Distributed Conflict Resolution for Connected Autonomous Vehicles

Changliu Liu, Chung-Wei Lin, Shinichi Shiraishi, *Member, IEEE*, and Masayoshi Tomizuka, *Fellow, IEEE*

***Abstract*—This paper proposes a novel communication-enabled distributed conflict resolution mechanism in order for a group of connected autonomous vehicles to navigate safely and efficiently in intersections without any traffic manager. The conflict resolution strategy for individual vehicle is decoupled temporally. In a decision maker, the vehicle computes the desired time slots to pass the conflict zones by solving a conflict graph locally based on the broadcasted information from other vehicles. In a motion planner, the vehicle computes the desired speed profile by solving a temporal optimization problem constrained in the desired time slot. The estimated time to occupy the conflict zones given the new speed profile is then broadcasted again. It is proved that the aggregation of these local decisions solves the conflicts globally. Theoretically, this method provides an efficient parallel mechanism to obtain local optimal solutions of a large-scale optimization problem (e.g., multivehicle navigation). Application-wise, as demonstrated by extensive simulation, this mechanism increases the efficiency of autonomous vehicles in terms of smaller delay time, as well as the efficiency of the traffic in terms of larger throughput when there is no traffic manager to mediate the conflicts.**

***Index Terms*—Connected autonomous vehicles, communication, motion planning, intersection management.**

I. Introduction

CONNECTED autonomous vehicles (CAVs) are believed to be the solution for future mobility [1]. However, it still remains unclear how the technology may help vehicles solve conflicts at intersections when a central traffic manager is not applicable. A typical conflicting scenario is illustrated in Fig. 1, where two vehicles cannot occupy a conflict zone at the same time. To resolve the conflicts, it is important to determine both the passing order, which can be understood as a scheduling problem, and the optimal vehicle trajectories under such order, which is a motion planning problem. In the following discussion, we motivate the design of a communication-based distributed conflict resolution mechanism from three different perspectives, i.e., the individual vehicle perspective, the transportation system perspective, as well as the combinatorial optimization perspective.

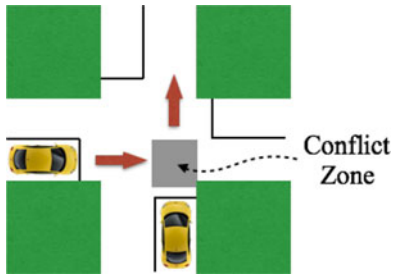

Fig. 1. Illustration of conflicts in multi-vehicle interactions.

A. Perspective From Individual Autonomous Vehicles

The behavior of an autonomous vehicle depends on how it perceives and predicts the surrounding world [2], which is constrained by its sensing capabilities. When there are large uncertainties in perception, the vehicle's behavior tends to be conservative. Vehicle to vehicle (V2V) communication can compensate the deficiency in perception as well as reduce the uncertainties in predicting other vehicles' behaviors.

Nonetheless, conflicts still exist even when there is communication. The conflict in Fig. 1 can be described in Chicken game [3], where every vehicle needs to choose between "yield" and "go." The desired outcome for the system is that one vehicle yields and the other goes. However, as this is a simultaneous game, when both vehicles have the same reasoning strategies, they may always come up with the same yield or go choice, which is unsafe when they both decide to "go," and inefficient when they both decide to "yield."

A conflict resolution mechanism is needed to let the vehicles determine the passing order and optimal trajectories by themselves. In this paper, we investigate the design of the mechanism on individual vehicles, and the macroscopic impacts (e.g., average delay and throughput of the transportation system) of the microscopic design.

B. Perspective From the Transportation System

Existing conflict resolution mechanisms are: 1) centralized mechanisms such as traffic lights to control the traffic flow [4]; 2) decentralized mechanisms such as traffic rules to provide

Manuscript received September 1, 2017; revised December 2, 2017; accepted December 10, 2017. Date of publication December 29, 2017; date of current version March 19, 2018. *(Corresponding author: Changliu Liu.)*

C. Liu is with the Department of Mechanical Engineering, University of California, Berkeley, Berkeley, CA 94720 USA, and also with the Toyota InfoTechnology Center, USA, Inc., Mountain View, CA 94043 USA (e-mail: changliuliu@berkeley.edu).

C.-W. Lin and S. Shiraishi are with the Toyota InfoTechnology Center, USA, Inc., Mountain View, CA 94043 USA (e-mail: cwlin@us.toyota-itc.com; sshiraishi@us.toyota-itc.com).

M. Tomizuka is with the Department of Mechanical Engineering, University of California, Berkeley, Berkeley, CA 94720 USA (e-mail: tomizuka@berkeley.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIV.2017.2788209

guidance for the vehicles to "negotiate" locally. However, these methods are inefficient, as vehicles must wait for red light or stop before the stop sign even if there is no vehicle from other direction. Meanwhile, conflicts are not fully resolved. If four vehicles arrive at the a four-way-stop intersection at the same time, it is undetermined who should go first.

In literature, many centralized mechanisms have been proposed to solve the conflicts based on vehicle to infrastructure (V2I) communication. For example, utilizing one-way communication from vehicle to infrastructure, intelligent traffic signal control algorithms are able to adaptively adjust traffic lights based on the information received from individual vehicles [5]–[8]. To fully exploit the advantage of bilateral V2I communication, a centralized traffic manager can talk to individual vehicles and allocate specific time slots for them to pass the intersection after processing the information from all vehicles [9]–[12]. The manager functions as a virtual traffic light that can change at infinite frequency. The advantage of these methods is that individual vehicles no longer need to worry about conflicts with other vehicles, though they still need to decide on their trajectories given the instructions from the traffic manager. Nonetheless, methods such as cooperative adaptive cruise control (CACC) [13] or centralized model predictive control (MPC) [14], [15] offer integrated scheduling and trajectory planning for multiple vehicles. The disadvantage of centralized methods is the strong dependency on infrastructure such as a traffic manager.

A distributed conflict resolution mechanism that does not depend on the manager is highly desired for small intersections and in the case that the manager is broken. A reservation-based method by V2V communication is discussed in [16], which allows vehicles to claim time slots to pass the intersection by broadcasting the "claims" to others. Whoever claims first gets the time slot. However, there may be deadlocks when the traffic density goes up. A distributed method considering V2V communication failure is discussed in [17]. However, it remains unclear how this method can be extended to intersections with multiple vehicles from multiple directions. Several deadlock-free communication protocols for multi-vehicle interactions at intersections are discussed in [18]. However, these methods are rigid in the sense that once the priority (based on first-come-first-serve) is determined, the passing order will not be adjusted in real time. The only exception is in the advanced maximum progression intersection protocol (AMP-IP) [18], where a low priority vehicle is allowed to go first if it is anticipated to leave all conflict zones earlier than the arrival time of all high priority vehicles. However, the traffic efficiency can be further improved if the passing order is adapted to real-time traffic, as will be demonstrated later in this paper. Nonetheless, existing methods remain in the strategic level and neglects system dynamics, whose scalability and applicability in complex real traffic scenarios are limited.

Indeed, a desired distributed conflict resolution mechanism should be able to

1) ensure physical **feasibility**, e.g., if vehicle 1 needs to yield vehicle 2, then there should exist a feasible trajectory for vehicle 1 to do that;
2) ensure **completeness**, e.g., the passing order should be decided for any pair of vehicles that go through a same conflict zone;
3) **avoid deadlocks** in the local decisions, e.g., two vehicles should not decide to yield each other at the same time;
4) solve the conflicts in **real-time**, e.g., vehicles can find feasible and conflict-free trajectories in finite time steps.
5) ensure closed-loop **stability** over time, e.g., the passing order and the conflict-free trajectories should be consistent in different time steps.

These points will be addressed in the proposed method.

C. Perspective From Combinatorial Optimization

The multi-vehicle navigation problem is widely studied as a combinatorial optimization problem [14], which optimizes the passing order (discrete scheduling problem) as well as vehicle trajectories (continuous motion planning problem) simultaneously. However, the problem is hard to solve in one processor since the scheduling problem is NP-hard, and the dimension of the motion planning problem grows with the number of vehicles in the system. Parallel computation is desired to scale down the problem. It is natural to distribute the computation load to individual vehicles. The interactions among processors should be regulated through the conflict resolution mechanism. Moreover, the priority in computation is not to find an optimal solution of the combinatorial problem, but to find one feasible solution in real time.

D. Contributions

In this paper, we propose a new *distributed* conflict resolution mechanism via V2V communication, which addresses the five requirements discussed earlier. The mechanism does not require a central manager, and it works in any traffic condition. The key idea is to let the vehicles 1) broadcast their estimated time intervals to occupy the conflict zones, 2) reach a consensus of passing order by solving a conflict graph locally given the broadcasted information, 3) adjust their speed profiles according to the passing order and update the estimated time intervals to occupy the conflict zones. In the case of communication failure, the vehicle can go back to the basic autonomous driving mode which relies only on local sensing instead of on the received information.

This distributed conflict resolution mechanism can achieve *stable local optimal* solutions of the combinatorial optimization problem in real time (Section IV), which can be applied to conflict resolution at intersections, or even problems beyond multi-vehicle navigation. It should be emphasized that we are *not* claiming that a distributed mechanism is better than a centralized mechanism. We believe, depending on design metrics (e.g., cost, reliability), each individual of them or even a combination of them can be an appropriate solution. The proposed mechanism in this paper can fill in the *distributed* part, where reliable and scalable solutions are still missing in literature.

The remainder of the paper is organized as follows. The mathematical problem for conflict resolution among multiple vehicles and multiple conflict zones is formulated as a constrained

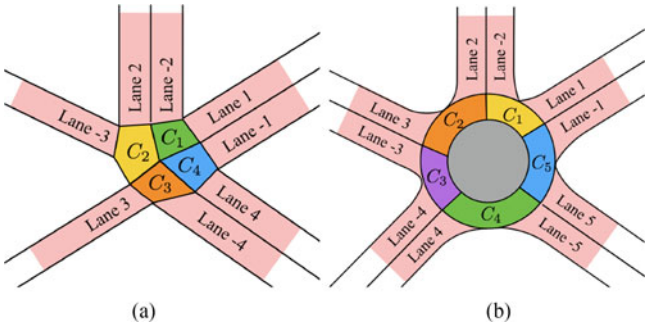

Fig. 2. Geometry of the environment. (a) Intersection. (b) Roundabout.

optimization in Section II. The distributed strategy to solve the conflicts is proposed in Section III. Section IV provides theoretical analysis of the strategy. Simulation results are presented in Section V. Section VI concludes the paper.

II. Problem Formulation

In this section, the geometry of the environments under consideration is discussed first. Then a multi-vehicle navigation problem is formulated from the system level, followed by a distributed formulation for individual vehicles.

A. Geometry of the Environment

Geometry-wise, we consider general intersections where multiple incoming lanes and multiple outgoing lanes intersect. Fig. 2 illustrates two typical environments: (real) intersection and roundabout. Every lane has a unique lane ID. The IDs of incoming lanes are positive and the IDs of outgoing lanes are negative. A conflict zone is formulated when the extensions of two incoming lanes intersect with each other. The conflict zones are denoted by $C_1, \ldots, C_L$ where L is the total number. In Fig. 2(a), four conflict zones are identified, while five conflict zones are identified in Fig. 2(b).

B. Centralized Formulation

Suppose there are N road participants in the system, indexed from 1 to N. The state of vehicle i at time t is denoted as $x_i(t)$. The system state at time step t is denoted as $x(t) := [x_1(t); \ldots; x_N(t)]$. Let G_i be the intention of vehicle i, which denotes its target lane after passing the intersection.

The system objective is to ensure that the intents of vehicles are satisfied efficiently, while maintaining the safety of the system. At time $\mathbf{t}_0$, given the initial state $x(\mathbf{t}_0)$ of the system, the trajectory $x(t)$ for $t > \mathbf{t}_0$ needs to be computed. The multi-vehicle navigation problem is formulated below,

$$\min_{x} \sum_{i} w_i J(x_i, G_i), \tag{1a}$$

$$\text{s.t. } \dot{x}_i(t) \in \Gamma(x_i(t)), \forall i, \tag{1b}$$

$$x(t) \in \mathcal{X}, \tag{1c}$$

where the objective function is a weighted sum of the cost function for every road participant. The weights correspond to priorities, e.g., the needs for the vehicles on the main street should be addressed first. Equation (1b) is the feasibility constraint to ensure that there is a low level controller to track the trajectory, e.g.,

$$\Gamma(x_i) := \{\dot{x}_i | \exists u_i, \text{ s.t. } \dot{x}_i = f(x_i, u_i)\}, \tag{2}$$

where $\dot{x}_i = \partial x_i / \partial t$, u_i is the vehicle control input (wheel angle and throttle torque) and f describes the vehicle dynamics. Equation (1c) is the safety constraint for the system that requires the minimum distance between any two vehicles is greater than a threshold $d_{\min}$, e.g.,

$$\mathcal{X} := \{x | d(x_i, x_j) \geq d_{\min}, \forall i, j, i \neq j\}, \tag{3}$$

where function d measures the minimum distance between two vehicles. The problem (1) is combinatorial in the sense that in order to satisfy $d(x_i, x_j) > d_{\min}$, vehicle i can either enter the conflict zone before vehicle j or after vehicle j, each corresponding to a different local optimum.

C. Distributed Formulation

The centralized optimization problem (1) can be broken into distributed optimization problems for individual vehicles. Each individual vehicle has only local view and local information, i.e., vehicle i only considers the road participants in its neighborhood $\mathcal{N}_i$. Constraint (1b) can be directly inherited for single vehicle, while in constraint (1c), other vehicles' motions need to be estimated. The optimization problem for vehicle i is formulated as,

$$\min_{x_i} J(x_i, G_i), \tag{4a}$$

$$\text{s.t. } \dot{x}_i(t) \in \Gamma(x_i(t)), \tag{4b}$$

$$d(x_i(t), \hat{x}_j^i(t)) \geq d_{\min}, \forall j \in \mathcal{N}_i, \tag{4c}$$

where $\hat{x}_j^i(t)$ is the estimate of $x_j(t)$ made by vehicle i. In current design of autonomous vehicles, $\hat{x}_j^i$ is estimated based on data from local sensors [2], [19]. In order to account for uncertainties in the estimation, the behavior of an autonomous vehicle tends to be conservative. The system may be trapped in a situation that all vehicles decide to slow down to yield, which is very inefficient. It will be shown in the following discussion that the behaviors of the vehicles can be less conservative when equipped with communication-based conflict resolution strategy. From the system level, less conservativeness implies smaller delay time and larger throughput.

D. Assumptions and Notations

Let x_i^* be the optimal trajectory that does not consider the collision avoidance constraint, e.g.,

$$x_i^* = \arg \min_{\dot{x}_i(t) \in \Gamma(x_i(t))} J(x_i, G_i). \tag{5}$$

If $[x_1^*(t), \ldots, x_N^*(t)] \in \mathcal{X}$ for all t, individual optimum aligns with system optimum. Otherwise, the scenario runs into a generalized Chicken game. Executing the optimal trajectory x_i^* can be considered as the choice of "go." Vehicles will collide with one another if none of them "yields," i.e., executing a trajectory

different than x_i^*. In this paper, it is assumed that the paths of the vehicles are fixed along x_i^*, and the vehicles only adjust their speed profiles to meet the collision avoidance constraints. This assumption is reasonable since vehicles are usually not allowed to change lane at intersections. In the following discussion, let $x_i^*(s)$ be the distance s parameterized path for vehicle i. The speed profile for vehicle i is denoted as $s_i(t)$ which is a mapping from time to distance along the path. Then $x_i^*(s_i(t))$ is the trajectory.

We say that vehicle i goes through the conflict zone C_l if there exists $s \in \mathbb{R}^+$ such that $\mathcal{B}_i(x_i^*(s)) \cap C_l \neq \emptyset$ where $\mathcal{B}_i$ denotes the area occupied by vehicle i at state $x_i^*(s)$. Define the segment on path x_i^* that intersects with the conflict zone C_l as $\mathcal{L}_{i,l} := \{s : \mathcal{B}_i(x_i^*(s)) \cap C_l \neq \emptyset\}$. Hence $\mathcal{L}_{i,l} = \emptyset$ if and only if vehicle i does not go through the conflict zone C_l. Denote the set of indices of conflict zones that vehicle i goes through as $\mathcal{A}_i := \{l : \mathcal{L}_{i,l} \neq \emptyset\}$. Then two vehicles i and j go through a same conflict zone if and only if $\mathcal{A}_i \cap \mathcal{A}_j \neq \emptyset$.

Define a discrete state $\mathcal{S}_i$ for vehicle i, where

1) $\mathcal{S}_i = IL$ if vehicle i is in an incoming lane, and is not the first vehicle in the lane;
2) $\mathcal{S}_i = FIL$ if vehicle i is in an incoming lane, and is the first vehicle in the lane;
3) $\mathcal{S}_i = I$ if vehicle i is at the intersection;
4) $\mathcal{S}_i = OL$ if vehicle i is in an outgoing lane.

Vehicle i may enter the control area with $\mathcal{S}_i = IL$ or FIL. $\mathcal{S}_i$ can transit from IL to FIL, from FIL to I, and from I to OL, i.e., becoming the first in lane, entering the intersection and leaving the intersection. It can leave the control area when $\mathcal{S}_i = OL$. For any vehicle i such that $\mathcal{S}_i = IL$ or OL, its front vehicle is denoted $\mathcal{F}_i$.

In this paper, it is assumed that there is no packet loss or delay in the communication. Moreover, all vehicles are equipped with perfect controllers that can execute the planned trajectories without any error.

III. Communication-Enabled Conflict Resolution

In this section, a conflict resolution mechanism will be discussed to let vehicles better determine the constraint (4c), hence the conflict-free trajectories. An overview of the strategy and system architecture will be provided first, followed by the detailed discussion of each module.

A. Overview of the Strategy

Communication helps the ego vehicle better determine the constraint (4c). Indeed, instead of estimating others' trajectories $\hat{x}_j^i$, what really matters to the ego vehicle is the time that other vehicles occupy the conflict zones. Hence the communication protocol is designed to be that all vehicles should broadcast their estimated time to occupy the conflict zones once they enter a control area of the intersection, e.g., the shaded area in Fig. 2, along with other basic information such as vehicle ID, current state (position, heading, speed and $\mathcal{S}_i$) and time stayed in the control area.

Based on the broadcasted information, the vehicles can achieve a consensus on the passing order and compute desired time slots to pass the conflict zones, which are then taken as temporal constraints on the vehicles' trajectories. This naturally breaks the problem into two parts as shown in Fig. 3(a):

1) decision making – determination of passing order, hence temporal constraints;
2) motion planning – computation of trajectory.

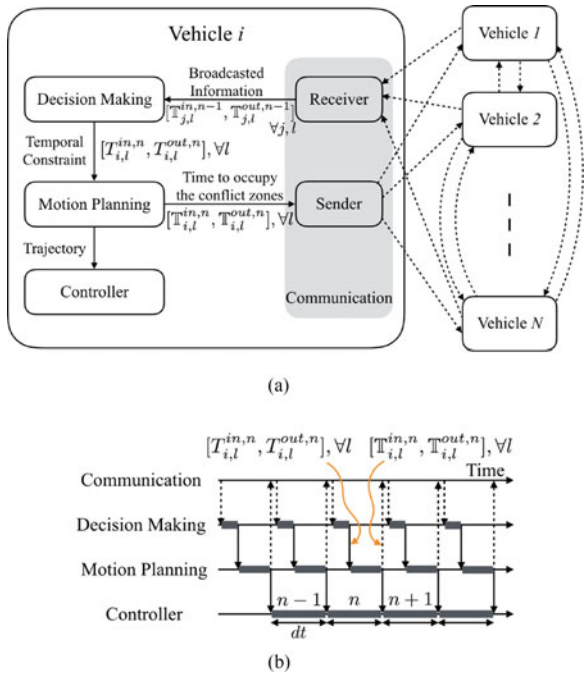

Fig. 3. Architecture of the conflict resolution mechanism. (a) System block diagram. $\mathbb{T}$ represents the broadcasted information. T represents the temporal constraints computed internally. The first subscript is vehicle index. The second subscript is conflict zone index. The second superscript is time step. (b) Time flow of execution.

The time flow and the coordination among different modules are shown in Fig. 3(b). It is assumed that all vehicles are synchronized. At time step $n-1$, the estimated time interval $[\mathbb{T}_{j,l}^{\text{in},n-1}, \mathbb{T}_{j,l}^{\text{out},n-1}]$ for vehicle j to occupy C_l is broadcasted for all j and l. At time step n, vehicle i evaluates all information received from other vehicles and computes the desired time slots to pass the conflict zones in the decision maker, i.e., $[T_{i,l}^{\text{in},n}, T_{i,l}^{\text{out},n}]$ for all l, which are then sent to the motion planner as temporal constraints. After motion planning, the planned trajectory is sent to the controller for execution and the estimated time slots to occupy the conflict zones given the new trajectory, i.e., $[\mathbb{T}_{i,l}^{\text{in},n}, \mathbb{T}_{i,l}^{\text{out},n}]$ for all l, are broadcasted to other vehicles. In Fig. 3(a) and (b), the dashed lines represent information flow. In Fig. 3(b), detailed architecture is only shown for vehicle i, while other vehicles are assumed to have the same architecture. The thick bars in Fig. 3(b) illustrate the busy time in each module. The decision maker runs sequentially with the motion planner, while the two run in parallel with the controller. The sampling time of the system dt represents one cycle in the decision maker and the motion planner. The controller can have a higher sampling frequency.

The decomposition of decision making and motion planning is also adopted in centralized intersection management, where

Algorithm 1: The decision making algorithm in computing the temporal constraints $[T_{i,l}^{in,n}, T_{i,l}^{out,n}], \forall l$ for vehicle i at time step n given information $[\mathbb{T}_{j,l}^{in,n-1}, \mathbb{T}_{j,l}^{out,n-1}], \forall j, l$.

```
Initialize, n = 0;
while in the control area do
    Receive other's information T^{in,n-1}_{j,l}, T^{out,n-1}_{j,l};
    Initialize Y_i = ∅, T^{in,n}_{i,l} = −∞, T^{out,n}_{i,l} = ∞;
    if S_i = IL then
        i yields its front vehicle (Y_i = {F_i});
    end
    for j that has spatial conflicts with i (j ∈ U_i) do
        if j has temporal advantage over i (j ∈ V_i) then
            if ∄Tie(i, j) or j has priority over i then
                i yields j (Y_i = Y_i ∪ {j});
            end
        end
        if i has temporal advantage over j (i ∈ V_j) then
            if ∃Tie(j, i) and j has priority over i then
                i yields j (Y_i = Y_i ∪ {j});
            end
        end
    end
    for j that i yields (j ∈ Y_i) do
        for C_l that both i and j traverse (l ∈ A_i ∩ A_j) do
            T^{in,n}_{i,l} = max{T^{in,n}_{i,l}, T^{out,n-1}_{j,l} + Δ_{S_i}};
        end
    end
    n = n + 1;
end
```

the manager takes the responsibility of decision making and the vehicles takes the responsibility of motion planning [10]. We will show that the system can be self-organized [20], e.g., solving the conflicts without a manager, by distributing the responsibility of decision making to individual vehicles.

In the following part, we discuss the decision making and the motion planning modules in Fig. 3(a) in detail.

B. Decision Making

At time step n, vehicle i needs to compute the desired time interval $[T_{i,l}^{\text{in},n}, T_{i,l}^{\text{out},n}]$ to pass the conflict zones given the broadcasted information $[\mathbb{T}_{j,l}^{\text{in},n-1}, \mathbb{T}_{j,l}^{\text{out},n-1}]$ for all j and l. The basic strategy is that whoever arrives first in a conflict zone goes first.[1] However, this strategy may create deadlocks when one vehicle arrives earlier in one conflict zone, while the other vehicle arrives earlier in another conflict zone. Tie breaking mechanism is needed. In this section, we discuss a general methodology to deal with distributed scheduling with multiple conflict zones, which is summarized in Algorithm 1.

Note that if $\mathcal{S}_i = IL$, it is physically "constrained" by its front vehicle and should yield all vehicles that its front vehicle yields. The decisions when $\mathcal{S}_i = FIL$ or I are the most important as conflicts usually come among vehicles in these two states. When $\mathcal{S}_i = OL$, the vehicle no longer needs to compute the desired time interval. However, its information should be broadcasted in order for the proceeding vehicles to follow the lane safely. In the following discussion, we focus on vehicle i with $\mathcal{S}_i = FIL$ or I.

1) Spatial Conflict: We say that there is a spacial conflict between vehicle i and j if and only if their paths go through a same conflict zone. Consider the scenario shown in Fig. 4(a), where nine vehicles locate in a six-way intersection. The shaded area denotes the six conflict zones. By adding links between any pair of vehicles that have spatial conflicts, we formulate an undirected graph as shown in Fig. 4(b), where every node represents one vehicle. Whenever there is a link between two vehicles, we need to decide which vehicle goes first. In other words, the undirected graph needs to be transformed into a directed graph as shown in Fig. 4(d) such that the passing order is decided as: a vehicle should yield its successor. Denote the set of vehicles that have spacial conflicts with vehicle i as

$$\mathcal{U}_i := \{j | \mathcal{S}_j = FIL \text{ or } I, \mathcal{A}_i \cap \mathcal{A}_j \neq \emptyset\}. \tag{6}$$

The graph in Fig. 4(b) is denoted as $\mathcal{U} := \cup_i \cup_{j \in \mathcal{U}_i} (i, j)$, where (i, j) represents a link between i and j. There is an undirected link between any i and j such that $j \in \mathcal{U}_i$. In literature, this graph is identified as a conflict graph [6]. Finding the optimal passing order regarding the conflict graph is NP-hard. The method presented in this paper is a heuristic approach which finds one feasible passing order in linear time.

2) Temporal Advantage: At time step n, we say that vehicle $j \in \mathcal{U}_i$ has temporal advantage over vehicle i, if one of the following conditions holds,

1) $\mathcal{S}_j = I, \mathcal{S}_i = FIL$ and vehicle i leaves some conflict zones later than vehicle j enters, i.e.,
$$\exists l \in \mathcal{A}_i \cap \mathcal{A}_j, \mathbb{T}_{i,l}^{\text{out},n-1} > \mathbb{T}_{j,l}^{\text{in},n-1}. \tag{7}$$
2) $\mathcal{S}_j = FIL, \mathcal{S}_i = I$ and vehicle j leaves all conflict zones earlier than vehicle i enters, i.e.,
$$\forall l \in \mathcal{A}_i \cap \mathcal{A}_j, \mathbb{T}_{j,l}^{\text{out},n-1} \leq \mathbb{T}_{i,l}^{\text{in},n-1}. \tag{8}$$
3) $\mathcal{S}_j = \mathcal{S}_i = FIL$ or I and vehicle j enters some conflict zones earlier than vehicle i, i.e.,
$$\exists l \in \mathcal{A}_i \cap \mathcal{A}_j, \mathbb{T}_{j,l}^{\text{in},n-1} \leq \mathbb{T}_{i,l}^{\text{in},n-1}. \tag{9}$$

According to the above definitions, for vehicle i and j with different discrete states, either i or j should have temporal advantage over the other. If vehicle i and j have the same discrete state, it is possible that both i and j have temporal advantages over the other. Denote the set of vehicles that have temporal advantages over vehicle i at time step n as $\mathcal{V}_i^n$. The superscript n in the following discussion is ignored for simplicity. Obviously, $\mathcal{V}_i \subset \mathcal{U}_i$. And $\mathcal{V} := \cup_i \cup_{j \in \mathcal{V}_i} (i, j)$ is a directed graph as shown in Fig. 4(c), where there is a directed link from any i to any $j \in \mathcal{V}_i$. However, there are cycles among nodes with the same discrete state, e.g., between nodes 6 and 7, as well as among nodes with different discrete states, e.g., among nodes 6, 7 and 3. If vehicles yield each other according to the graph, there are deadlocks.

3) Tie Breaking: For any vehicle i and vehicle $j \in \mathcal{V}_i$, it is called a tie if

[1]Note that this is different from the strategies discussed in [18], which only considers the arrival time at the intersection.

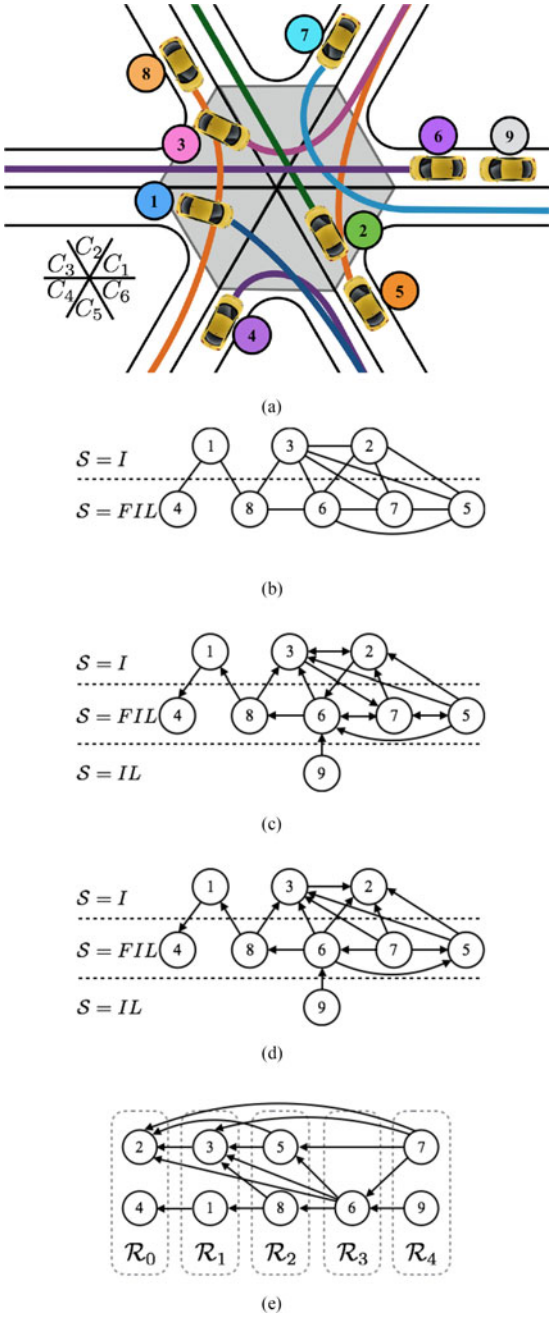

Fig. 4. The conflict graphs. (a) The scenario: nine vehicles in a six-way intersection. (b) Graph of spacial conflicts $\mathcal{U}$ at one time step. Link $(i,j) \in \mathcal{U}$ implies that vehicle j has spacial conflicts with vehicle i at some conflict zone. (c) Graph of temporal advantages $\mathcal{V}$ at one time step. Link $(i,j) \in \mathcal{V}$ implies that vehicle j has temporal advantages over vehicle i. (d) Graph of passing order $\mathcal{Y}$ at one time step. Link $(i,j) \in \mathcal{Y}$ implies that vehicle i yields vehicle j. (e) Convergence of the graph $\mathcal{Y}$ through time. The passing order converges at time 0 for leaf nodes in $\mathcal{R}_0$, then for nodes with depth 1 in $\mathcal{R}_1$ at time 1, and so on.

1) $\mathcal{S}_i = \mathcal{S}_j$ and there exists a sequence of vehicles $\{q_m\}_1^M$ with $q_1 = i$, $q_M = j$, $M \geq 2$ and $\mathcal{S}_{q_m} = \mathcal{S}_i$ for all m such that $q_m \in \mathcal{V}_{q_{m+1}}$ for $m = 1, \ldots, M-1$.

or 2) $\mathcal{S}_i = I$, $\mathcal{S}_j = FIL$ and there exists a sequence of vehicles $\{q_m\}_1^M$ with $q_1 = i$, $q_M = j$ and $M \geq 2$ such that $q_m \in \mathcal{V}_{q_{m+1}}$ for $m = 1, \ldots, M-1$.

Let $\text{Tie}(i,j)$ denote all these sequences. The relationship in a tie is neither symmetric nor exclusive, i.e., $\exists \text{Tie}(i,j)$ neither implies $\exists \text{Tie}(j,i)$ nor $\nexists \text{Tie}(j,i)$. In Fig. 4(c), there is a tie from node 5 to node 6 via the sequence $\{5,7,6\}$, but not a tie from node 6 to node 5 since $5 \notin \mathcal{V}_6$. There is a tie from node 2 and node 3 via the sequence $\{2,3\}$ and a tie from node 3 and node 2 via the sequence $\{3,2\}$.

We assume that each vehicle has a unique priority score P, which is related to the weights in (1a). For example, the priority score of a fire truck is higher than that of a passenger vehicle. We say vehicle i has priority over j if there exists a sequence in $\text{Tie}(i,j)$ such that $P(i) > P(k)$ for all $k \neq i$ in the sequence. Vehicles in the intersection should always have priority over vehicles in the incoming lanes. For vehicles in the same discrete state, the order implied by the priority score should not change over time. If vehicle i has priority over $j \in \mathcal{V}_i$, instead of i yielding j, vehicle j should yield vehicle i, although vehicle j has temporal advantage. For example, in Fig. 4, we identify the score P with the vehicle index. Since node 5 has priority in the sequence $\{5,7,6\}$, the link from 5 to 6 is reversed in Fig. 4(d). Since there is a tie between node 2 and node 6, the link from 2 to 6 is also reversed in Fig. 4(d).

4) Passing Sequence: After tie breaking, all those remaining links for vehicle i represent the set of vehicles that vehicle i decides to yield at time step n, which is denoted by $\mathcal{Y}_i^n$. The superscript n in the following discussion is ignored for simplicity. Indeed, $\mathcal{Y} := \cup_i \cup_{j \in \mathcal{Y}_i} (i,j)$ is a directed graph as shown in Fig. 4(d), which encodes the order for the vehicles to pass the intersection. It is not necessary for vehicle i to construct the whole graphs $\mathcal{U}$ and $\mathcal{V}$ to determine $\mathcal{Y}_i$. For example, vehicle 4 in Fig. 4 only needs to compute $\mathcal{U}_4$ and $\mathcal{V}_4$ locally to determine that $\mathcal{Y}_4 = \emptyset$. Those locally decisions form the passing sequence globally. In the extreme case, the passing order follows the order specified by the priority score. If all vehicles agree on the above tie breaking mechanism, they can solve the conflicts even if the vehicles plan and control their motions differently.

According to Algorithm 1, if $\mathcal{S}_i = IL$, the vehicle i yields its front vehicle, i.e., $\mathcal{Y}_i = \{\mathcal{F}_i\}$, as shown by vehicle 9 in Fig. 4(d). If vehicle i decides to yield vehicle j, then for all $l \in \mathcal{A}_i \cap \mathcal{A}_j$, we set

$$T_{i,l}^{\text{in},n} \geq \mathbb{T}_{j,l}^{\text{out},n-1} + \Delta_{\mathcal{S}_i}, \tag{10}$$

where $\Delta_{\mathcal{S}_i}$ is a margin to increase the robustness of the algorithm, which is chosen such that $\Delta_{IL} > \Delta_{FIL} > \Delta_I$. Δ_{IL} is chosen to be greater than Δ_{FIL} in order to let the leading vehicles have temporal advantages over vehicles in the middle of other lanes. For example, vehicle 7 will have temporal advantage over vehicle 9 in Fig. 4(d). Similarly, Δ_{FIL} is chosen to be greater than Δ_I.

C. Motion Planning Under Temporal Constraints

At time step n, given the temporal constraint $[T_{i,l}^{\text{in},n}, T_{i,l}^{\text{out},n}]$ specified by the decision maker, the motion planning problem

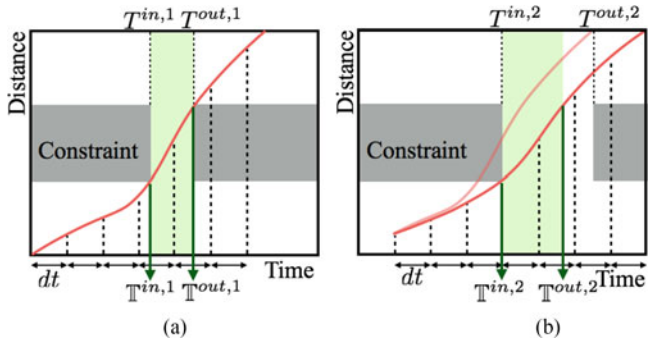

Fig. 5. Speed profile planning and execution. T is the constraint determined by Algorithm 1. $\mathbb{T}$ is the expected time to occupy the conflict zone given the planned speed profile. (a) Time step 1. (b) Time step 2.

(4) for vehicle i becomes,

$$\min_{s_i} J(x_i^*(s_i), G_i), \tag{11a}$$

$$\text{s.t. } \dot{x}_i^*(s_i)\dot{s}_i \in \Gamma(x_i^*(s_i)), \tag{11b}$$

$$s_i(t) \notin \mathcal{L}_{i,l}, \forall t \notin [T_{i,l}^{\text{in},n}, T_{i,l}^{\text{out},n}], \forall l, \tag{11c}$$

where $s_i(t)$ is the speed profile that needs to be optimized. Constraint (11c) specifies that the vehicle should only enter the conflict zone C_l in the time interval $[T_{i,l}^{\text{in},n}, T_{i,l}^{\text{out},n}]$. For simplicity, the constraint for car following is omitted in presentation (but is included in problem solution). A method to efficiently solve problem (11) via temporal optimization is discussed in [21]. In this paper, it is assumed that vehicles can take unbounded deceleration, which is reasonable when vehicle speed is low. Moreover, since $T_{i,l}^{\text{out},n} = \infty$ by Algorithm 1, there is always a solution of problem (11). In the worst case, vehicle i just stops immediately. Indeed, the vehicles do not necessarily take unbounded deceleration as will be demonstrated in Section V, since the conflicts are resolved before they enter the intersection. The feasibility of problem (11) under bounded deceleration will be studied in the future.

Given the optimal solution s_i^* of (11), the expected time slot $[\mathbb{T}_{i,l}^{\text{in},n}, \mathbb{T}_{i,l}^{\text{out},n}]$ for vehicle i to occupy the conflict zone C_l is computed as

$$\mathbb{T}_{i,l}^{\text{in},n} := \min_{s_i^*(t)\in\mathcal{L}_{i,l}} t \geq T_{i,l}^{\text{in},n},\ \mathbb{T}_{i,l}^{\text{out},n} := \max_{s_i^*(t)\in\mathcal{L}_{i,l}} t \leq T_{i,l}^{\text{out},n}. \tag{12}$$

If $\mathcal{L}_{i,l} = \emptyset$, then $\mathbb{T}_{i,l}^{\text{in},n} := \infty$ and $\mathbb{T}_{i,l}^{\text{out},n} := -\infty$. If vehicle i has entered or left C_l, then $\mathbb{T}_{i,l}^{\text{in},n}$ and $\mathbb{T}_{i,l}^{\text{out},n}$ are chosen as the time that it entered or left C_l respectively.

The planning and execution of the speed profile are illustrated in Fig. 5. The vertical axis denotes the distance along the path, while the horizontal axis is time. There is only one conflict zone, hence one constraint (shown in gray). For simplicity, the subscripts are ignored. At the first time step, given the constraint $[T^{\text{in},1}, T^{\text{out},1}]$, the optimal speed profile in problem (11) is computed, shown as the curve in Fig. 5(a), which is then sampled in the time domain and sent to the controller for execution. Given the trajectory, the expected time $[\mathbb{T}^{\text{in},1}, \mathbb{T}^{\text{out},1}]$ to occupy the conflict zone (shaded) is computed according to (12) and broadcasted. In this example, $[\mathbb{T}^{\text{in},1}, \mathbb{T}^{\text{out},1}] = [T^{\text{in},1}, T^{\text{out},1}]$, i.e., the constraint is tight. Based on all broadcasted information, the constraint is updated as shown in Fig. 5(b). A new trajectory is then computed, which starts from the next sampling point at the next time step on the previous trajectory, and is different from the previous trajectory. At the second time step, the new trajectory is sampled and sent to the controller, while the expected time to occupy the conflict zone is computed and broadcasted. In this example, $[\mathbb{T}^{\text{in},2}, \mathbb{T}^{\text{out},2}] \subset [T^{\text{in},2}, T^{\text{out},2}]$, i.e., the constraint is not tight. Then the above process repeats.

IV. Theoretical Results

In this section, we will show that the proposed strategy solves the conflicts safely and efficiently in real time by verifying the five requirements discussed in the introduction section. The physical feasibility is verified in the motion planning part. Proposition 2 ensures that the passing order is completely determined. Proposition 3 states that there is no deadlock for any pair of vehicles that pass through a same conflict zone at every time step. Then Proposition 4 shows that a stable consensus on conflict-resolution can be reached in finite time steps.

Lemma 1: For any j that i yields, i.e., $j \in \mathcal{Y}_i$, there is a directed path from i to j in the graph $\mathcal{V}$ of temporal advantage.

Proof: For any j such that $j \in \mathcal{Y}_i$, if j has the temporal advantage, i.e., $j \in \mathcal{V}_i$, then there is a link from i to j. Otherwise, i has the temporal advantage, i.e., $i \in \mathcal{V}_j$. According to lines 15–17 in Algorithm 1, there exists $\text{Tie}(j, i)$, hence a directed path from i to j. Thus the claim is true. ■

Proposition 2 (Completeness): For any j that has spacial conflicts with i, at least one statement is true: "i yields j" or "j yields i." In other words, $j \in \mathcal{U}_i$ implies $j \in \mathcal{Y}_i$ or $i \in \mathcal{Y}_j$.

Proof: We will prove by cases. For any j that has spacial conflicts with i, i.e., $j \in \mathcal{U}_i$, either j or i or both of them should have the temporal advantage over the other, i.e., $j \in \mathcal{V}_i$ or $i \in \mathcal{V}_j$. Consider the case $j \in \mathcal{V}_i$. When car i runs Algorithm 1, it will yield to car j, i.e., $j \in \mathcal{Y}_i$, if $\nexists Tie(i, j)$ or j has priority over i, according to lines 10–12 of Algorithm 1. The case where this is not true is that $\exists Tie(i, j)$ and i has priority over j, which will assign i to $\mathcal{Y}_j$ when car j executes Algorithm 1 on line 15–17. Similar arguments follow for the case $i \in \mathcal{V}_j$. Hence $j \in \mathcal{Y}_i$ or $i \in \mathcal{Y}_j$. ■

Proposition 3 (Deadlock-Free): There is no cycle in the directed graph $\mathcal{Y}$ of passing order.

Proof: Suppose there is a cycle in the directed graph $\mathcal{Y}$, i.e., there exists a list of nodes $\{q_m\}_1^M$ for $M \geq 2$ such that $q_1 \in \mathcal{Y}_{q_M}$ and $q_m \in \mathcal{Y}_{q_{m+1}}$ for $m = 1, \ldots, M-1$. For simplicity, we identify q_{i+kM} with q_i for any $i, k \in \mathbb{N}$. If $\mathcal{S}_{q_m}$ are the same for all m, find m^* such that q_{m^*} has the highest priority score. If $\mathcal{S}_{q_m}$ are not the same for all m, find m^* such that $S_{q_{m^*-1}} = FIL$ and $S_{q_{m^*}} = I$. In both cases, q_{m^*} has a higher priority score than q_{m^*-1}. When q_{m^*} runs Algorithm 1, since q_{m^*} has higher priority, $q_{m^*-1} \in \mathcal{Y}_{q_{m^*}}$ can hold only if the conditions in lines 9–10 are satisfied, i.e., $q_{m^*-1} \in \mathcal{V}_{q_{m^*}}$. However, consider the chain $q_{m^*} \leftarrow q_{m^*+1} \ldots \leftarrow q_{M+m^*-1}$ in the graph $\mathcal{Y}$. Lemma 1 implies that there is a directed path in the graph $\mathcal{V}$ from any node in the chain to its successor. Hence there is a directed path from

q_{M+m^*-1} to q_{m^*} in the graph $\mathcal{V}$, which implies the existence of Tie(q_{m^*}, q_{m^*-1}). However, this violates the condition in line 10 in Algorithm 1. In this way, q_{m^*-1} cannot belong to $\mathcal{Y}_{q_{m^*}}$. A contradiction is reached! Hence there is no cycle in the directed graph $\mathcal{Y}$. ■

Proposition 4 (Finite Time Convergence): If S_i and $\mathcal{U}_i$ remain the same for all i for more than N time steps, then $\mathcal{Y}_i^n$ and $[\mathbb{T}_{i,l}^{\text{in},n}, \mathbb{T}_{i,l}^{\text{out},n}]$ converge in at most N steps to $\mathcal{Y}_i^*$ and $[\mathbb{T}_{i,l}^{\text{in}*}, \mathbb{T}_{i,l}^{\text{out}*}]$ such that

$$\mathbb{T}_{i,l}^{\text{in}*} \geq \mathbb{T}_{j,l}^{\text{out}*} + \Delta_{\mathcal{S}_i}, \forall l, \forall j \in \mathcal{Y}_i^*. \tag{13}$$

Proof: At time step n, based on the information $[\mathbb{T}_{i,l}^{\text{in},n-1}, \mathbb{T}_{i,l}^{\text{out},n-1}]$, the graph $\mathcal{Y}^n := \cup_i \cup_{j \in \mathcal{Y}_i^n} (i,j)$ is determined. Then the time slots $[\mathbb{T}_{i,l}^{\text{in},n}, \mathbb{T}_{i,l}^{\text{out},n}]$ for all i and l are computed and broadcasted. We will prove the statement by induction. Suppose the current time step is 0. The algorithm will converge in the first time step for agents that are leaf nodes of $\mathcal{Y}^0$: the vehicles that can go through the intersection first and does not have to yield to any other vehicle, then for vehicles that only yield to these, at depth 1 of $\mathcal{Y}^1$, and so on.

Denote all leaf nodes as $\mathcal{R}_0 := \{i | \mathcal{Y}_i^0 = \emptyset\}$, and nodes with n-depth as

$$\mathcal{R}_n := \{i \notin \cup_{k=0}^{n-1} \mathcal{R}_k | \mathcal{Y}_i^n \subset \cup_{k=0}^{n-1} \mathcal{R}_k\}. \tag{14}$$

Vehicles in $\mathcal{R}_n$ only yield vehicles in $\mathcal{R}_0, \ldots, \mathcal{R}_{n-1}$. An example of $\mathcal{R}_n$ for $n = 0, \ldots, 4$ is shown in Fig. 4(e).

Claim that for any $i \in \mathcal{R}_0$ and for all $t > 0$,

$$\mathcal{Y}_i^t = \mathcal{Y}_i^0 = \emptyset, [\mathbb{T}_{i,l}^{\text{in},t}, \mathbb{T}_{i,l}^{\text{out},t}] = [\mathbb{T}_{i,l}^{\text{in},0}, \mathbb{T}_{i,l}^{\text{out},0}], \forall l.$$

Consider j that has spacial conflicts with i, i.e., $j \in \mathcal{U}_i$. Since $j \notin \mathcal{Y}_i^0$, then $i \in \mathcal{Y}_j^0$ according to Proposition 2, i.e., vehicle j yields vehicle i in step 0. Hence vehicle j cannot have temporal advantage over vehicle i at step 1 and all remaining steps, which implies that $j \notin \mathcal{V}_i^t$ for all j and $t > 0$. Thus there does not exist any directed path from i to j in the graph $\mathcal{V}^t$. So $\mathcal{Y}_i^t = \emptyset$ according to Lemma 1. For $i \in \mathcal{R}_0$, $[\mathbb{T}_{i,l}^{\text{in},t}, \mathbb{T}_{i,l}^{\text{out},t}] = [\mathbb{T}_{i,l}^{\text{in},0}, \mathbb{T}_{i,l}^{\text{out},0}]$ for all t since the temporal constraints remain the same, and hence the optimal speed profiles remain the same.

Suppose for any $i \in \mathcal{R}_{n-1}$ and for all $t > n-1$,

$$\mathcal{Y}_i^t = \mathcal{Y}_i^{n-1}, [\mathbb{T}_{i,l}^{\text{in},t}, \mathbb{T}_{i,l}^{\text{out},t}] = [\mathbb{T}_{i,l}^{\text{in},n-1}, \mathbb{T}_{i,l}^{\text{out},n-1}], \forall l.$$

Claim that for any $i \in \mathcal{R}_n$ and for all $t > n$,

$$\mathcal{Y}_i^t = \mathcal{Y}_i^n, [\mathbb{T}_{i,l}^{\text{in},t}, \mathbb{T}_{i,l}^{\text{out},t}] = [\mathbb{T}_{i,l}^{\text{in},n}, \mathbb{T}_{i,l}^{\text{out},n}], \forall l.$$

Consider j at depth $k < n$ that has spacial conflicts with i, i.e., $j \in \bigcup_{k=0}^{n-1} \mathcal{R}_k \cap \mathcal{U}_i$. By definition (14), $\mathcal{Y}_j^n \subset \bigcup_{k=0}^{n-1} \mathcal{R}_k$, and $i \notin \mathcal{Y}_j^n$. By Proposition 2, $j \in \mathcal{Y}_i^n$. Hence vehicle i cannot hold temporal advantage over any vehicle j for any step $t > n$. Furthermore, there does not exist any directed path from j to i in the graph $\mathcal{V}^t$. So $j \in \mathcal{Y}_i^t$. Then consider j at depth $k \geq n$ that has spacial conflicts with i, i.e., $j \in \mathcal{U}_i \setminus \bigcup_{k=0}^{n-1} \mathcal{R}_k$. By definition (14), $j \notin \mathcal{Y}_i^n$. Then $i \in \mathcal{Y}_j^n$ according to Proposition 2, i.e., vehicle j yields vehicle i in step n. Hence any vehicle j cannot hold temporal advantage over vehicle i in step $n+1$ and all remaining steps. Furthermore, there does not exist any directed path from i to j in the graph $\mathcal{V}^t$ for $t > n$. Thus $j \notin \mathcal{Y}_i^t$. Then $\mathcal{Y}_i^t = \mathcal{Y}_i^n = \cup_{k=0}^{n-1} \mathcal{R}_k \cap \mathcal{U}_i$. For $i \in \mathcal{R}_n$, $[\mathbb{T}_{i,l}^{\text{in},t}, \mathbb{T}_{i,l}^{\text{out},t}] = [\mathbb{T}_{i,l}^{\text{in},n}, \mathbb{T}_{i,l}^{\text{out},n}]$ for all $t > n$ since the temporal constraints remain the same after step $n-1$.

By induction, $\mathcal{Y}_i^n$ and $[\mathbb{T}_{i,l}^{\text{in},n}, \mathbb{T}_{i,l}^{\text{out},n}]$ converge to some $\mathcal{Y}_i^*$ and $[\mathbb{T}_{i,l}^{\text{in}*}, \mathbb{T}_{i,l}^{\text{out}*}]$ such that (13) holds. The steps to convergence equal to the maximum depth n of $\mathcal{R}_n$, which is bounded by N, the total number of vehicles in the system. ■

Proposition 4 implies that if the sampling time dt is short enough compared to the time needed between two transitions of S_i's, the system can still reach consensus when S_i's are changing. Nonetheless, after a transition of some S_i, the system needs several steps to settle down. The consistency of the passing orders $\mathcal{Y}^n$ considering those transitions is more intricate to prove, which will be left as future work. Indeed, the consistency is demonstrated in simulation.

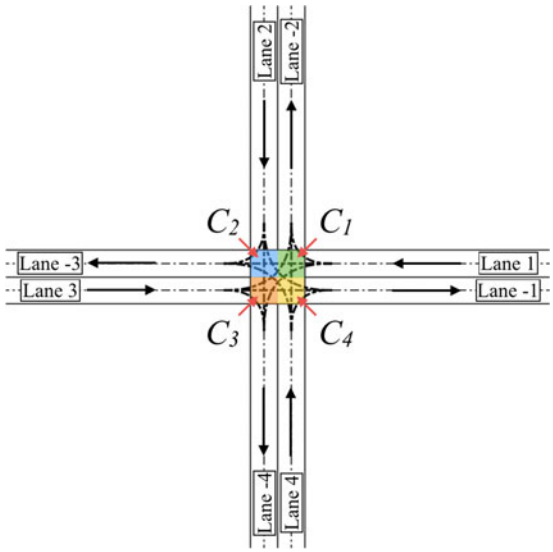

Fig. 6. The simulation environment.

V. Simulation Results

In this section, we illustrate the performance of the proposed distributed conflict resolution mechanism through extensive traffic simulations. The sampling time in the system is chosen to be $dt = 0.1$ s. The robustness margins are chosen as $\Delta_1 = 0.5$ s, $\Delta_2 = 0.3$ s and $\Delta_3 = 0.1$ s. The priority score P for a vehicle is chosen to be the time that the vehicle stays in the control area. If there is a tie, then the vehicle with smaller ID has the priority. The cost function of the vehicle penalizes 1) the deviation from a target speed, 2) the magnitude of acceleration or deceleration, 3) the magnitude of jerk, and 4) the time spent in every conflict zone. The target speed varies for different vehicles.

The simulation environment is a narrow four-way intersection as shown in Fig. 6. There is only one incoming lane and one outgoing lane in every direction. Four conflict zones are identified. The control area is the whole graph. For any $i \neq j$, there is a path from lane i to lane $-j$. Hence there are 12 different paths as shown in Fig. 6. Right turn paths only go through one conflict zone. Straight paths go through two conflict zones. Left turn paths go through all four conflict zones (a vehicle is treated as a 2D object instead of a point).

TABLE I
CONDITIONS IN THE CASE STUDY

Vehicle ID	Target Speed	From	To	Enter Time
1	10 m/s	Lane 1	Lane -3	0.2 s
2	12.5 m/s	Lane 2	Lane -1	0.2 s
3	10.75 m/s	Lane 3	Lane -2	0.2 s
4	17.75 m/s	Lane 4	Lane -2	0.6 s

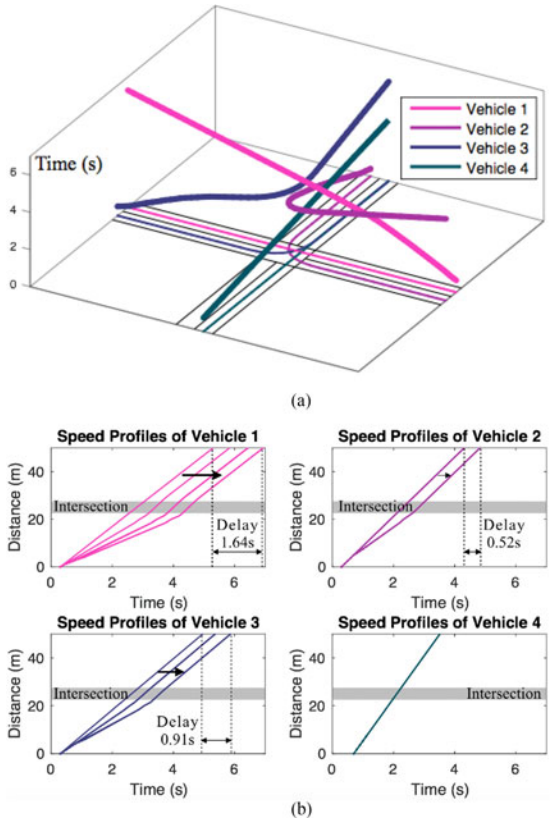

Fig. 7. Speed profiles and trajectories in the case study. (a) Executed trajectories in the time-augmented space. (b) Planned speed profiles in different time steps.

In the following discussion, a microscopic case study will be presented first followed by the result of macroscopic traffic simulation.

A. Microscopic Case Study

In the case study, there are four vehicles. The conditions of the vehicles (target speed, current lane, target lane, and time to enter the control area) are shown in Table I. The paths and the executed trajectories are shown in the time-augmented space in Fig. 7(a). The planned speed profiles in different time steps are shown in Fig. 7(b). The left most speed profile in every subplot is the traffic-free speed profile and the others are the replanned speed profiles given the temporal constraints. Fig. 8 shows the expected time intervals (the colored thick bars) for the vehicles to occupy the conflict zones. The thin vertical line indicates the current time. Vehicles 1, 2 and 3 enter the control area at the same time. According to the traffic-free speed profiles, there are temporal conflicts between vehicle 1 and vehicle 3 in conflict zones 1 and 2, and between vehicle 2 and vehicle 3 in all conflict

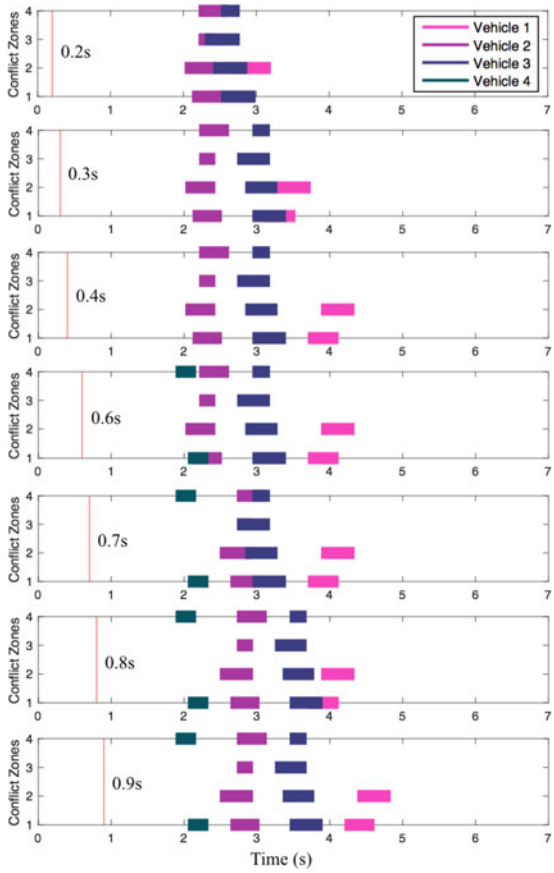

Fig. 8. Conflict resolution in the case study. The scenario in 0.5 s is omitted since it is the same as the scenario in 0.4 s.

zones. Since vehicle 2 has the temporal advantage over vehicle 3, vehicle 3 yields vehicle 2. Similarly, vehicle 1 yields vehicle 3. It takes two time steps for the conflicts to be resolved. At 0.6 s, vehicle 4 enters, which creates new conflicts. The system settles down after 3 time steps as shown in Fig. 8, which verifies Proposition 4. The planned speed profiles change accordingly as shown in Fig. 7(b). The right most speed profile in each subplot is the executed speed profile.

B. Macroscopic Traffic Simulation

1) Traffic: In the macroscopic traffic simulation, the traffic is generated at every incoming lane by a Poisson distribution where the density λ is chosen to be 0.5, 0.25 and 0.1, which implies that on average, vehicles arrive every 2 s, 4 s and 10 s. Two groups of traffic are generated. In the first group, 50% of vehicles go straight, 25% turn right and 25% turn left. In the second group, all vehicles go straight. The second group is introduced to create a relatively fair comparison among performances under distributed strategies and performances under traffic lights. Since we don't have left turn lane or left turn light, when a vehicle wants to turn left, it will block all the vehicles behind, thus significantly increase the delay time. The desired longitudinal speed v_i^r of the vehicle i follows from a uniform distribution from 7.5 m/s to 15 m/s. The video for "all direction" traffic with $\lambda = 0.5$ under the proposed strategy can be found at `https://youtu.be/DbsZstmXMB8`.

2) Comparison: The proposed mechanism is compared against other mechanisms as listed below.

Case 1 (3D): 3D intersection such as overpass without connectivity. In this case, there is no conflict among vehicles at the intersection. Since the delay is only caused by car following, the simulation result provides a lower bound for the delay time and an upper bound for the throughput.

Case 2 (NC): Unmanaged 2D intersection without connectivity. Vehicles are able to see vehicles from other directions when approaching the intersection. Then the best strategy for the autonomous vehicle is that: if there is no other vehicles from other directions or other vehicles are too far from the intersection (i.e., there is no temporal conflict even if the other vehicle accelerates with maximum acceleration), cross the intersection without stop; if there are other vehicles from other directions that are close to the intersection, stop and "first stop first go." The delay time in this case is upper bounded by the delay time in the case of a four-way-stop intersection.

Case 3 (TL-5): 2D intersection with traffic light that changes every 5 s without connectivity. For example, the traffic light for the horizontal direction (lane 1 and lane 3) is green from 0 s to 5 s and red from 5 s to 10 s while the traffic light for the vertical direction (lane 2 and lane 4) is red from 0 s to 5 s and green from 5 s to 10 s.

Case 4 (TL-10): 2D intersection with traffic light that changes every 10 s without connectivity.

Case 5 (MP-IP): 2D intersection with the maximum progression intersection protocol (MP-IP) [18]. Vehicles broadcast their intentions and estimated time slots to occupy the conflict zones. Conflicting vehicles can make concurrent progress inside the intersection, though low priority vehicles need to yield high priority vehicles, i.e., entering the conflict zones after the high priority vehicles leave. In the simulation, the priority is determined by the priority score P.[2]

Case 6 (AMP-IP): 2D intersection with the advanced maximum progression intersection protocol (AMP-IP) [18]. In addition to MP-IP, the lower priority vehicles are allowed to cross and clear the conflict zone before the earliest possible arrival of the higher-priority vehicle to that conflict zone.

In Cases 1 to 4, there is no communication among vehicles and the vehicles are equipped with adaptive cruise control for car following. In Cases 5 and 6, vehicles communicate with one another. The two protocols only determine the passing order, not the vehicle trajectories. In the simulation, the vehicles under the two cases adopt the motion planning algorithm discussed in Section III-C. The temporal constraints are determined by (10) according to the passing order. To create a fair comparison, the adaptive cruise control algorithm is integrated into the motion planning algorithm. At each time step, the output of the adaptive cruise control module will be treated as an upper bound on vehicle's acceleration, which is added to the optimization (11).

[2]Note that the control area is topologically equivalent to the intersection discussed in [18]. Every incoming lane, every outgoing lane and every conflict zone correspond to an intersection cell, though the cell for incoming lane or outgoing lane can be occupied by more than one vehicle. Hence the priority policy determined by P is consistent with the first-come-first-serve policy discussed in [18].

The following statistics are compared: 1) the average delay time and 2) the throughput in certain time horizon.

3) Average Delay: The delay time of a vehicle is computed as the difference between the actual time and the traffic-free time for the vehicle to travel cross the control area as shown in Fig. 7(b). The average delay of all vehicles traveled in the control area in 10 min under different mechanisms are shown in Table II. The first number in the table is the mean and the second number is the standard deviation. The proposed strategy always outperforms other mechanisms except for the case with 3D intersection which provides a theatrical lower bound of this problem. When the traffic density is low, the performances of Case 2 (without communication) and Cases 5 and 6 (with communication) are similar to the performance of the proposed method, which outperforms the cases with traffic lights. When the traffic density goes up, the performance of Case 2 gets worse dramatically as it almost functions as a stop sign mechanism. The proposed method still outperforms the cases with traffic lights (Cases 3 and 4) since it is more flexible. For example, in the proposed mechanism, four simultaneous right turns are allowed, while in the traffic light case, at most two simultaneous right turns can be tolerated.

The proposed method always outperforms Cases 5 and 6. Though more parallelism inside the intersection area (i.e., allowing more vehicles to cross the intersection at the same time) has been introduced in these two cased compared to Case 2, the rigidity of the priority queue (which does not adjust in real time) limits their performances. For example, consider the case study in Section V-A. Since vehicle 4 arrives later than others, it has to wait for others according to MP-IP in Case 5. Even with AMP-IP in Case 6, vehicle 4 wouldn't be able to cut in front of vehicle 2, since it does not leave conflict zone 1 before vehicle 2 enters. Hence high-speed vehicles in Cases 5 and 6 experience larger delay compared to those in the proposed method, where they can cut into the queue only causing other vehicles to slow down slightly. Moreover, the average delay goes up from 8.8 s to 52.4 s in Case 5 with "straight only" traffic $\lambda = 0.5$ if the motion planning algorithm is replaced with only adaptive cruise control (ACC). Since the travel time in the intersection is not penalized in ACC, vehicles tend to stop right before the intersection and consequently take longer time to traverse the intersection (as their acceleration is bounded) than they do when they optimize their speed profiles to slow down before approaching the intersection and then speed up to pass the intersection at full speed. Hence the efficiency of the proposed algorithm benefits from both the decision making module (determination of efficient passing order) and the motion planning module (temporal optimization) as well as their integration.

The histograms of the delay times for all vehicles under various mechanisms in 10 min for "straight only" traffic when $\lambda = 0.5$ are shown in Fig. 9(a). The delay times in the proposed mechanism are more concentrated in the intervals with smaller delays than all other cases except for Case 1 (3D). In the same traffic scenario, the distributions of the maximum deceleration of the vehicles under different mechanisms are shown in Fig. 9(b). Vehicles with the proposed strategy have smaller decelerations (worst case -8.3 m/s^2, average -2.7 m/s^2) since they can adjust their speed profile in advance. The maximum decelerations of

TABLE II
THE AVERAGE DELAY TIME FOR TRAFFIC IN 10 MIN

Traffic	λ	Case 1: 3D	Case 2: NC	Case 3: TL-5	Case 4: TL-10	Case 5: MP-IP	Case 6: AMP-IP	Proposed
All directions	0.5	0.5 s ± 0.8 s	135.6 s ± 78.6 s	53.2 s ± 30.6 s	57.8 s ± 34.4 s	31.2 s ± 19.7 s	20.5 s ± 13.2 s	11.4 s ± 7.0 s
	0.25	0.2 s ± 0.4 s	2.9 s ± 2.8 s	3.2 s ± 2.6 s	5.2 s ± 3.8 s	1.9 s ± 1.7 s	1.2 s ± 1.2 s	0.5 s ± 0.7 s
	0.1	0.1 s ± 0.2 s	0.4 s ± 0.7 s	2.1 s ± 2.0 s	3.8 s ± 3.9 s	0.4 s ± 0.6 s	0.3 s ± 0.6 s	0.2 s ± 0.3 s
Straight only	0.5	0.5 s ± 0.8 s	134.8 s ± 81.0 s	22.0 s ± 14.1 s	29.3 s ± 17.6 s	8.8 s ± 6.4 s	6.3 s ± 4.7 s	4.3 s ± 3.3 s
	0.25	0.2 s ± 0.6 s	9.2 s ± 9.3 s	2.8 s ± 2.3 s	4.4 s ± 3.9 s	2.5 s ± 2.9 s	2.2 s ± 2.9 s	2.1 s ± 2.7 s
	0.1	0.1 s ± 0.4 s	0.5 s ± 0.7 s	1.9 s ± 1.8 s	3.9 s ± 3.9 s	0.3 s ± 0.5 s	0.3 s ± 0.5 s	0.3 s ± 0.5 s

TABLE III
THE TRAFFIC THROUGHPUT (# OF VEHICLES) IN 10 MIN

Traffic	λ	Case 1: 3D	Case 2: NC	Case 3: TL-5	Case 4: TL-10	Case 5: MP-IP	Case 6: AMP-IP	Proposed
All directions	0.5	1170	660	984	965	1023	1099	1139
	0.25	590	589	589	587	590	590	590
	0.1	230	230	230	228	230	230	230
Straight only	0.5	1206	641	1121	1091	1166	1186	1199
	0.25	599	597	595	589	599	599	599
	0.1	245	245	245	245	245	245	245

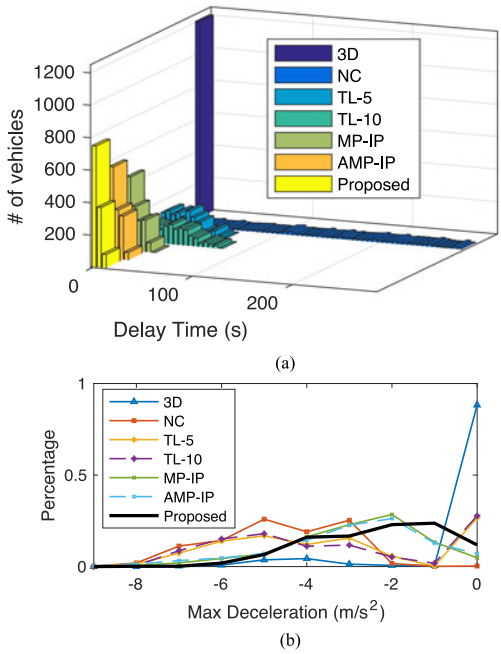

Fig. 9. The statistics of delay time and maximum deceleration under different mechanisms in "straight only" traffic $\lambda = 0.5$ in 10 min. (a) The histogram of delay time for vehicles under different mechanisms. Every bar represents a 5 s interval. (b) The distribution of the max deceleration under different mechanisms.

vehicles with MP-IP or AMP-IP has similar distributions to the proposed case, since the same motion planning algorithm (11) is applied in these two cases.

4) Throughput: The throughput is computed as the number of vehicles that cross the control area in a given time slot. The throughput in 10 min in all scenarios are shown in Table III. When the traffic density is high, Case 2 reduces to the case with stop signs. Hence the throughput is roughly upper bounded by $10 * 60/\delta$ where δ is the average time in seconds that is required for a single vehicle to cross the intersection. In the simulation, $\delta \approx 1$. Hence the throughput in Case 2 is around 600 when $\lambda = 0.5$, which is much smaller than that in other cases. However, in the proposed method, the throughput almost doubles, which is higher than those in Cases 3 to 6 with traffic light or existing V2V intersection protocols, and is very close to that in Case 1 where the intersection is 3D, thus verifies the effectiveness of the proposed method.

VI. CONCLUSION

This paper proposed a novel communication-enabled distributed conflict resolution mechanism for multiple connected autonomous vehicles (CAVs) to navigate at intersections without a traffic manager. Theoretically, it was a parallel algorithm to obtain a feasible solution of a large scale combinatorial optimization in real time. All vehicles close to the intersection were requested to broadcast their estimated times to occupy the conflict zones. The conflict resolution strategy was decoupled temporally. Based on the received information, a vehicle computed a set of vehicles that it needed to yield and the desired time slots to pass the conflict zones in a decision maker. Then it computed a desired speed profile according to the desired time slots in a motion planner. The speed profile was sent to the controller for execution while the estimated time to occupy the conflict zones given the current speed profile was broadcasted again. The aggregation of these local decisions formed a global solution to a multi-vehicle navigation problem. The feasibility and stability of the multi-agent system when all vehicles were using the same strategy were proved theoretically. In the simulation, it was shown that the proposed mechanism outperformed the case when vehicles did not communicate with each other, the cases with traffic lights, and the cases with existing V2V intersection protocols, in that the proposed mechanism generated smaller average delay and larger throughput, which validated the effectiveness of the proposed mechanism.

We conclude that: 1) communication can increase the efficiency of distributed motion planning; 2) conflicts can be solved without a centralized manager; 3) the proposed method outperforms existing distributed intersection protocols due to its flexibility in adjusting the local "yield or go" decisions, hence the passing order in real time.

In the future, the proposed distributed conflict resolution mechanism will be applied to more scenarios with diverse geometric patterns and traffic features. Security issues and synchronization among asynchronous vehicles will be considered. Moreover, we will analyze the safety and stability of the system when vehicles are using different strategies or only a portion of vehicles are connected.

REFERENCES

[1] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *Proc. IEEE World Forum Internet Things*, 2014, pp. 241–246.

[2] C. Liu and M. Tomizuka, "Enabling safe freeway driving for automated vehicles," in *Proc. IEEE 2016 Amer. Control Conf.*, 2016, pp. 3461–3467.

[3] M. Elhenawy, A. A. Elbery, A. A. Hassan, and H. A. Rakha, "An intersection game-theory-based traffic control algorithm in a connected vehicle environment," in *Proc. 2015 IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 343–347.

[4] S. Guberinic, G. Senborn, and B. Lazic, *Optimal Traffic Control: Urban Intersections*. Boca Raton, FL, USA: CRC Press, 2007.

[5] J. Kaths, E. Papapanagiotou, and F. Busch, "Traffic signals in connected vehicle environments: Chances, challenges and examples for future traffic signal control," in *Proc. 2015 IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 125–130.

[6] K. Pandit, D. Ghosal, H. M. Zhang, and C.-N. Chuah, "Adaptive traffic signal control with vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 4, pp. 1459–1471, May 2013.

[7] A. Skabardonis *et al.*, "Advanced traffic signal control algorithms," California PATH, Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. UCB-ITS-PRR-2014-04, 2013.

[8] B. Xu, X. J. Ban, Y. Bian, J. Wang, and K. Li, "V2I based cooperation between traffic signal and approaching automated vehicles," in *Proc. 2017 IEEE Intell. Veh. Symp.*, Jun. 2017, pp. 1658–1664.

[9] K. Dresner and P. Stone, "Multiagent traffic management: A reservation-based intersection control mechanism," in *Proc. 3rd Int. Joint Conf. Auton. Agents Multiagent Syst.-Vol. 2*, 2004, pp. 530–537.

[10] K. Dresner and P. Stone, "Multiagent traffic management: An improved intersection control mechanism," in *Proc. 4th Int. Joint Conf. Auton. Agents Multiagent Syst*, 2005, pp. 471–477.

[11] D. Miculescu and S. Karaman, "Polling-systems-based autonomous vehicle coordination in traffic intersections with no traffic signals," arXiv:1607.07896, 2016.

[12] M. Bashiri and C. H. Fleming, "A platoon-based intersection management system for autonomous vehicles," in *Proc. 2017 IEEE Intell. Veh. Symp.*, Jun. 2017, pp. 667–672.

[13] I. H. Zohdy, R. K. Kamalanathsharma, and H. Rakha, "Intersection management for autonomous vehicles using iCACC," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, 2012, pp. 1109–1114.

[14] N. Murgovski, G. R. de Campos, and J. Sjöberg, "Convex modeling of conflict resolution at traffic intersections," in *Proc. IEEE 54th Annu. Conf. Decision Control*, 2015, pp. 4708–4713.

[15] L. Riegger, M. Carlander, N. Lidander, N. Murgovski, and J. Sjöberg, "Centralized MPC for autonomous intersection crossing," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst.*, 2016, pp. 1372–1377.

[16] M. VanMiddlesworth, K. Dresner, and P. Stone, "Replacing the stop sign: Unmanaged intersection control for autonomous vehicles," in *Proc. 7th Int. Joint Conf. Auton. Agents Multiagent Syst.-Vol. 3*, 2008, pp. 1413–1416.

[17] V. Savic, E. M. Schiller, and M. Papatriantafilou, "Distributed algorithm for collision avoidance at road intersections in the presence of communication failures," in *Proc. 2017 IEEE Intell. Veh. Symp.*, Jun. 2017, pp. 1005–1012.

[18] S. Azimi, G. Bhatia, R. Rajkumar, and P. Mudalige, "Reliable intersection protocols using vehicular networks," in *Proc. ACM/IEEE 4th Int. Conf. Cyber-Physical Syst.*, 2013, pp. 1–10.

[19] C. Liu, J. Chen, T.-D. Nguyen, and M. Tomizuka, "The robustly-safe automated driving system for enhanced active safety," SAE Tech. Paper, Tech. Rep. 2017-01-1406, 2017.

[20] J. Liu, *Autonomous Agents and Multi-Agent Systems: Explorations in Learning, Self-Organization, and Adaptive Computation*. Singapore: World Scientific, 2001.

[21] C. Liu, W. Zhan, and M. Tomizuka, "Speed profile planning in dynamic environments via temporal optimization," in *Proc. Intell. Veh. Symp.*, 2017, pp. 154–159.

Changliu Liu received the B.S. degrees in mechanical engineering and economics from Tsinghua University, Beijing, China, in 2012, and the M.S. degree in mechanical engineering, the M.A. degree in mathematics, and the Ph.D. degree in mechanical engineering from the University of California, Berkeley, Berkeley, CA, USA, in 2014, 2016, and 2017, respectively. Her research interests include robotics, motion planning, and optimization.

Chung-Wei Lin received the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley, Berkeley, CA, USA. He is a Researcher with Toyota InfoTechnology Center, USA, Inc., Mountain View, CA, USA. His research includes modeling, design, analysis of cyber-physical systems, and security and certification of automotive systems.

Shinichi Shiraishi (M'00) received the B.S., M.S., and Ph.D. degrees in electronics engineering from Hokkaido University, Sapporo, Japan, in 1997, 1999, and 2002, respectively. He is currently a Senior Manager with Toyota InfoTechnology Center, USA, Inc., Mountain View, CA, USA. His research interests include software assurance, software architecture, and modeling languages.

Masayoshi Tomizuka (M'86–SM'95–F'97) received the Ph.D. degree in mechanical engineering from Massachusetts Institute of Technology, Cambridge, MA, USA, in 1974. In 1974, he joined the faculty of the Department of Mechanical Engineering, University of California, Berkeley, Berkeley, CA, USA, where he currently holds the Cheryl and John Neerhout, Jr., Distinguished Professorship Chair. His current research interests include optimal and adaptive control, digital control, signal processing, motion control, and control problems related to robotics, precision motion control, and vehicles. He served as the Program Director of the Dynamic Systems and Control Program of the Civil and Mechanical Systems Division of NSF (2002–2004). He served as a Technical Editor of the ASME *Journal of Dynamic Systems, Measurement and Control* (1988–1993), and an Editor-in-Chief of the IEEE/ASME TRANSACTIONS ON MECHATRONICS (1997–1999). He is the recipient of the Charles Russ Richards Memorial Award (ASME, 1997), the Rufus Oldenburger Medal (ASME, 2002), and the John R. Ragazzini Award (2006). He is a Fellow of the ASME and IFAC.