielding Lane-Changes," *IEEE Trans. Intell. Transpor. Syst.*, 17(5), pp. 1404–1414.
- [11] Butakov, V. A., and Ioannou, P., 2014, "Personalized Driver/Vehicle Lane Change Models for Adas," *IEEE Trans. Vehicular Technol.*, 64(10), pp. 4422–4431.
- [12] Nie, J., Zhang, J., Ding, W., Wan, X., Chen, X., and Ran, B., 2016, "Decentralized Cooperative Lane-Changing Decision-Making for Connected Autonomous Vehicles," *IEEE Access*, 4, pp. 9413–9420.
- [13] Awal, T., Murshed, M., and Ali, M., 2015, "An Efficient Cooperative Lane-Changing Algorithm for Sensor- and Communication-Enabled Automated Vehicles," Intelligent Vehicles Symposium, Seoul, South Korea, June 28–July 1, pp. 1328–1333.
- [14] Lombard, A., Perronnet, F., Abbas-Turki, A., and El Moudni, A., 2017, "On the Cooperative Automatic Lane Change: Speed Synchronization and Automatic 'courtesy'," Design, Automation & Test in Europe Conference & Exhibition, Lausanne, Switzerland, Mar. 27–31, pp. 1655–1658.
- [15] Sun, K., Zhao, X., and Wu, X., 2021, "A Cooperative Lane Change Model for Connected and Autonomous Vehicles on Two Lanes Highway by Considering the Traffic Efficiency on Both Lanes," *Transpor. Res. Interdiscip. Perspect.*, 9, p. 100310.
- [16] Karimi, S., and Vahidi, A., 2020, "Receding Horizon Motion Planning for Automated Lane Change and Merge Using Monte Carlo Tree Search and Level-K Game Theory," Proceedings of American Control Conference, Denver, CO, July 1–3, pp. 1223–1228.
- [17] Shao, K., Zheng, J., Deng, B., Huang, K., and Zhao, H., 2020, "Active Steering Control for Vehicle Rollover Risk Reduction Based on Slip Angle Estimation," *IET Cyber-Syst. Rob.*, 2(3), pp. 132–139.
- [18] Zhou, D., Wang, Z., Bandyopadhyay, S., and Schwager, M., 2017, "Fast, On-line Collision Avoidance for Dynamic Vehicles Using Buffered Voronoi Cells," *IEEE Rob. Autom. Lett.*, 2(2), pp. 1047–1054.
- [19] Wang, M., Wang, Z., Paudel, S., and Schwager, M., 2018, "Safe Distributed Lane Change Maneuvers for Multiple Autonomous Vehicles Using Buffered Input Cells," IEEE International Conference on Robotics and Automation, Brisbane, Australia, May 21–25, pp. 1–7.
- [20] Wang, M., and Schwager, M., 2019, "Distributed Collision Avoidance of Multiple Robots With Probabilistic Buffered Voronoi Cells," International Symposium on Multi-Robot and Multi-Agent Systems, New Brunswick, NJ, Aug. 22–23, pp. 169–175.
- [21] Schmerling, E., Leung, K., Vollprecht, W., and Pavone, M., 2018, "Multimodal Probabilistic Model-Based Planning for Human-Robot Interaction," 2018 IEEE International Conference on Robotics and Automation, Brisbane, Australia, May 21–25, pp. 1–9.
- [22] Spica, R., Cristofalo, E., Wang, Z., Montijano, E., and Schwager, M., 2020, "A Real-Time Game Theoretic Planner for Autonomous Two-Player Drone Racing," *IEEE Trans. Robot.*, 36, pp. 1389–1403.
- [23] Wu, A., and How, J. P., 2012, "Guaranteed Infinite Horizon Avoidance of Unpredictable, Dynamically Constrained Obstacles," *Auton. Rob.*, 32(3), pp. 227–242.
- [24] Chiang, H. T. L., HomChaudhuri, B., Vinod, A., Oishi, M., and Tapia, L., 2017, "Dynamic Risk Tolerance: Motion Planning by Balancing Short-Term and Long-Term Stochastic Dynamic Predictions," International Conference on Robotics and Automation, Marina Bay Sands, Singapore, May 29–June 3, pp. 3762–3769.
- [25] Pereira, A., and Althoff, M., 2017, "Overapproximative Human Arm Occupancy Prediction for Collision Avoidance," *IEEE Trans. Autom. Sci. Eng.*, 15(2), pp. 818–831.
- [26] LaValle, S. M., 2006, *Planning Algorithms*, Cambridge University Press, Cambridge, UK.
- [27] Proctor, J. L., Brunton, S. L., and Kutz, J. N., 2014, "Dynamic Mode Decomposition With Control," *SIAM J. Appl. Dyn. Sys.*, 15(1), pp. 142–161.
- [28] Folkstad, C., Pastor, D., Mezic, I., Mohr, R., Fonoberova, M., and Burdick, J., 2020, "Extended Dynamic Mode Decomposition With Learned Koopman Eigenfunctions for Prediction and Control," American Control Conference, Denver, CO, July 1–3, pp. 3906–3913.

- [29] Dai, L., Xia, Y., Gao, Y., and Cannon, M., 2016, "Distributed Stochastic MPC of Linear Systems With Additive Uncertainty and Coupled Probabilistic Constraints," *IEEE Trans. Automat. Contr.*, 62(7), pp. 3474–3481.
- [30] Sotoudeh, S. M., and HomChaudhuri, B., 2019, "Ensured Collision Avoidance Over a Finite Time Horizon for Autonomous Vehicles in Presence of Uncertainty," *Dynamic Systems and Control Conference*, Park City, UT, Oct. 8–11, Vol. 59148, American Society of Mechanical Engineers, p. V001T01A003.
- [31] HomChaudhuri, B., 2019, "Distributionally Robust Stochastic Model Predictive Control for Collision Avoidance," *Dynamic Systems and Control Conference*, Park City, UT, Oct. 8–11, American Society of Mechanical Engineers, pp. 1–8.
- [32] Kong, J., Pfeiffer, M., Schildbach, G., and Borrelli, F., 2015, "Kinematic and Dynamic Vehicle Models for Autonomous Driving Control Design," *Intelligent Vehicles Symposium*, Seoul, South Korea, June 28–July 1, pp. 1094–1099.
- [33] Gipps, P., 1981, "A Behavioural Car-Following Model for Computer Simulation," *Transp. Res. Part B Methodol.*, 15(2), pp. 105–111.
- [34] Deo, N., and Trivedi, M. M., 2018, "Multi-Modal Trajectory Prediction of Surrounding Vehicles With Maneuver Based LSTMS," *Intelligent Vehicles Symposium*, Changshu, China, June 26–30, pp. 1179–1184.
- [35] Aswani, A., Gonzalez, H., Sastry, S. S., and Tomlin, C., 2013, "Provably Safe and Robust Learning-Based Model Predictive Control," *Automatica*, 49(5), pp. 1216–1226.
- [36] Delage, E., and Ye, Y., 2010, "Distributionally Robust Optimization Under Moment Uncertainty With Application to Data-Driven Problems," *Oper. Res.*, 58(3), pp. 595–612.
- [37] Zymler, S., Kuhn, D., and Rustem, B., 2013, "Distributionally Robust Joint Chance Constraints With Second-Order Moment Information," *Math. Program.*, 137(1–2), pp. 167–198.
- [38] Paden, B., Cáp, M., Yong, S. Z., Yershov, D., and Frazzoli, E., 2016, "A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles," *IEEE Trans. Intell. Vehicles*, 1(1), pp. 33–55.
- [39] Pimenta, L. C., Kumar, V., Mesquita, R. C., and Pereira, G. A., 2008, "Sensing and Coverage for a Network of Heterogeneous Robots," *IEEE Conference on Decision and Control*, Cancun, Mexico, Dec. 9–11, pp. 3947–3952.
- [40] Mao, Y., Szmuk, M., and Açikmese, B., 2016, "Successive Convexification of Non-Convex Optimal Control Problems and its Convergence Properties," *IEEE Conference on Decision and Control*, Las Vegas, NV, Dec. 12–14, pp. 3636–3641.
- [41] Limon, D., Alvarado, I., Alamo, T., and Camacho, E., 2010, "Robust Tube-Based MPC for Tracking of Constrained Linear Systems With Additive Disturbances," *J. Process. Control.*, 20(3), pp. 248–260.
- [42] Boyd, S., and Vandenberghe, L., 2004, *Convex Optimization*, Cambridge University Press, Cambridge, UK.
- [43] Wang, Y., and Boyd, S., 2009, "Fast Model Predictive Control Using Online Optimization," *IEEE Trans. Control Syst. Technol.*, 18(2), pp. 267–278.
- [44] HomChaudhuri, B., Vahidi, A., and Pisu, P., 2017, "Fast Model Predictive Control-Based Fuel Efficient Control Strategy for a Group of Connected Vehicles in Urban Road Conditions," *IEEE Trans. Control Syst. Technol.*, 25(2), pp. 760–767.
- [45] Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V., 2017, "CARLA: An Open Urban Driving Simulator," *Proceedings of the 1st Annual Conference on Robot Learning*, Mountain View, CA, Nov. 13–15, pp. 1–16.