

Distributed Model Predictive Control of Connected Multi-Vehicle Systems at Unsignalized Intersections

Qiang Ge

School of Vehicle and Mobility
Tsinghua University
Beijing, China
gq17@mails.tsinghua.edu.cn

Guillaume Sartoretti

Department of Mechanical Engineering
National University of Singapore
Singapore
guillaume.sartoretti@nus.edu.sg

Jingliang Duan

School of Mechanical Engineering
University of Science and Technology Beijing
Beijing, China
duanj115@163.com

Shengbo Eben Li

School of Vehicle and Mobility
Tsinghua University
Beijing, China
lishbo@tsinghua.edu.cn

Yuming Yin

School of Mechanical Engineering
Zhejiang University of Technology
Hangzhou, China
yinyuming@zjut.edu.cn

Sifa Zheng*

School of Vehicle and Mobility
Tsinghua University
Beijing, China
zsf@tsinghua.edu.cn

Abstract—This paper proposes a four-component framework for connected vehicles in general scenarios, which consists of: 1) an information flow topology, 2) cooperative controllers, 3) node dynamics, and 4) formation geometry, extending original concepts from vehicular platoon control. In this framework, we present a real-time planning and distributed cooperative control method, which is scenario-agnostic and scalable to larger groups of vehicles. Specifically, we first develop a real-time receding-horizon planning approach, which improves plan consistency by reasoning about and reusing the previous trajectory. Our distributed model predictive controllers track the resulting reference trajectories and avoid collisions by allowing neighboring vehicles to exchange their intentions. In contrast to many existing methods, our approach to decentralized cooperation does not need to assume fixed paths for vehicles, nor any priority among them. We present simulation results with up to 8 heterogeneous vehicles, showing how our framework allows agents to establish efficient interactions to simultaneously traverse an unsignalized intersection smoothly and safely. Video results are available at <https://youtu.be/Q3KjKuIquAo> and <https://youtu.be/x1zS4ynaW-s>.

Keywords—Intelligent connected vehicles, intention sharing, real-time planning, distributed control

I. INTRODUCTION

Li and Zheng *et al.* recently proposed a framework for vehicular platoon control, composed of four key components: information flow topology, distributed controllers, node dynamics, and formation geometry [1]. With the increasing proportion of intelligent and connected vehicles (ICVs) in modern traffic flows, inter-vehicle cooperation has been growing from a relatively closed intra-platoon problem to a much more intricate and complex one that involves vehicle following, merging, as well as conflict resolution at intersections, ramps, or roundabouts. This work extends the original four-component framework to accommodate general multi-connected vehicle systems, by incorporating new relevant features unseen in

traditional platoons, such as general trajectory planning scenarios, distinct cooperative controllers, mobility in the full 2D plane, as well as dynamic topologies. Our framework relies on receding-horizon planning with improved plan consistency, as well as distributed model predictive control (DMPC) on the full 2D plane (i.e., without the assumption of *fixed paths* for vehicles) with local communications to allow vehicles to exchange their intentions towards priority-free cooperation.

Trajectory planning aims at generating a sequence of discrete waypoints, which can then be tracked by a low-level controller to allow vehicles to reach their target destination. Existing methods are either based on optimization [2], search [3], or sampling [4]. While optimization based methods can yield smooth and safe trajectories, they usually suffers from high computation times, especially under complex constraints. The other two types of method are relatively more time-efficient, but often at the cost of path resolution. Moreover, sampling or search-based methods performs worse in the presence of dynamic obstacles, which are common for vehicle driving tasks. To improve planning efficiency, Guan *et al.* recently proposed the integrated decision and control framework (IDC) for automated vehicles, in which path planning considers only static traffic elements like road boundaries and traffic lights [5]. There, the key insight is to reduce the complexity of trajectory planning, to then only consider other vehicles as moving obstacles during the control phase. This allows real-time planning, but sometimes at the cost of oscillations of the closed-loop trajectory tracking. Inspired by this idea, we introduce a receding-horizon trajectory planning method, which relies on general feature points that can be easily extracted from the environment, to output smooth trajectories at low computational cost. We further explicitly reason about plan consistency using the previous vehicle's trajectory, to prevent oscillations and improve replanning efficiency.

Our trajectory planning approach then requires a low-level controller that can explicitly handle state constraints (such as

*Corresponding Author

This study was supported by National Key R&D Program of China with 2020YFB1600200 and Tsinghua-Toyota Joint Research Fund.

the ones needed for online collision avoidance among moving vehicles), for which the natural choice is model-predictive control (MPC). That is, with our choice of trajectory planner and problem statement, we cast multi-vehicle cooperation as a distributed MPC (DMPC) problem. Many related works focus on intersection control using DMPC [6]–[10], but assume vehicles drive on predefined fixed paths. This assumption helps them formulate a two-level optimization problem – upper-level optimal scheduling and low-level local longitudinal control – but also introduces three main drawbacks. First, driving on fixed paths requires the assignment of priorities among vehicles at each conflict point, which is hard to optimize for in a distributed manner. When the traffic is sparse (i.e., when each conflict point can be decoupled from the whole conflict graph and become a one-variable integer decision problem), priorities can be easy to determine with an implicit “first come, first serve” rule. However, efficient cooperation is most critical in high-traffic scenarios. Second, the available road space is not fully utilized under the fixed-path assumption, where steering away from the fixed path is not allowed, even when it would be a better option. That is, the vehicles’ action space is unnecessarily limited, resulting in suboptimal solutions. Third, the scheduling strategy is coupled with the lanes topology and the conflict points, thus limiting the generalizability of these approaches to different tasks. For example, when approaching a new scenario, e.g., an irregular asymmetric intersection, the upper-level coordination algorithm often has to be redesigned.

Inspired by recent works in which agents share their intentions for better cooperation [11], [12], this work relies on local interactions among vehicles to share each others’ predicted future states, which are then used by their DMPC controllers to avoid collisions. In doing so, the receding-horizon nature of DMPC helps decompose a difficult, global optimization problem in time, thus allowing vehicles to iteratively reach collaboration without the need for fixed-path assumptions or priorities. Specifically, our DPMC controller is defined over acceleration and steering angle, thus allowing vehicles to fully utilize their 2D environment, as evidenced by our simulation results, where multiple vehicles are tasked to simultaneously traverse a large, unsignalized intersection while safely avoiding collisions. Finally, we note that our general framework is scenario-agnostic, since it only relies on general features of the environment for planning and on intention-sharing for collaborative DPMC, and thus should generalize to new scenarios or tasks without significant changes.

II. FOUR-COMPONENT FRAMEWORK

The classic four-component framework of vehicular platoon control [1] is: (1) node dynamics, which describes the behavior of each involved vehicle; (2) information flow topology, which defines how the nodes exchange information with each other; (3) distributed controller, which implements the feedback control only using neighboring information; (4) formation geometry, which indicates the desired inter-vehicle distance when platooning. We give the new definitions and explain the extensions of the four-component framework below:

- 1) **Information Flow Topology.** The information exchange relationship among vehicles. In a platoon, topology is usually fixed; in general connected multi-vehicle systems, however, switching topology with varying number of nodes becomes much more significant. Moreover, whether the topology is fixed or not, it could be centralized, distributed or hybrid. Another extension lies in that the system may include “dumb” nodes, corresponding to vehicles without communication capabilities.
- 2) **Cooperative Controller.** The control law takes both local perception and communication information as input. Control can be executed by either master node/slave node under a centralized network, or any node under a distributed network. In the case of centralized cooperation, the slave controller can be a local feedback tracker trying to follow the instructions from a central node. In the case of a distributed network, the controller could interact with other peer neighbors by exchanging their intentions in the prediction time horizon. Specially, a dumb node such as an unconnected vehicle could only be observed and predicted by cooperative controllers, and probably should be treated with a more conservative policy.
- 3) **Node Dynamics.** Input-output property of the vehicle. While platoons usually consider homogeneous dynamics, multi-ICV systems often consist of vehicles with heterogeneous dynamics. Furthermore, node behaviors in multi-ICV systems include both longitudinal and lateral dynamics, thus further increasing their complexity. This change requires nodes to show higher intelligence and accuracy during perception, decision, control, and execution.
- 4) **Formation Geometry.** Local planning method and environmental context information. In a platoon, the geometry is usually a single scalar constant describing the distancing policy. When considering general scenarios and tasks, the geometry may consist of multiple time-variant constraints: inter-vehicle safety distances, traffic light phases, speed limits, and varying lane topologies. Centralized control usually relies on known/fixed scenarios, and thus cannot handle general tasks like traveling in an asymmetric intersection without a priori knowledge about it. However, a distributed controller can consider only its local reference trajectory (which could be determined by feature points encoding interesting part of the environmental information) and surrounding obstacles, and does not seek any global understanding of the (bird’s-eye view) road topology. In other words, once the navigation direction is determined, distributed control may be decoupled from the structure of the surroundings.

The four components are independent concepts. From the perspective of controller design, the other three components take different effects. Information flow topology influences the upstream input of the controller, node dynamics describe the

TABLE I
CATEGORIZATION OF CONNECTED MULTI-VEHICLE COOPERATION

| Information Flow Topology | | | | | |
|--|-----------------------------------|----------------------------|---------------------------------|---------------------------|-----------------------|
| <i>switching</i> | <i>fixed</i> | <i>centralized</i> | <i>distributed</i> | <i>ICV rate < 1</i> | <i>ICV rate = 1</i> |
| [5], [7], [9], [10], [13]–[17] | [6], [18]–[20], [21] ¹ | [6], [8], [10], [14]–[19] | [5], [7]–[9], [13], [21] | [5] | [6]–[10], [13]–[21] |
| Cooperative Controller | | | | | |
| <i>upper scheduling + lower tracking</i> | | <i>global optimization</i> | <i>feedback linearization</i> | <i>MPC</i> | <i>neural network</i> |
| [8], [10], [14]–[16], [18] | | [17], [19], [21] | [20] | [7]–[9] | [5], [13], [17] |
| Formation Geometry | | | | | |
| <i>multi-lane road</i> | <i>single-lane road</i> | <i>ramp/roundabout</i> | <i>intersection</i> | | |
| [13], [18], [21] | [5], [20] | [10] | [5]–[10], [14]–[17], [19], [21] | | |
| Node Dynamics | | | | | |
| <i>1 DoF²</i> | | <i>≥ 2 DoF</i> | | <i>homogeneous</i> | <i>heterogeneous</i> |
| [6]–[10], [13]–[17], [20] | | [5], [18], [19], [21] | | [5]–[10], [13]–[19], [21] | [20] |

¹ Topology assumed to be fixed during each predictive horizon.

² Degree of Freedom.

downstream actuator, while formation geometry relates the dynamic reference trajectory and time-varying parameters in the controller simultaneously. Under the proposed framework, one

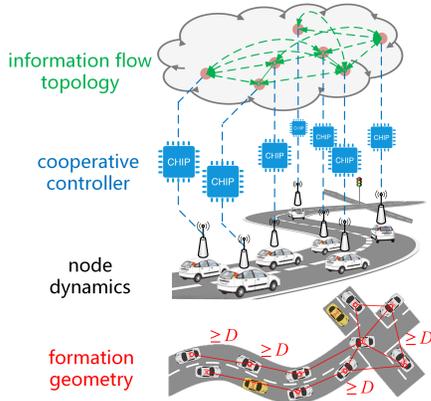


Fig. 1. Four-component framework

can design local planning algorithms and reactive distributed controllers to enhance the efficiency and safety of the whole traffic with a continuously varying ICV proportion. In addition, the component could be used as independent parameters for sensitivity analysis. For example, given the proportion of ICVs (describing information flow topology) and distribution of vehicles' dynamic parameters (describing node dynamics), the optimization effect of different cooperative controllers at intersections, roundabouts, and other scenarios (describing formation geometry) could be statistically analyzed.

We list some of the important works categorized by the framework in Table I as an intuitive example. Each of the four components could be divided according to different standards. Note that there could be more genres added as new columns, e.g., many classic fixed topologies analyzed in a platoon. However, when it comes to general connected multi-vehicle systems, the dynamic feature of the vehicle group usually makes the topology switching. We refer the reader to [1] for an overview of platoon-related literature, which is not listed in this table for brevity. We further do not include details of trajectory planning (in the formation geometry component) for simplicity.

III. DISTRIBUTED COOPERATION

In this section, we first develop a local trajectory planning method, which takes into account plan consistency. After that, we formulate a distributed model predictive control problem. The two methods will specify the formation geometry and cooperative controller in Section II, and serve the four-component example in Section IV.

A. real-time consistent trajectory planning

We use cubic Bezier curves and straight lines to generate smooth spatiotemporal trajectories, assuming there are no obstacles at this stage. The input are lane midline feature points, speed limit, and traffic light carrying the environmental context, as shown in Fig. 2, which can easily be obtained from onboard sensing modules or roadside infrastructure. Our

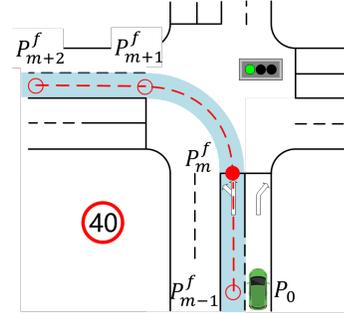


Fig. 2. Feature points and environmental context at an intersection

approach assumes that each vehicle has access to its location, denoted as $P_0 = [X_0, Y_0, \phi_0]^T$ and thus can identify the next feature point $P_m^f = [X_m^f, Y_m^f, \phi_m^f]^T$ indexed by m . More specifically, m will be updated by $m + 1$ as the vehicle drives into a certain vicinity of P_m^f . If another two assisting points $P_1^a = [X_1^a, Y_1^a]^T$ and $P_2^a = [X_2^a, Y_2^a]^T$ are determined, the trajectory points can be expressed using the following equations with regard to a continuous variable τ , which is parameterized by X_0, X_1^a, X_2^a, X_m^f and Y_0, Y_1^a, Y_2^a, Y_m^f . when $\tau \in [0, 1]$,

$$\begin{aligned} \mathcal{R}(\tau | P_0, P_1^a, P_2^a, P_m^f) \\ = \begin{bmatrix} X_0 & 3X_1^a & 3X_2^a & X_m^f \\ Y_0 & 3Y_1^a & 3Y_2^a & Y_m^f \end{bmatrix} \begin{bmatrix} (1-\tau)^3 \\ (1-\tau)^2\tau \\ (1-\tau)\tau^2 \\ \tau^3 \end{bmatrix}, \end{aligned} \quad (1)$$

when $\tau \in (1, +\infty)$,

$$\mathcal{R}(\tau | P_m^f) = \begin{bmatrix} X_m^f & \cos \phi_m^f \\ Y_m^f & \sin \phi_m^f \end{bmatrix} \begin{bmatrix} 1 \\ \tau - 1 \end{bmatrix}. \quad (2)$$

Since the reference trajectory should be tangent to the current point P_0 and the feature point P_m^f , the first and second auxiliary points P_1^a, P_2^a are determined by

$$\begin{cases} X_1^a = X_0 + L_0 \cos(\phi_i) \\ Y_1^a = Y_0 + L_0 \sin(\phi_i) \end{cases} \quad (3a)$$

$$\begin{cases} X_2^a = X_m^f - L_3 \cos(\phi_m^f) \\ Y_2^a = Y_m^f - L_3 \sin(\phi_m^f) \end{cases}. \quad (3b)$$

The distance L_0 and L_3 could be generated in different ways. One intuitive way is to set them to some fixed values or define them as a function of the vehicle speed. However, the reference trajectory could vary with each round of planning even if there were no outer interventions, e.g., obstacles. Note that this phenomenon may introduce unwanted fluctuations for the whole closed-loop system consisting of the planning module and the controlled vehicle dynamics.

Here we propose a method with which the successive planned trajectories share some extent of consistency: the next-step curve is a part of the current curve. When there are no obstacles or disturbances, the online planned trajectory equivalently makes the vehicle follow one continuous curve. Specially, when the vehicle deviates from the planned trajectory to avoid a crash, the new trajectory will strike a balance between its original intent and its current position. To achieve this effect, we can derive an equation with the condition that there exist common points on both the last trajectory and the new trajectory:

$$\mathcal{R}(\tau | P_0, P_1^a, P_2^a, P_m^f) = \mathcal{R}(\tau' | P_0', P_1^{a'}, P_2^{a'}, P_m^f) \quad (4)$$

If points $\tau = \tau_1, \tau_2$ are on the last-step trajectory, and they correspond to $\tau' = \tau_1', \tau_2'$ on the current trajectory. Given τ' and the distance that vehicle has travelled during the most recent planning interval (namely distance $d_{P_0 P_0'}$ in Fig. 3), the mapping relationship can be easily estimated:

$$\frac{1 - \tau'}{1 - \tau} = \frac{d_{P_m^f P_0}}{d_{P_m^f P_0'}} \approx \frac{1}{1 - \tau_{\text{firstpoint}}} \quad (5)$$

Note that d in (5) denotes the generalized string length of the curve. The division between string lengths could be estimated from the value of τ at the first reference point on the translucent one, i.e., $\tau_{\text{firstpoint}}$. Therefore, choosing $\tau_1, \tau_2 \in (0, 1)$, calculating τ_1', τ_2' according to (5), with known parameters $X_0, X_1^a, X_2^a, X_m^f, Y_0, Y_1^a, Y_2^a, Y_m^f$ and $X_0', Y_0', (4)$ turns into a linear system of equations relating $X_1^{a'}, X_2^{a'}, Y_1^{a'}$, and $Y_2^{a'}$, which can be solved rapidly using matrix operations. The value of τ_1, τ_2 here will determine which part of the two curves are overlapping. Intuitively, the planned trajectory will consistently lead to P_m^f even if the vehicle has deviated from its expected path. In other words, the planning module shows a form of natural resilience against disturbance. This is another advantage of our method, besides the consistency in no-disturbance environments.

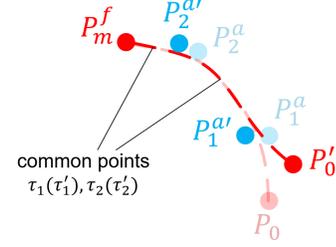


Fig. 3. Real-time iterative planning with cubic bezier curve

After a continuous curve is determined, discrete points can be serially found on it according to the speed profile within the prediction time horizon, considering the speed limit, traffic light phase, and distance to the stop line (indicated in Fig. 2). We present the overall algorithm for real-time consistent trajectory planning in Algorithm 1. This algorithm works in an iterative and receding-horizon manner, i.e., the output $L_0^{\text{new}}, L_3^{\text{new}}$ will be used as the input L_0, L_3 of the next planning step. Additionally, the feature point P_m^f will be updated by P_{m+1}^f according to the output flag UpdateFeaturePoint. Note that the constant L can be designed as a function of the vehicle speed.

Algorithm 1 Real-time Consistent Planning

Input: predictive horizon N_p , current position P_0 , feature point P_m^f , speed sequence \mathbf{v} , time interval T , last-step output L_0, L_3 . constants: $L > 0$; $\tau_1, \tau_2 \in (0, 1)$ and $\tau_1 \neq \tau_2$; ErrorRate $\in (0, 0.2)$; $\tau_{\text{bound}} \in (0, 1)$
Output: $\mathcal{R}_0(\tau)$, $L_0^{\text{new}}, L_3^{\text{new}}$, UpdateFeaturePoint

- 1: UpdateFeaturePoint = False
- 2: $\tau = [0, 0, \dots, 0]_{1 \times (N_p + 1)}$
- 3: calculate P_1^a, P_2^a with (3)
- 4: $\mathcal{R}_0(\cdot) \doteq \mathcal{R}(\cdot | P_0, P_1^a, P_2^a, P_m^f)$
- 5: **for** idx = 2 : $N_p + 1$ **do**
- 6: $\Delta L = \mathbf{v}[\text{idx}] \cdot T$
- 7: specify $\tau[\text{idx}]$ with binary search, such that $|\|\mathcal{R}_0(\tau[\text{idx}]), \mathcal{R}_0(\tau[\text{idx} - 1])\|_2 - \Delta L| \leq \text{ErrorRate} \cdot \Delta L$
- 8: **end for**
- 9: **if** $\tau[2] > \tau_{\text{bound}}$ **then**
- 10: UpdateFeaturePoint = True
- 11: $L_0^{\text{new}} = L$
- 12: $L_3^{\text{new}} = L$
- 13: **else**
- 14: $\tau_1' = (\tau[2] - \tau_1) / (\tau[2] - 1)$ //using (5)
- 15: $\tau_2' = (\tau[2] - \tau_2) / (\tau[2] - 1)$ //using (5)
- 16: solve linear system of equations regarding $X_1^{a'}, X_2^{a'}, Y_1^{a'}, Y_2^{a'}$

$$\begin{cases} \mathcal{R}(\tau_1' | P_0', P_1^{a'}, P_2^{a'}, P_m^f) = \mathcal{R}(\tau_1 | P_0, P_1^a, P_2^a, P_m^f) \\ \mathcal{R}(\tau_2' | P_0', P_1^{a'}, P_2^{a'}, P_m^f) = \mathcal{R}(\tau_2 | P_0, P_1^a, P_2^a, P_m^f) \end{cases}$$
- 17: $L_0^{\text{new}} = \|P_0' - P_1^{a'}\|_2$
- 18: $L_3^{\text{new}} = \|P_2^{a'} - P_m^f\|_2$
- 19: **end if**

B. nonlinear distributed predictive control

The objective and constraints of the nonlinear distributed model predictive controller are listed below. There, the trajectory $\mathcal{R}_0(\tau[i+1])$ generated by the aforementioned algorithm 1 is denoted as $x_{t+i}^R, i = 0, 1, \dots, N_p - 1$.

$$\min_{u_t, \dots, u_{t+N_p-1}} \sum_{i=0}^{N_p-1} (x_{t+i} - x_{t+i}^R)^T Q (x_{t+i} - x_{t+i}^R) + u_{t+i}^T R u_{t+i} \quad (6)$$

$$\text{s.t.} \quad x_{t+i+1} = f(x_{t+i}, u_{t+i}) \quad (7a)$$

$$D - (x_{t+i} - x_{t+i}^j)^T Q (x_{t+i} - x_{t+i}^j) \leq 0 \quad (7b)$$

$$x_{\min} \leq x_{t+i} \leq x_{\max} \quad (7c)$$

$$u_{\min} \leq u_{t+i} \leq u_{\max} \quad (7d)$$

$$x_{t+i,\min}^k \leq x_{t+i} \leq x_{t+i,\max}^k \quad (7e)$$

N_p is the predictive horizon length. $0 \leq j \leq N_{\text{neb}}(t)$, $N_{\text{neb}}(t)$ is the number of neighbors of the vehicle at time t . These neighbors are identified by either onboard sensors or wireless communication, all of which are within a sensing range, considered large enough to ensure detection in advance. $1 \leq k \leq N_{\text{bound}}(t)$, $N_{\text{bound}}(t)$ is the number of constraints determined by drivable boundaries of the vehicle at time t . Q and R are weighting matrices. D is the inter-vehicle distancing parameter, which is simplified to a constant here, as the vehicle profiles are assumed to be identical. u_{\max} , u_{\min} and x_{\max} , x_{\min} are bounds on the control and state vectors. x_{t+i}^R is the reference trajectory point at time $t+i$. x_{t+i}^j is the predicted/intended state of the j -th surrounding vehicle at time $t+i$. Connected vehicles will share their predicted state from the last solution of their local optimal control problem; for non-connected vehicles, a simple prediction by extending their current state for N_p steps is applied. x_{t+i}^k is the k -th drivable boundary at time $t+i$. $f(\cdot)$ is the nonlinear vehicle dynamics in [22].

IV. SIMULATION VERIFICATION

We select a cooperative driving task with distributed connected vehicles in an unsignalized intersection scenario. We specify the simulation settings by the four components as follows.

- Information Flow Topology

Vehicles are independent and equal nodes communicating with other connected nodes within a given distance. This assumption makes the topology switching. The minimum range could be estimated by the maximum speed \times the prediction time horizon $\times 2$. We do not specify this value since our simulation is on a small area, in which it is reasonable to assume the environment fully connected.

- Cooperative Controller

The distributed predictive controllers are constructed according to (6) and (7). $Q = \text{diag}(100, 100, 0, 0, 0, 0)$, $R = \text{diag}(21, 500)$. $x_{\min} = [-\infty, -\infty, -\infty, 0, -4, -3]$, $x_{\max} = [+\infty, +\infty, +\infty, 20, 4, 3]$. $u_{\min} = [-3.5, -\pi/5]$, $u_{\max} = [2.5, \pi/5]$. All quantities are in the international system of units. $T_s = 0.1$ s, $N_p = 20$.

- Node Dynamics

In order to provide a convenient interface for the controller to call, we build a heterogeneous dynamic module in a parametric modelling manner. Throttle feedforward and feedback are embedded, so that this model can automatically track the desired acceleration instructions from the upstream controller. The lateral dynamics adopt the classic single-track model, which is discretized in a tailored way to keep it numerically

stable [22]. Fig. 4 gives the general logic of the longitudinal dynamics, considering heterogeneous vehicles. Note that the heterogeneity could be both over parametric differences like scale, weight, ratios etc., or over structural differences in powertrain. Three types of dynamics are used in the following simulation. Specifically, the vehicle classes follow the *CarSim* prototypes ‘‘C-Class, Hatchback 2017’’, ‘‘F-Class, Sedan w/ CVT’’, and the *TruckSim* prototype ‘‘LCF Van 5.5T/8.5T’’ respectively. The accuracy verification of dynamics module is not included in this paper, since it is not focused on the accurate prediction of a field test, but the heterogeneity and imperfect models when carrying out MPC.

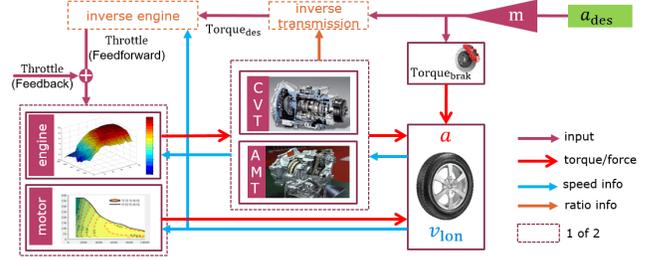


Fig. 4. Heterogeneous longitudinal dynamics

- Formation Geometry

The real-time planning algorithm in section III-A is adopted to generate the trajectories. The reference speeds are always 8 m/s. The squared inter-vehicle distance bound D is set to be 20 and sent to the cooperative controller. The environmental context, such as lane boundaries are neglected for simplicity. Note that when carrying out a field test, the time-varying boundaries should be included for safety guarantees.

A. Simulation with 3 vehicles

The simulation process is illustrated by Fig. 5. The speeds of the three vehicles are shown in Fig. 6. This is another evidence, beyond the trajectories shown in Fig. 5, demonstrating that all vehicles have adaptively selected their own way to collaborate with others. For example, vehicle A turns right and brakes, whereas vehicle B and C keep smoothly driving. Human drivers interact without communication but with traffic rules, social etiquette and common sense, and we show that ICVs could behave similarly only with simple intention sharing mechanism based on low-bandwidth wireless communication.

This effect is a balance achieved by the vehicles spontaneously, without any explicitly assigned priority or rules. In addition, the mobility of both longitudinal and lateral directions helps vehicles to find a better joint solution, compared to many methods assuming fixed paths. More importantly, since the global route topology within the intersection is not necessary for decision making, the simulation could easily extended to other scenarios.

The control inputs together with historical intentions of the three vehicles are shown in Fig. 7. Each translucent line represents the 20-step-long predicted control values, which reveals the receding-horizon nature of our MPC controller. Note that the executed actions’ solid lines are mostly within the envelop of the translucent ones, essentially demonstrating

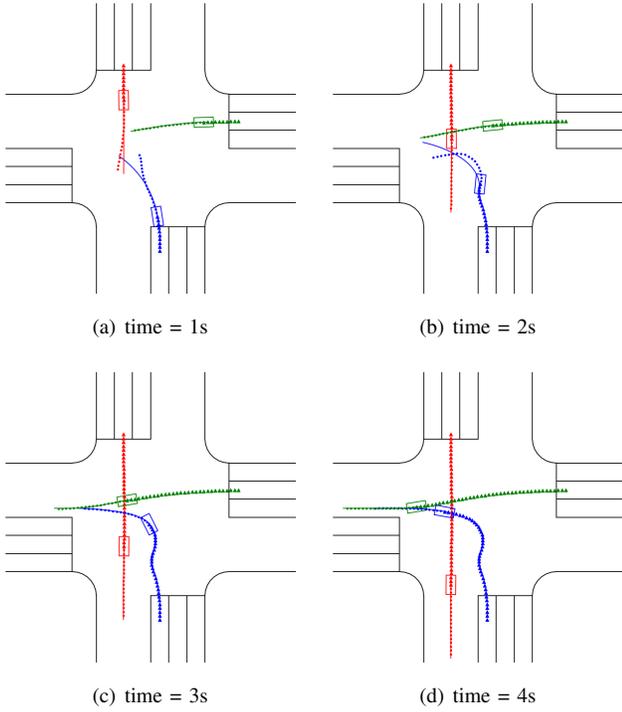


Fig. 5. Trajectories of three heterogeneous ICVs. Round dot: intention; Triangle dot: trajectory; Solid line: planned trajectory.

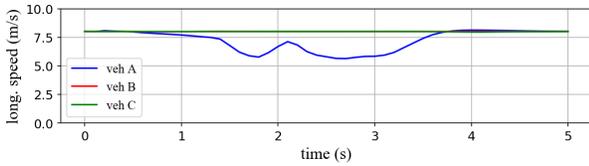


Fig. 6. Speed

that the vehicles have successfully cooperated after implicit interactions. If consensus is not approached, the action lines would frequently go outside the intention lines and show more fluctuations.

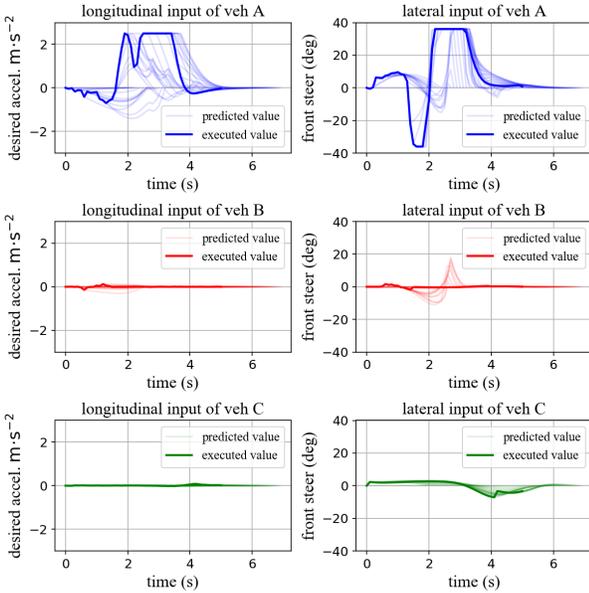
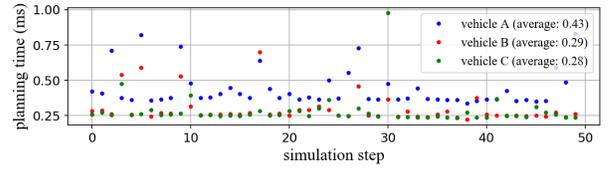
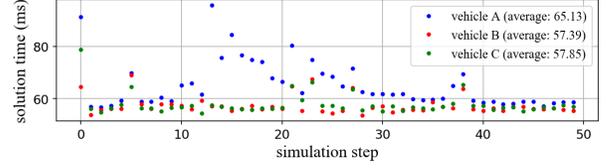


Fig. 7. Intentions and control values

The planning time and MPC solution time are shown



(a) planning time



(b) solution time

Fig. 8. Step-wise time consumption

in Fig. 8(a) and Fig. 8(b). Algorithm 1 is implemented in C++ and wrapped by a *windows* dynamic link library (.dll), which is called by python in our simulations. The optimal control problem is solved by *ipopt* [23] under the *CasAdi* framework [24], on a laptop with an *intel core i7* CPU. The consistent planning is validated to be effective and fast, while we acknowledge that the computation burden of nonlinear MPC could become an issue when applied to real vehicles with much more constraints, likely less computational resources, and more stringent real-time requirements.

B. Simulation with 8 vehicles

In Fig. 9, 8 vehicles (2 trucks, 3 CVT-cars, and 3 AMT-cars) manage to drive into their respective target lanes without collisions or full stops. Acceleration/deceleration and left/right steering, i.e., the same actions as human drivers, were dynamically computed and executed, tracking a dynamic trajectory. This scenario and task further reveal the potential of decentralized cooperation of ICVs under dense and complex traffic environments.

Readers may be interested with the optimality and safety guarantees of distributed control. Such a distributed cooperation (without any upper-level optimization) is probably non-optimal, if judged from the sum of all objectives and the combination of all action spaces, as expected when relying on decentralized control. However, each vehicle node could still guarantee safety, given a feasible initial state with enough safety margin, and an optimal controller capable of satisfying constraints.

V. CONCLUSION

We generalize the four-component framework from vehicular platoon field to generic multi-vehicle tasks such as unsignalized traffic junctions. We further propose a real-time trajectory planning and distributed control method, where downstream distributed controllers allow vehicles to share their intentions to avoid collisions. Our planning method can adapt to various urban scenarios, since it only requires easy-to-acquire feature points, while our control method is distributed and thus naturally scalable. We further verify the proposed methods in an unsignalized intersection scenario involving up

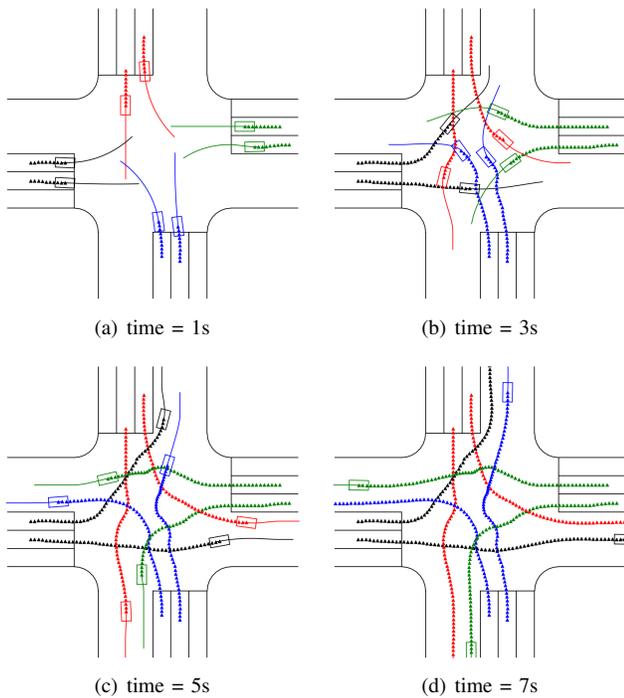


Fig. 9. Trajectories of eight heterogeneous ICVs. Triangle dot: trajectory; Solid line: planned trajectory.

to 8 heterogeneous vehicles planning individual collision-free paths based on local communications.

Future work will consider the use of learning-based and model-based methods to enhance the solution efficiency of the constrained nonlinear optimal tracking problem, to both improve performances and decrease computation time towards real-time implementation under realistic conditions. Additionally, we will investigate the framework's generalizability with more scenarios.

ACKNOWLEDGMENT

Ge Qiang would like to acknowledge the State Scholarship Fund provided by the China Scholarship Council that supported his studies in Singapore.

REFERENCES

- [1] S. E. Li, Y. Zheng, K. Li, and J. Wang, "An overview of vehicular platoon control under the four-component framework," in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 286–291.
- [2] B. Li, T. Acarman, Y. Zhang, Y. Ouyang, C. Yaman, Q. Kong, X. Zhong, and X. Peng, "Optimization-based trajectory planning for autonomous parking with irregularly placed obstacles: A lightweight iterative framework," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [3] H. Mouhagir, V. Cherfaoui, R. Talj, F. Aioun, and F. Guillemard, "Trajectory planning for autonomous vehicle in uncertain environment using evidential grid," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 12 545–12 550, 2017.
- [4] S. Liu, N. Atanasov, K. Mohta, and V. Kumar, "Search-based motion planning for quadrotors using linear quadratic minimum time control," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 2872–2879.
- [5] Y. Guan, Y. Ren, Q. Sun, S. E. Li, H. Ma, J. Duan, Y. Dai, and B. Cheng, "Integrated decision and control: toward interpretable and computationally efficient driving intelligence," *IEEE transactions on cybernetics*, 2022.

- [6] Y. Zhang and C. G. Cassandras, "Decentralized optimal control of connected automated vehicles at signal-free intersections including comfort-constrained turns and safety guarantees," *Automatica*, vol. 109, p. 108563, 2019.
- [7] X. Qian, J. Gregoire, A. De La Fortelle, and F. Moutarde, "Decentralized model predictive control for smooth coordination of automated vehicles at intersection," in *2015 European control conference (ECC)*. IEEE, 2015, pp. 3452–3458.
- [8] X. Zhao, J. Wang, G. Yin, and K. Zhang, "Cooperative driving for connected and automated vehicles at non-signalized intersection based on model predictive control," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 2121–2126.
- [9] A. Katriniok, P. Kleibbaum, and M. Joševski, "Distributed model predictive control for intersection automation using a parallelized optimization approach," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5940–5946, 2017.
- [10] A. I. Mahbub, A. A. Malikopoulos, and L. Zhao, "Decentralized optimal coordination of connected and automated vehicles for multiple traffic scenarios," *Automatica*, vol. 117, p. 108958, 2020.
- [11] W. Kim, J. Park, and Y. Sung, "Communication in multi-agent reinforcement learning: Intention sharing," in *International Conference on Learning Representations*, 2020.
- [12] B. Alrifae, "Networked model predictive control for vehicle collision avoidance," Ph.D. dissertation, Dissertation, RWTH Aachen University, 2017, 2017.
- [13] C. Yu, X. Wang, X. Xu, M. Zhang, H. Ge, J. Ren, L. Sun, B. Chen, and G. Tan, "Distributed multiagent coordinated learning for autonomous driving in highways based on dynamic coordination graphs," *Ieee transactions on intelligent transportation systems*, vol. 21, no. 2, pp. 735–748, 2019.
- [14] B. Xu, S. E. Li, Y. Bian, S. Li, X. J. Ban, J. Wang, and K. Li, "Distributed conflict-free cooperation for multiple connected vehicles at unsignalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 93, pp. 322–334, 2018.
- [15] C. Chen, Q. Xu, M. Cai, J. Wang, J. Wang, and K. Li, "Conflict-free cooperation method for connected and automated vehicles at unsignalized intersections: Graph-based modeling and optimality analysis," *IEEE Transactions on Intelligent Transportation Systems*, 2022.
- [16] Q. Ge, Q. Sun, Z. Wang, S. E. Li, Z. Gu, S. Zheng, and L. Liao, "Real-time coordination of connected vehicles at intersections using graphical mixed integer optimization," *IET Intelligent Transport Systems*, vol. 15, no. 6, pp. 795–807, 2021.
- [17] Y. Guan, Y. Ren, S. E. Li, Q. Sun, L. Luo, and K. Li, "Centralized cooperation for connected and automated vehicles at intersections by proximal policy optimization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12 597–12 608, 2020.
- [18] M. Cai, Q. Xu, C. Chen, J. Wang, K. Li, J. Wang, and X. Wu, "Formation control with lane preference for connected and automated vehicles in multi-lane scenarios," *Transportation research part C: emerging technologies*, vol. 136, p. 103513, 2022.
- [19] B. Li, Y. Zhang, Y. Zhang, N. Jia, and Y. Ge, "Near-optimal online motion planning of connected and automated vehicles at a signal-free and lane-free intersection," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1432–1437.
- [20] J. Hu, P. Bhowmick, F. Arvin, A. Lanzon, and B. Lennox, "Cooperative control of heterogeneous connected vehicle platoons: An adaptive leader-following approach," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 977–984, 2020.
- [21] Z. Wang, Y. Zheng, S. E. Li, K. You, and K. Li, "Parallel optimal control for cooperative automation of large-scale connected vehicles via admm," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1633–1639.
- [22] Q. Ge, Q. Sun, S. E. Li, S. Zheng, W. Wu, and X. Chen, "Numerically stable dynamic bicycle model for discrete-time control," in *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*. IEEE, 2021, pp. 128–134.
- [23] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.
- [24] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.