Cooperative Driving at Blind Crossings Using Intervehicle Communication

Li Li, *Member, IEEE*, and Fei-Yue Wang, *Fellow, IEEE*

***Abstract*—Cooperative driving technology with intervehicle communication has attracted increasing attention recently. It aims to improve driving safety and efficiency using appropriate motion scheduling of all the encountered vehicles. Under cooperative driving control, the motion of individual vehicles could be conducted in a safe, deterministic, and smooth manner. This is particularly useful to heavy-duty vehicles since their acceleration/deceleration capacity is relatively low. Specifically in this paper, cooperative driving at blind crossings (crossings without traffic lights) is studied. A concept of safety driving patterns is proposed to represent the collision-free movements of vehicles at crossings. The solution space of all allowable movement schedules is then described by a spanning tree in terms of safety driving patterns; four trajectory planning algorithms are formulated to determine the driving plans with least execution times using schedule trees. The group communication strategy for intervehicle networks is also analyzed. Finally, simulation studies have been conducted, and results demonstrate the potentiality and usefulness of the proposed algorithms for cooperative driving at blind crossings.**

***Index Terms*—*Ad hoc* networks, cooperative driving, driving safety, intervehicle communication, trajectory planning.**

I. Introduction

INCREASING congestion and accidents inspired the concept of automated highway systems (AHSs) more than 20 years ago. A variety of techniques had been introduced to increase the capacity and safety of the existing highway systems since then. Among these techniques, the technology of cooperative driving with intervehicle communication is now considered to be a potential solution to alleviate traffic jams and reduce/prevent collisions.

The concept of cooperative driving was first presented by the Association of Electronic Technology for Automobile Traffic and Driving (JSK) in Japan in the early 1990s [1]. It was originally used as flexible platooning of automated vehicles with a short intervehicle distance over a couple of lanes. At that time, it was known as super smart vehicle systems (SSVSs) [1]–[3]. Using appropriate intervehicle communication to link vehicles, cooperative driving enables vehicles to perform safe and efficient lane changing and merging and, thus, improves the traffic control performance. Since then, the feasibility and benefits of cooperative driving have been further discussed and examined worldwide, i.e., in the California PATH project in the USA [4], [5], the Chauffeur project in the EU [6], and the Demo 2000 Cooperative Driving System in Japan [7].

Generally, all these approaches focus on two questions, namely 1) how to exchange the information among vehicles and 2) how to guide vehicles using the obtained information. The answer to the former question is intervehicle communication [4]–[11]. It enables the vehicles to share information about their driving status and goals, which greatly extend the horizon of drivers or intelligent driving systems. The latter question is answered by using cooperative trajectory planning [4]–[7], [12]–[15].

The conventional methods of intervehicle communication links vehicles through one or more remote service stations (i.e., the communication tower mentioned in [8] and [9]). The driving information of a vehicle will be first transformed to service station and then broadcast to other related vehicles or the vehicles inquiry service station to locate other vehicles. These approaches considerably increase the cost for building and maintaining remote service stations.

Different from the aforementioned methods, many new designs use the peer-to-peer *ad hoc* network to achieve vehicular information sharing. As shown in [10], peer-to-peer *ad hoc* communication networks integrate four valuable features, namely 1) *ad hoc* connectivity, 2) local peer-to-peer networking, 3) short-range communication, and 4) interpersonal communication. However, intervehicle communication is still under further research due to varied driving behaviors and high mobility [11].

From the control systems perspective, cooperative driving can be used to enhance the ride safety and quality. Especially, cooperative trajectory planning is a frequently used control method that designs time-varying velocity profiles for the encountered vehicles. Thus, the motion of individual vehicles can be performed in a safe, smooth, and deterministic manner. This is particularly useful to heavy-duty vehicles, because their acceleration/deceleration capacity is quite low [12]–[15].

In this paper, we use the idea of cooperative driving to study vehicle collision avoidance at road junctions, which is more complex than vehicle platoon control. Intersection collisions represent a significant portion of highway accidents and have gotten much attention recently [16]–[21]. In [16], a simple case of two vehicles had been analyzed using game theory. The

Manuscript received September 22, 2004; revised May 3, 2005 and January 31, 2006. This work was supported in part by the Federal Department of Transportation through the ATLAS Center at the University of Arizona and Grants 60334020, 2002CB312200, 2004GG1104001, and 2006CB705500 from China. The review of this paper was coordinated by Prof. D. Lovell.

L. Li is with the Department of Systems and Industrial Engineering, University of Arizona, Tucson, AZ 85719 USA (e-mail: forcelee@gmail.com).

F.-Y. Wang is with the Key Laboratory for Complex Systems and Intelligence Science, Chinese Academy of Sciences, Beijing 100080, China, and also with the Program for Advanced Research in Complex Systems, Department of Systems and Industrial Engineering, University of Arizona, Tucson, AZ 85719 USA (e-mail: feiyue@sie.arizona.edu).

Digital Object Identifier 10.1109/TVT.2006.878730

0018-9545/$20.00 © 2006 IEEE

differential game approaches assume that all the encountered vehicles cannot know each others' decision [16], [17]. Thus, cooperative driving methods should naturally outperform them, especially when simultaneously dealing with more than two vehicles. Most current approaches had begun to address this benefit, i.e., [18]–[21].

The difficulties of cooperative driving at junctions lie in two aspects. The first one is how to quickly determine whether a particular driving plan is safe or not. Comparing to longitudinal platoon control cases where all vehicles are moving in the same direction, it is much more complex to guarantee the ride safety for crossing at junctions, because vehicles may steer. The second one is to choose the most efficient plan that guides the vehicles to pass the junction by using the least time to alleviate congestions.

To solve both problems, this paper first introduces the concept of safe driving pairs and then extends the virtual vehicle mapping skills into passing scenarios at junctions. The rest of this paper is arranged as follows: In Section II, the basic assumptions for driving at blind crossings and the hierarchical intervehicle communication strategy are described. A simple but practical framework is formulated in Section III to solve cooperative trajectory planning problems, whereas in Section IV, we analyze four possible trajectory planning cases and present the corresponding solution algorithms. In Section V, several simulation results showing the feasibility of the proposed planning method are given. The proposed framework is further discussed in Section VI. Section VII concludes this entire paper.

II. Problem Description

A. Driving Scenarios at Blind Crossings

One basic method toward a collision-free and fast movement of vehicles is to install traffic light systems. Although traffic light systems are not the cure for all, properly using them can promote traffic safety and efficiency. However, because of cost and some other reasons such as short-term operations in mining areas, construction sites, or military zones, many road junctions have no traffic lights. In this paper, we will focus on vehicle driving scenarios at such blind crossings.

Usually, vehicle flows are assumed to arrive continuously at a junction area. At a particular time, we only need to consider a few vehicles that are moving in the vicinity of the crossing. Under this consideration, the continuous traffic flow can be truncated into small segments, which greatly simplifies the problem.

The simple grouping algorithm used here is to label the vehicles by the times that they enter the virtual circle centered at the junction point. As shown in Fig. 1, the four shadow vehicles inside the circle will be considered as a group to take part in cooperative driving; whereas the other three vehicles will not be considered temporarily. The radius of this virtual circle should be determined appropriately by intervehicle communication protocol that has been selected for this application.

Moreover, it is also assumed in this paper that all vehicles have relatively low speeds when approaching the junction, and they will not change lanes after entering the virtual circle since we consider it too close to safely change lanes within the virtual circle. Slowing down gives the vehicles more time to negotiate with each other and prepare for emergence.

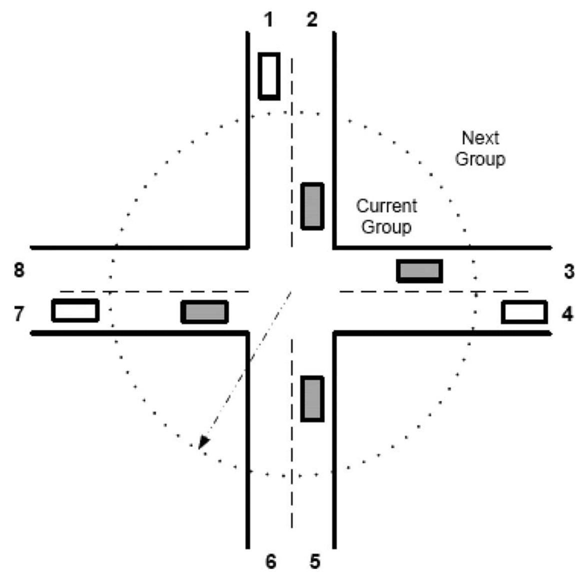


Fig. 1. Vehicle grouping at crossings. The lanes are labeled clockwise.

B. Intervehicle Communication

The intervehicle communication plays an important role in cooperative driving, because the necessary driving information of other vehicles needs to be transmitted to the vehicle that makes the final driving decision.

There were many discussions on designing, implementing and testing of the intervehicle communication, e.g., the Cooperative Optimized Channel Access for INtervehicle communication (COCAIN) proposed by Kaltwasser and Kassubek in [22], the Telecommunication Network for Cooperative Driving (TELCO) proposed by Verdone in [23], and the Dedicated Omnipurpose intervehicle communication Linkage Protocol for HIghway automatioN (DOLPHIN) proposed by Tokuda *et al.* in [24].

As discussed in [25]–[27], in a blind crossing driving scenario, the encountered vehicles need to share the following information with one another:

1) vehicle ID: this ID can be generated by the unique communication hardware equipped on the vehicle;
2) vehicle classification (length and width): as revealed in [28], vehicle size is an important parameter that affects intersection collisions;
3) the current moving lane, the desired moving lane number, and the current speed variation profile of each vehicle before the cooperative driving plan is made;
4) the driving plan of each vehicle after the cooperative driving plan is made;
5) the real-time position (represented as lane number and distance from the crossings) and speed of each vehicle;
6) the emergency signal if needed.

In order to efficiently exchange information, we propose a new model based on the hierarchical subgroup message-deliver models discussed in [24]–[34].

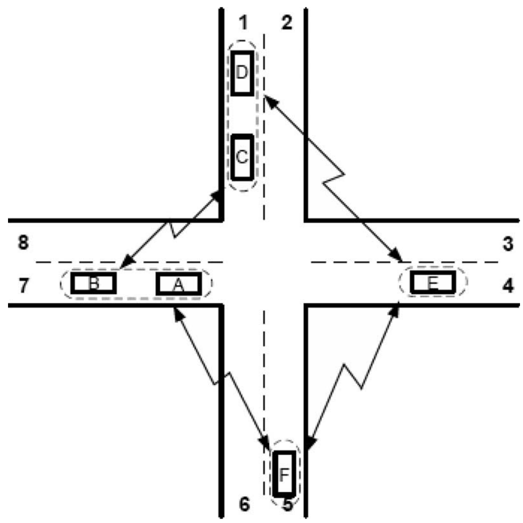


Fig. 2. Vehicle subgrouping and subgroup intervehicle communication.

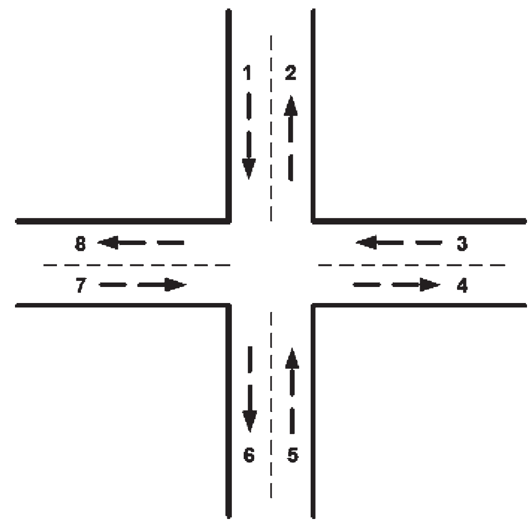


Fig. 3. Diagram of two-lane junctions. The lanes are labeled clockwise.

When approaching the crossing areas, the original vehicle platoon splits into several independent groups. The maximum allowable size of such a group is set as three in this paper. The initial groups are constructed by randomly assigning several temporary dominant vehicles and letting them pick up their neighbor vehicles into its group. The groups that contain more than three vehicles will reject the tail vehicles to meet the limits. The rejected vehicles will try to merge into other groups or formulate a new group.

In this way, the vehicles are supposed to be self-organized into several small groups, as shown in Fig. 2, before they enter the virtual circle.

A vehicle is assumed to store the real-time positions, speeds, and desired driving lane information of all vehicles in the same group. It communicates with other vehicles periodically to update position and speed information. Simulations presented in Section V show that three is an appropriate size for an independent group for blind crossing scenarios.

When two vehicles, for instance, vehicles A and C shown in Fig. 2, in two different groups communicate with each other, vehicle A will transfer the driving information of all vehicles in its same group to C. The received driving information from C will soon be delivered to other vehicles in the group of vehicle A. So does vehicle C to A. Then, a vehicle will store all the information that it received.

It is also assumed that all vehicles will exchange their "temporal" velocity variation profile before the cooperative driving plan is set. Here, "temporal" means the velocity change plan applied before the cooperative driving plan begins. Thus, one vehicle can predict the movements of other vehicles based on received driving information before the cooperative driving plan is set.

Although the data transmission rate is constrained by several factors including media and communication protocol, we assume that the vehicles within the virtual circle can properly and, in a timely manner, acquire the necessary driving information of the other vehicles. Based on the experimental results given in [32]–[34], such assumptions are valid for most blind crossings, because the radius of the virtual circle and the number of encountered vehicles are both limited in those driving scenarios.

The intervehicle communication network will use tags to label the groups/vehicles encountered. If all the vehicles' moving information has been collected by one vehicle, it will proceed to deal with the trajectory planning and simultaneously send messages to block off other vehicles for taking this job.

After the cooperative driving plan is made, the schedule will be properly delivered. Constrained by the communication rate, the driving plan should be represented concisely. A detailed description of the representation method will be presented in Section IV.

When one vehicle leaves the virtual circle, it will leave the current communication group and join a new platoon. Then, emergency signals will be broadcast to all vehicles with the highest priority.

III. Cooperative Driving Schedule

A. Collision-Free Driving Represented by Safe Patterns

In order to guarantee the safety of driving, the concept of safe driving patterns is first introduced.

Generally, the collision may occur when we have the following.

1) Two vehicles move along the same lane, and the lagging one runs into the leading one.
2) Two vehicles moves on different lanes but pass the same junction area simultaneously.

The most frequently used control strategy for blind crossings is the zone blocking strategy. It divides the crossing zone into server subzones, and only one vehicle is allowed to enter a zone at any time to avoid collisions.

Based on a similar idea, a simplified strategy is proposed here, which only allows safe driving patterns (vehicle pairs) to pass the junctions simultaneously. It is similar to the concept of phase of traffic flows in traffic light control.

For instance, consider the two-lane road junction shown in Fig. 3. Suppose one vehicle moves from lane 1 and another vehicle moves from lane 5. Then, there only exist four driving

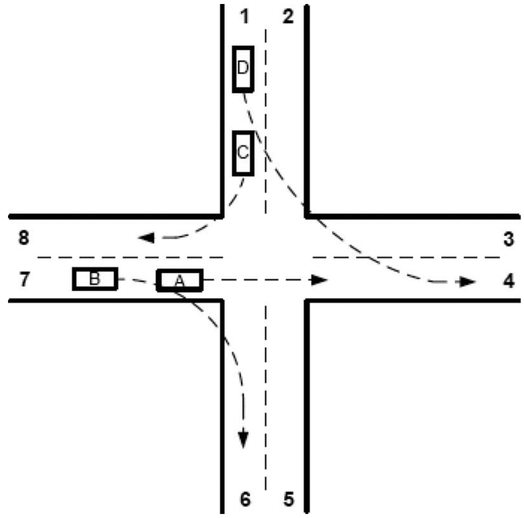


Fig. 4. Four-vehicle driving scenario for a two-lane junction.

patterns that allow two vehicles to cross the junction area safely and simultaneously.

1) One vehicle drives from lane 1 to lane 8, while the other vehicle drives from lane 5 to lane 2.
2) One vehicle drives from lane 1 to lane 6, while the other vehicle drives from lane 5 to lane 2.
3) One vehicle drives from lane 1 to lane 8, while the other vehicle drives from lane 5 to lane 4.
4) One vehicle drives from lane 1 to lane 6, while the other vehicle drives from lane 5 to lane 4.

In terms of safe driving patterns, we can represent a driving schedule as an ordered series of safe patterns. To illustrate this idea, consider a typical driving scenario shown in Fig. 4, in which vehicle A needs to move from lane 7 to lane 4, vehicle B from lane 7 to lane 6, vehicle C from lane 1 to lane 8, and vehicle D from lane 1 to lane 4.

One possible driving schedule for this scenario is to let vehicle A pass the junction first; then, let vehicles B and C pass the junction at the same time; and, finally, let vehicle D pass. Apparently, we can represent this schedule using the following sequence:

$$A_4^7 \left| B_4^7, C_8^1 \right| D_4^1 \tag{1}$$

where the superscript denotes the original lane of a vehicle, and the subscript denotes the destination lane. For simplicity, $A_4^7|B_4^7, C_8^1|D_4^1$ may also be written as $A \mid B,\ C \mid D$ in the rest of this paper if the absence of superscripts and subscripts does not make the driving plan ambiguous.

Here, "|" is a separator symbol that divides the sequence into three subsets, namely 1) A, 2) B and C, and 3) D. Especially, B and C are in one subset, which indicates that they will pass the junctions at the same time.

Generally, consider N vehicles $i_1, i_2, \ldots, i_N$ moving toward the junction. Their order of passing the junction area, i.e., the cooperative driving schedule is specified as

$$i_1, i_5, \ldots, |i_2, i_4, \ldots, |i_3, \ldots, i_N. \tag{2}$$

When one particular driving schedule is determined, the corresponding trajectory planning process for each vehicle will then be carried out. Notice that different driving schedules lead to different passing times. An optimal (suboptimal) cooperative driving plan needs to find the schedule that completes the total driving process with the (nearly) least time. In the rest of this section, we will discuss how to generate all allowable driving schedules, whereas in Section IV, the optimal schedule search process will be described.

B. Solution Tree Generation and Labeling

In general, searching the allowable driving schedule will yield a tree in which each node represents a particular driving plan (sequence) except for the root node. One basic algorithm to generate such a tree is given as follows.

Basic Solution Tree Generation Algorithm

Suppose there are N vehicles under consideration.

1) Generate the root node of the tree.
2) Generate children for the root node, which represent all the possible permutation orders of the vehicle sequence without any separator symbols. It is apparent that all these orders can be enumerated through basic permutation algorithm.
3) For each node in the second level of the tree, generate $N-1$ children by inserting only one separator into the driving sequence that is represented by it since there are only $N-1$ positions that a separator can be inserted.
4) For each node in the third level of the tree, generate $N-2$ children by inserting only one separator into the driving sequence that is represented by it since there are only $N-2$ positions that a separator can be inserted.

$\vdots$

N) For each node in the $N-1$ level of the tree, generate one child by inserting only one separator into the driving sequence that is represented by it since there is only one position that a separator can be inserted.

However, a great number of nodes in this tree can be discarded since they represent invalid driving schedules, for which no trajectory planning is needed.

One apparent fact is that leading vehicles will always pass the junction areas earlier than lagging vehicles in the platoon, because we assume that the lagging vehicle will not change lanes. For instance, in the driving scenario shown in Fig. 4, vehicle A should always pass the crossing earlier than vehicle B. Thus, the node $B\ A\ C\ D$ and all its children should be invalid.

Based on this fact, we can modify the aforementioned algorithm as follows.

Modified Solution Tree Generation Algorithm

Suppose there are N vehicles under consideration.

1) Generate the root node of the tree.
2) Generate children for the root node, which represent all the possible permutation orders of the vehicle sequence without considering separator symbols. Then, prune all the obtained nodes that represent invalid order of driving.
3) Same as the Basic Solution Tree Generation Algorithm.

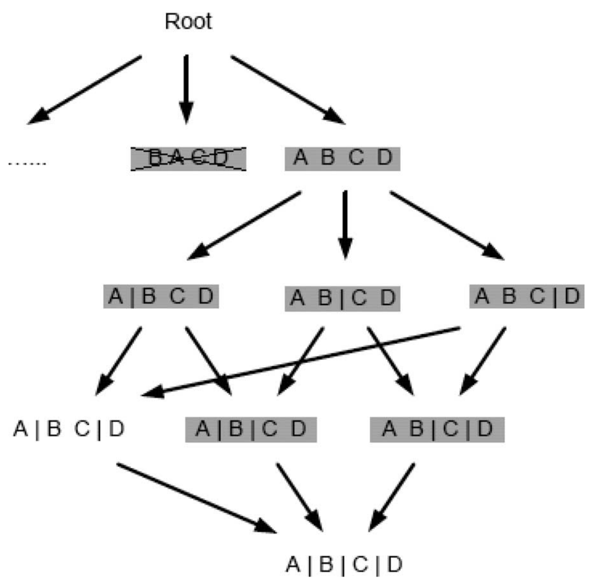


Fig. 5. Schedule tree stemmed from the driving scenario shown in Fig. 4. The shadow nodes represent invalid driving schedule.

To guarantee a collision-free movement, the safety of the simultaneously moving subsets in a driving sequence has to be checked. It is obvious that each subset in a valid driving schedule should constitute of a safe driving pattern.

Note that M vehicles can safely cross the junctions at the same time if and only if every pair of these vehicles is a safety pattern. Therefore, a labeling algorithm is given as follows.

Safety Pair Labeling Algorithm

If there are M $(M \geq 2)$ vehicles $j_1, j_2, \ldots, j_M$ that pass the crossings simultaneously, check the $M(M-1)/2$ distinct vehicle pairs (j_1, j_2), $(j_1, j_3), \ldots,$ (j_{M-1}, j_M) to see whether they are all safety patterns. If not, label the corresponding node as an unsafe node.

Using these two considerations, most unsafe driving schedules can be directly pruned from the solution tree. For example, the driving plan tree that stemmed from the scenarios in Fig. 4 only has a few valid nodes after pruning and labeling, i.e., there are only two valid driving schedules for the ordered sequence A, B, C, D (see Fig. 5). The first schedule is specified by (1), and the second is to simply let vehicles A, B, C, and D pass the junction area sequentially.

After labeling, trajectory planning will be executed for every valid node with respect to its driving schedule. Some driving plans will be discarded since they are implicitly forbidden by the vehicle dynamics constraints.

IV. Cooperative Trajectory Planning

A. Trajectory Planning

In general, a vehicle's trajectory crossing the junctions can be divided into three sequential stages.

1) Approach the junction. The vehicle should avoid collision between the leading vehicle and itself. Moreover, it should also avoid collision between itself and the last vehicle that had passed the junction before it but does not form a safety pair with it.
2) Cross the junction, which is considered as a decelerate–accelerate process.
3) Leave the junction. It should avoid collision between the new leading vehicle and itself.

In order to solve the lane-merging trajectory planning problems, the virtual vehicle mapping technique was proposed and employed by Uno *et al.* in [35] and [36]. The concept of a virtual vehicle is to map a vehicle on a lane onto an object lane; then, the interested vehicle can be controlled with respect to the virtual vehicle to guarantee safety.

Considering the risk of failure, an algorithm similar to what is proposed in [35] and [36] is applied here. The only difference is that the leading vehicle is not mapped onto the symmetry position of the desired lane. Actually, it will be mapped into a position that lags off the mirror point to compensate the communication delay and vehicle deceleration at steering.

For example, as shown in Fig. 6, vehicle A moves from lane 7 to lane 6 first; then, vehicle B moves from lane 1 to lane 6. Since vehicle B passes the junction area right after vehicle A, it must make enough headway to avoid collision. Therefore, it generates a virtual vehicle A' by mapping vehicle A into its own lane using the data transmitted via the intervehicle communication. Classical longitudinal control will then be performed between virtual vehicle A' and vehicle B.

The lengths of the two headways should be determined by the dynamic properties and the length/size of encountered vehicles as well as the applied intercommunication protocols.

The trajectory profile of each vehicle in the driving schedule will be generated one by one with respect to the corresponding driving order passing the junction. Every vehicle can "know" the trajectories of the other related vehicles that need to map onto its lane.

Since we need to improve traffic efficiency, the driving plan should keep the headway between the potential leading/virtual vehicle and itself to the minimum safe distance.

Generally, we can classify various driving scenarios into four cases.

Case A) There is neither a leading vehicle moving in the same lane nor a virtual vehicle that needs to map before the planning vehicle enters the junction area.

Case B) There is a leading vehicle moving in the same lane but no virtual vehicles that need to be mapped before the planning vehicle enters the junction areas. Indeed, it equivalently means that the planning vehicle will pass the junction area right after the leading vehicle.

Case C) There is not a leading vehicle moving in the same lane but a virtual vehicle that needs to be mapped before the planning vehicle enters the junction area.

Case D) There is a leading vehicle moving in the same lane, and there is also a virtual vehicle that needs to be mapped before the planning vehicle enters the junction area. Actually, it means that the planning vehicle will first follow the leading vehicle and then follow the virtual vehicle before it enters the junction.

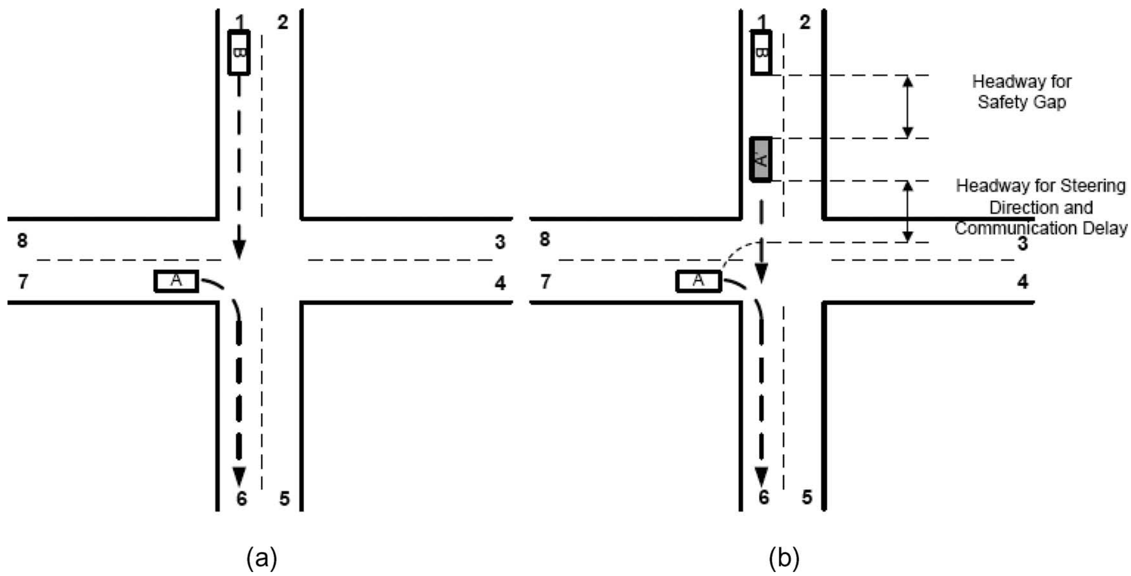


Fig. 6. Trajectory generation considering one virtual vehicle. (a) With actual vehicles only. (b) With virtual vehicle A' introduced.

Suppose the position of the planning vehicle is denoted by x_p, and the accelerate/decelerate rate of the vehicle is denoted by $\ddot{x}_p$. Their dynamics are constrained by

$$\ddot{x}_{p\,\min} \le \ddot{x}_p \le \ddot{x}_{p\,\max} \tag{3}$$

$$0 \le \dot{x}_p \le v_{\max} \tag{4}$$

where $\ddot{x}_{p\,\min}$ and $\ddot{x}_{p\,\max}$ are the accelerate/decelerate rate bounds, and $v_{\max}$ is the maximum allowable safe speed.

If the planning vehicle needs to steer directions, there is an additional constraint, which is stated as follows:

$$\dot{x}_p(t_3) \le v_{s_{\max}} \tag{5}$$

where $v_{s_{\max}}$ is the maximum allowable speed for steering, and t_3 is the time when the planning vehicle enters the crossings.

The corresponding trajectory planning algorithms can be formulated as follows.

1) Algorithm I. Trajectory Planning for Case A

I.1) Solve the time-optimal trajectory planning problem before the planning vehicle enters the junction areas.

Obviously, the objective is

$$\min_u t_3 \tag{6}$$

which is constrained by the following boundary conditions:

$$x_p(t_0) = x_{p0} \quad x_p(t_3) = x_{p3} \tag{7}$$

where x_{p0} is the start position, x_{p3} is the final position at the boundary of junction areas, and t_0 is the time when planning starts. u denotes the controllable velocity change profile from here to the end of this paper.

I.2) There is no planning at this stage. We simply assume that the time consumed in crossing junction areas only depends on the speed of the vehicle when it enters the junction areas.

The time when the planning vehicle leaves the junction areas should be

$$t_4 = t_s + f_1\left(\dot{x}_p(t_3)\right) \tag{8}$$

and the speed should be

$$\dot{x}_p(t_4) = f_2\left(\dot{x}_p(t_3)\right) \tag{9}$$

where t_4 is the time when the planning vehicle leaves the junction areas.

I.3) Solve the tracking problem for the new leading vehicle before the virtual vehicle leaves the junction area.

Suppose the trajectory of the new leading vehicle is given as

$$\ddot{x}_n = f_3(t) \tag{10}$$

where x_n is the position of the new leading vehicle, and $f_3(t)$ is its movement-descriptive function in terms of time.

Thus, the error e_{np} in headway from the preselected safe distance L_3 can be written as

$$e_{np} = x_n - x_p - L_3. \tag{11}$$

Obviously, the objective is

$$\min_u \int_{t=t_4}^{t_4+T} |e_{np}|dt \tag{12}$$

which is constrained by (6), (7), and the boundary condition in (10). Here, T is a predetermined time span that is long enough to appropriately describe the planning vehicle's movement after it leaves the junction areas.

2) Algorithm II. Trajectory Planning for Case B

II.1) Solve the leading vehicle tracking problem before the leading vehicle leaves the junction area.

Suppose the trajectory of the leading vehicle is given by

$$\ddot{x}_l = f_4(t) \tag{13}$$

where x_l is the position of the leading vehicle, and $f_4(t)$ is its movement-descriptive function in terms of time.

Thus, the error e_{lp} in headway from the preselected safe distance L_1 can be written as

$$e_{lp} = x_l - x_p - L_1. \tag{14}$$

Obviously, the objective is

$$\min_u \int_{t=t_0}^{t_1} |e_{lp}| dt \tag{15}$$

which is constrained by the initial condition

$$x_p(t_0) = x_{p0} \tag{16}$$

(6), (7), and the boundary condition in (10), where t_1 is the time when the leading vehicle enters the junction areas.

II.2) Same as Algorithm I.1, except the initial condition is given by

$$x_p(t_1) = x_{p1} \tag{17}$$

where x_{p1} is the planning vehicle's position at time t_1.

II.3) Same as Algorithm I.2.

II.4) Same as Algorithm I.3.

3) Algorithm III. Trajectory Planning for Case C

III.1) Solve the virtual vehicle tracking problem before the virtual vehicle leaves the junction area.

Assume the trajectory of the virtual vehicle to be

$$\ddot{x}_v = f_5(t) \tag{18}$$

where x_v is the position of the virtual vehicle, and $f_5(t)$ is its movement-descriptive function in terms of time.

Thus, the error e_{vp} in headway from the preselected safe distance L_2 can be written as

$$e_{vp} = x_v - x_p - L_2. \tag{19}$$

Obviously, the objective is

$$\min_u \int_{t=t_0}^{t_2} |e_{vp}| dt \tag{20}$$

which is constrained by the given initial condition (17) and the dynamic constraints (6) and (7), where t_2 is the time when the virtual vehicle leaves the junction areas.

III.2) Same as Algorithm I.1 except the initial condition is given as

$$x_p(t_2) = x_{p2} \tag{21}$$

where x_{p2} is the planning vehicle's position at time t_2.

III.3) Same as Algorithm I.2.

III.4) Same as Algorithm I.3.

4) Algorithm IV. Trajectory Planning for Case D

IV.1) Same as Algorithm II.1.

IV.2) Same as Algorithm III.1 except the initial condition is given as in (18), and t_1 is the time when the leading vehicle enters the junction areas.

IV.3) Same as Algorithm I.1 except the initial condition is given as in (22), and t_2 is the time when the virtual vehicle leaves the junction areas.

IV.4) Same as Algorithm I.2.

IV.5) Same as Algorithm I.3.

It should be pointed out that the choice of vehicle dynamic models and/or longitudinal driving controllers will not vary the feasibility of the proposed planning framework. Furthermore, some other driving performance index such as ride comfort can be formulated and added here. Some related discussion on objective choice of vehicle trajectory planning can be found in [37] and [38].

However, there is still one important issue that needs to be clarified. Limited by the communication rate, the cooperative driving plan should not be too complicated. Otherwise, the cooperative driving plan cannot be correctly delivered to all the encountered vehicles.

Here, we further constrain the trajectory of one vehicle that constitutes at most five steady acceleration/deceleration processes. Thus, one vehicle's trajectory will be determined by at most six data sets given as follows:

$$\langle \text{start time } t_0, \text{velocity } \dot{x}_p(t_0) \rangle, \langle \text{velocity change time } t_1 \text{ velocity } \dot{x}_p(t_1) \rangle, \ldots, \langle \text{end time } t_5, \text{velocity } \dot{x}_p(t_5) \rangle. \tag{22}$$

The velocity between time t_i and t_{i+1} can be obtained as

$$\dot{x}_p(t) = \dot{x}_p(t_i) + [\dot{x}_p(t_{i+1}) - \dot{x}_p(t_i)] \cdot \frac{t - t_i}{t_{i+1} - t_i}. \tag{23}$$

This method greatly reduces the computation costs of the aforementioned trajectory planning problems without losing too much generality. Moreover, this technique also relieves the burden of the intervehicle communication network since there are only 12 variables in (23) that need to be encoded and delivered to the involved vehicle. The vehicle will resolve the control inputs from this simplified velocity profile based on its own dynamic equation.

B. Best Driving Plan Search

The total time cost of trajectory is counted from the time when the cooperative driving process begins to the time when the last vehicle leaves the junction area. All the undiscarded

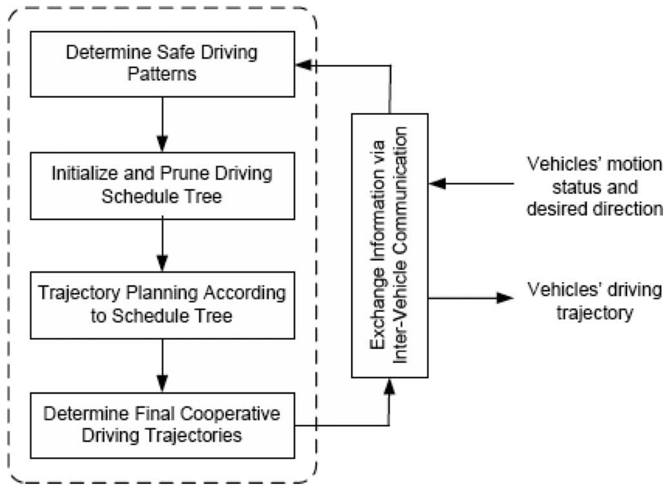


Fig. 7. Cooperative driving planning framework.

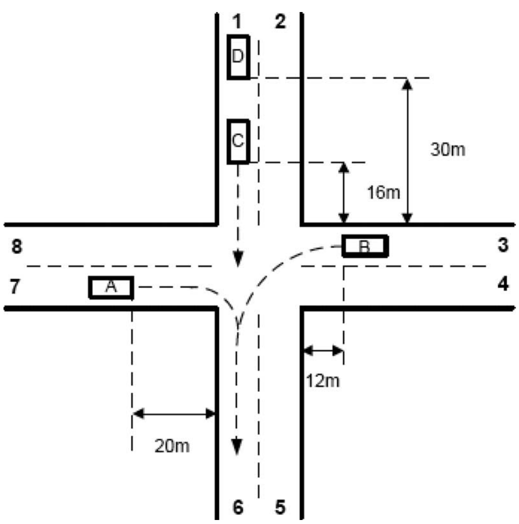


Fig. 8. Four-vehicle driving scenario for a two-lane junction.

driving plans will be compared, and the one with the least time cost will be chosen as the actual driving plan.

The diagram of this cooperative driving planning framework is shown in Fig. 7. To summarize, the safety of the framework is guaranteed by selecting safe driving patterns, and the efficiency is achieved by adopting the time-optimal driving plan.

In the searching process of optimal driving plan, the upper level nodes in the schedule tree will be examined earlier with respect to the lower level plans. If one plan is proven to be valid, then, all its children nodes will be omitted, because a valid node always represents the plan that uses less time than that of its children plans. For example, if solution $A \mid B\, C \mid D$ in Fig. 5 is a valid plan, then its child node $A \mid B \mid C \mid D$ needs no analysis.

V. Simulation Results

A. Simulation Results for Trajectory Planning

A demonstration simulation case shown in Fig. 8 is studied here. In this scenario, vehicle A moves from lane 7 to lane 6, vehicle B from lane 3 to lane 6, vehicles C and D from lane 1 to lane 6.

The speeds of the four vehicles are all 2 m/s at the beginning time. The current distances between vehicles A, B, C, and D and the junction areas is 20, 12, 16, and 30 m, respectively. The width of each lane is 2 m.

Suppose the four vehicles have identical dynamic properties. The maximum navigation speeds for them are all 2 m/s before they enter the junction and 5 m/s after they leave. The maximum steering speeds are all 1 m/s.

The maximum control input is constrained by

$$-0.5 \le \ddot{x}_p \le 0.5 \text{ m/s}^2. \tag{24}$$

The length of each vehicle is assumed to be 4 m. The safety headway L_1 defined in (14) is chosen as 4 m, whereas the safety headway L_2 defined in (19) is 7 m.

Apparently, there exist several allowable driving plans. Let us calculate the trajectory planning process for the driving plan $B \mid C \mid D \mid A$ as an example. It lets vehicles B, C, D, and A pass the junction sequentially. The trajectory planning will be carried out for vehicles B, C, D, and A sequentially.

1) The trajectory planning that vehicle B should adopt is Algorithm I.

   First, it will carry out a time-optimal trajectory until it enters the junction areas. Obviously, in the first 2 s, vehicle B accelerates to 4 m/s with $\ddot{x}_p = 0.5$ m/s^2. Then, it slows down with $\ddot{x}_p = -0.5$ m/s^2 in the next 2 s.

   Suppose that vehicle B passes the junction areas within 1.5 s, and its speed becomes 1 m/s when it leaves the junction areas. Then, it accelerates to 5 m/s with $\ddot{x}_p = 0.5$ m/s^2 in the next 8 s. It will keep this speed in the rest of the planning. Thus, its speed profile should be the one shown in Fig. 9(a).

2) Vehicle C will pass the junction areas right after vehicle B. Apparently, it should use planning Algorithm III.

   Here, we divide its trajectory into three parts.

   2.1) First, it carries out a decelerate process to enlarge the headway between the virtual vehicle B and itself.
   2.2) Then, it passes the junction areas after vehicle B.
   2.3) Finally, it tracks the new head vehicle B.

   The initial distance between vehicle C and virtual vehicle B is -2 m. Mapping vehicle B into the lane of vehicle C, we can have the virtual vehicle's speed trajectory, as shown in Fig. 9(b).

   To accurately solve the minimum tracking error problem (21) is difficult and unnecessary. Here, we adopt the following driving trajectory for simplicity:

   2.1) First, it keeps its speed at 2 m/s until $t = 4.5$ s.
   2.2) Then, it decelerates to 1 m/s with $\ddot{x}_p = -0.5$ m/s^2 and accelerates continuously with $\ddot{x}_p = 0.5$ m/s^2 from $t = 6.5$ s to $t = 14.5$ s.
   2.3) Finally, it keeps constant speed at 5 m/s in the rest of our planning. Thus, the whole speed profile for vehicle C can be illustrated as shown in Fig. 9(c).

3) The trajectory of vehicle D should follow Algorithm II.

   It can be depicted as follows:

   3.1) It keeps its speed at 2 m/s from $t = 0$ s to $t = 6$ s.

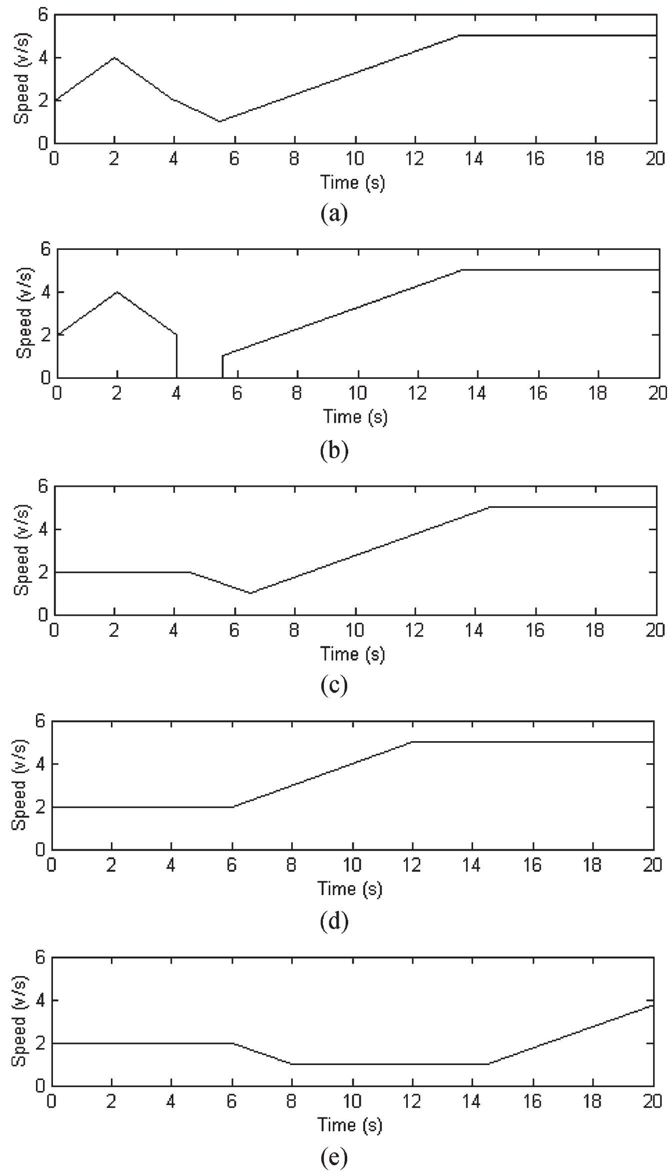


Fig. 9. Speed profiles for (a) vehicle B, (b) virtual vehicle B, (c) vehicle C, (d) vehicle D, and (e) vehicle A.

3.2) After crossing the junction areas, it accelerates to 5 m/s with $\ddot{x}_p = 0.5$ m/s^2 in the next 6 s and keeps this speed in the rest of our planning. Fig. 9(d) shows the speed profile for it.

4) Finally, let us apply Algorithm III for vehicle A.

Similarly, we can get a simple speed profile as follows [the entire process is shown in Fig. 9(e)]:

4.1) Keep speed at 2 m/s from $t = 0$ s to $t = 6$ s, and then, decelerate to 1 m/s from $t = 6$ s to $t = 8$ s.

4.3) Reach crossing at $t = 13$ s with a speed of 1 m/s, and cross the junction in 1.5 s with the speed unchanged.

4.4) Accelerate to 5 m/s with $\ddot{x}_p = 0.5$ m/s^2 in the next 8 s, and keep this speed in the rest of our planning.

Thus, the total time consumed for driving plan $B \mid C \mid D \mid A$ is 14.5 s. The time costs for other driving plans are also listed in Table I. It is clear that the optimal driving plan significantly save time for the entire traffic flow. In this case, the final optimal driving schedule is $B \mid C \mid D \mid A$.

TABLE I
COMPARISON OF DIFFERENT DRIVING PLANS

Driving Plan	Time Consumed
B* | *C* | *D* | *A	14.5s
B* | *A* | *D* | *C	18.5s
A* | *B* | *C* | *D	17.5s
A* | *C* | *D* | *B	17.0s
C* | *D* | *A* | *B	19.5s
……	……

B. Simulation Results for Intervehicle Communication

To test our subgroup managing strategy for communication in multicar at blind crossings, a simple time-division event-driven multiagents communication model is set and simulated.

There are four parameters in this model, which are listed as follows:

1) communication group size;
2) total number of vehicles;
3) contacting rate: It is a probability that is used to describe how easy one "free" vehicle can get connected with another "free" vehicle. Here, "free" means that vehicles are not engaged in a conversation. The higher the contacting rate, the easier one free vehicle sets up a conversation with another free vehicle. This parameter is determined by the media, protocol, and burden of the intervehicle communication networks. Apparently, a high burden will lead to a low contacting rate.
4) forbidding time length: It is a number to describe how long one free vehicle should wait before it communicates again with the same vehicle that it has "talked" to before. Notice that one vehicle prefers to receive new information rather than obtain old information again and again from the same vehicle.

It is assumed that the data information will be transferred at constant speed. When one free vehicle carries out conversation with another free vehicle, they will become occupied and do not response to other requests.

In our simulation, for simplicity, it is assumed that one certain vehicle A needs 0.1–0.2 s to send the driving information of one vehicle (maybe this information belongs to vehicle A; maybe this information belongs to another vehicle, for example, vehicle C, but currently stored in vehicle A's information cache) to another vehicle B, and vehicle A is smart to avoid sending vehicle B's driving information to vehicle B.

By choosing different values of these parameters, we compare the effect of communication group size and forbidding time length. Fig. 10 shows the average communication time with respect to different group sizes while the forbidding time length is chosen to be 1. Obviously, three is an appropriate size for an independent group in this case. Similar conclusions were reached for different forbidding time lengths.

Fig. 11 shows the average communication time with respect to different forbidding time lengths, while the group size is chosen as 3. It reveals that the forbidding time length should be mediate. However, there is not a simple rule to choose an appropriate forbidding time length for different group sizes and the total number of vehicles.

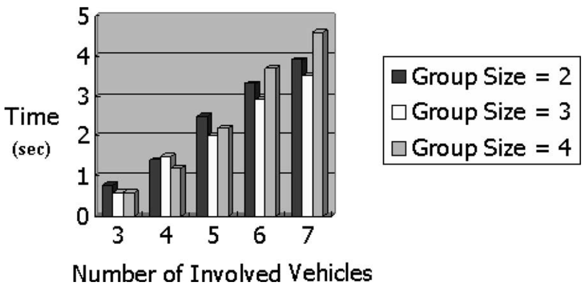


Fig. 10. Average communication time with respect to different group sizes (forbidding time length $= 1$).

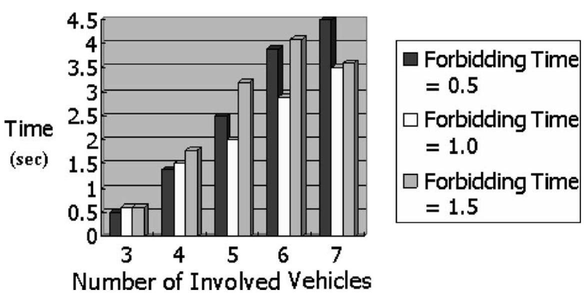


Fig. 11. Average communication time with respect to different forbidding time lengths (group size $= 3$).

VI. Discussion

Although our algorithms work well in many scenarios of driving at junctions in simulation, there are several problems that need to be further addressed.

1) The complexity of this cooperative driving planning will increase quickly with the number of vehicles. How to generate the solution space more efficiently has to be further analyzed before the proposed algorithm is applied to real applications.

2) Although the illustrations above focus on two-lane crossings, the proposed algorithms can be employed for multiple-lane crossings with slight modifications. The safety pair labeling algorithm can still guarantee the feasibility of the driving pattern, whereas the driving schedule generation procedure will become tedious.

More specifically, consider the driving scenario shown in Fig. 4(a). Each vehicle may have two possible destination lanes. This simply leads to a great number of nodes in the first level of the corresponding driving schedule tree. Then, the complete driving schedule express method needs to be employed to distinguish different driving schedules. For instance, the driving schedule generation tree of Fig. 4(a) may contain nodes (A_7^{14}, B_{15}^6, C_3^5), (A_8^{14}, B_{15}^6, C_3^5), etc. in the first level.

Besides, if the vehicle is allowed to change lanes after having entered the virtual circle, the related trajectory planning problem has to be further examined too. With lane changing, the collision around the crossings can be approximately divided into the following three types:

a) rear crash, which the lagging vehicle runs into the lead one. For instance, vehicle D collides with vehicle C as shown in Fig. 5(b).

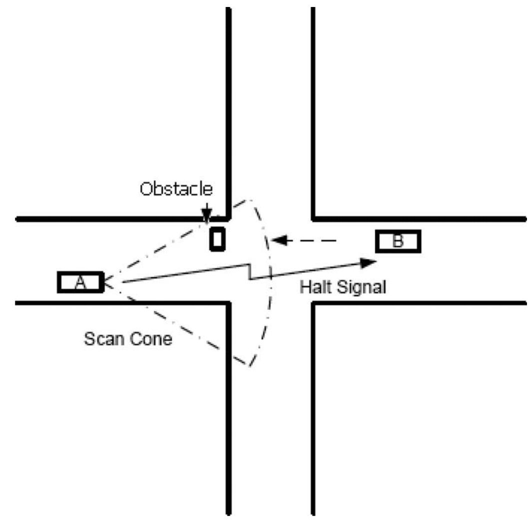


Fig. 12. Handling an emergency case. The obstacle might be a suddenly appeared pedestrian.

b) side crash, which normally happens during left turn. For instance, vehicle A collides with vehicle B as shown in Fig. 5(b).

c) angle crash, which can be avoided by appropriately steering control.

The first two problems can be formulated as lane changing path planning problems and then solved by extending the techniques in [5] together with the modified virtual vehicle mapping algorithms. Detailed discussion on this problem can be found in [39]. Constrained by the paper length, the authors will further explain the related issues in that paper.

3) The actual driving scenario at junctions might be more complex than what we have discussed here. One important case is how to deal with emergencies. One potential answer is to share each vehicle's sensor information through the intervehicle communication as what has been discussed in [5], [25]–[28], and [39]–[42].

If an emergence happens, the first "knowing" vehicle will soon stop all the related vehicles synchronically. For example, as shown in Fig. 12, an emergent halt signal sent out from vehicle A quickly stops vehicle B that intends to move to the blocked lane, when it suddenly "sees" an obstacle on that lane. The cooperative driving plan will be regenerated when the detected obstacles moved away.

4) Here, the speeds and positions of all the vehicles are presumably measured accurately. However, this assumption may not hold in some situations. Thus, the headway reserved for communication delay and steering direction should be intentionally enlarged to counterbalance the measurement error and the driver uncertainty. Besides, if the sensors on a vehicle fail and send out wrong data, it may still cause severe accidents. The framework discussed above does not provide a mechanism to detect such faults yet. One possible way to solve this problem is to introduce a road (intersection) monitor system to check the potential faults via vehicle-to-road communication, e.g., [19]–[21].

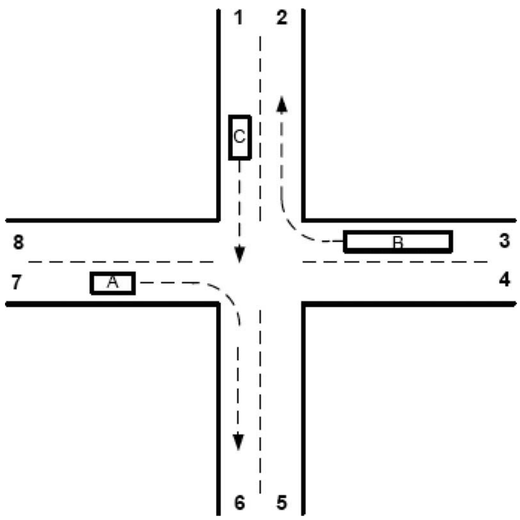


Fig. 13. Driving scenario for two-lane crossings.

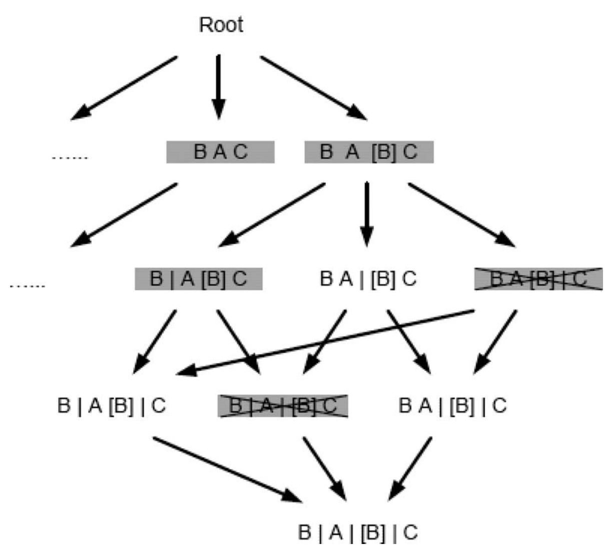


Fig. 14. Corresponding driving schedule tree for the driving scenario shown in Fig. 13. The shadow nodes represent invalid driving schedule.

5) It should be pointed out that if the calculated tactics or strategies for vehicle approaches are not enforced through automated control, the maneuvers can only be safe if the drivers follow all instructions. Although nobody can be 100% sure to guarantee that drivers follow directions all the time, it is widely believed that proper guidance would help a driver to reduce the probability of making wrong decisions/actions. To further enlarge the reserved headway can help to reduce collision when drivers make inappropriate actions. However, this contradicts the driving efficiency requirement. How to balance these two problems still needs further discussions.

6) The geometry constraints of vehicles and junction areas have not been well discussed in this paper. However, they should not be neglected in real conditions.

For instance, if a van truck steers from lane 1 to lane 8 as depicted in Fig. 13, it may block off lane 2 at the same time. This problem can be solved by judging whether two vehicles form a safe driving pattern by considering both their moving directions and geometry characteristics.

Another problem is that the longer a vehicle, the more time it will take to pass the intersection. For some especially long vehicles, the aforementioned driving schedule generation algorithm may not do, because some possible driving sequences cannot be simply represented as permutation orders of the vehicles. In other words, two or more than two other short vehicles may pass the intersection sequentially, while the long vehicle is still in its steering process.

A simple strategy to deal with such cases is to generate more nodes in the first level of the driving schedule tree by inserting the so-called mirror symbols. Suppose at most two other short vehicles pass the intersection sequentially before a long vehicle finishes its steering process. Thus, if node

$$\cdots i_{\text{long}}\ i_{\text{particular}} \cdots \tag{25}$$

is in the first level, where i_{long} denotes the long vehicle, then, we need to add some nodes formulated as

$$\cdots i_{\text{long}}\ i_{\text{particular}}\ \cdots\ [i_{\text{long}}] \cdots \tag{26}$$

into the first level.

Let us use an example to explain it. Consider a three-vehicle driving scenario for a two-lane junction shown in Fig. 13. Since vehicle B is too long, we can add a node

$$B, A|[B], C \tag{27}$$

where mirror symbol $[B]$ indicates that after vehicle A passes the intersection, vehicle B is still in the crossing areas.

The following is obvious.

a) For a valid driving schedule node, i_{long} and $[i_{\text{long}}]$ cannot be in the same subset. Thus, B, A, $[B]|C$ does not indicate a valid driving schedule.
b) The subset containing i_{long} and the subset containing $[i_{\text{long}}]$ should be neighbors. Thus, $B \mid A \mid [B], C$ does not indicate a valid driving schedule either.
c) Using these two rules, most invalid nodes can be directly pruned from the schedule tree (see Fig. 14).

7) In many situations, i.e., in most occidental cities, blind crossings are ruled on the basis of priorities (e.g., a vehicle gives right of way to vehicles proceeding at the right of the vehicle). If the first-arrive-first-pass priority rule is employed, it will be much easier to generate the driving schedule.

The corresponding process can be expressed as follows.

a) Order all the encountered vehicles by their estimated time of arriving at the intersection and the right way rules.
b) Based on this order, generate the nodes in the first level of the driving schedule tree. Notice that each vehicle may choose different destination lanes when running

in multiple-lane crossings; there are more than one node in this level.

c) Then, by gradually inserting separator symbols into this sequence, a special driving schedule tree can then be obtained. The tree generation algorithm is exactly the same as what is presented in this paper.

VII. Conclusion

This paper extends the idea of cooperative driving platoon to collision-free driving at blind crossings. A framework for generating time-optimal collision-free cooperative plans is developed. Simulation results indicate that the proposed algorithms have the potential for actual applications in driving at blind crossings, especially for construction sites and military excises where heavy-duty vehicles are often involved.

Acknowledgment

The authors would like to thank the anonymous reviewers for their insightful comments, which provided valuable guidance on how to improve this paper.

References

[1] S. Tsugawa, "Inter-vehicle communications and their applications to intelligent vehicles: An overview," in *Proc. IEEE Intell. Veh. Symp.*, 2002, vol. 2, pp. 564–569.

[2] S. Tsugawa, T. Saito, and A. Hosaka, "Super smart vehicle system: AVCS related systems for the future," in *Proc. Intell. Veh. Symp.*, 1992, pp. 132–137.

[3] M. Nakamura, H. Kawashima, T. Yoshikai *et al.*, "Road-vehicle cooperation driving system," in *Proc. Veh. Navigat. and Inf. Syst. Conf.*, 1994, pp. 425–430.

[4] J. K. Hedrick, M. Tomizuka, and P. Varaiya, "Control issues in automated highway systems," *IEEE Control Syst. Mag.*, vol. 14, no. 6, pp. 21–32, Dec. 1994.

[5] A. R. Girard, J. B. de Sousa, J. A. Misener *et al.*, "A control architecture for integrated cooperative cruise control and collision warning systems," in *Proc. IEEE Conf. Decision and Control*, 2001, vol. 2, pp. 1491–1496.

[6] O. Gehring and H. Fritz, "Practical results of a longitudinal control concept for truck platooning with vehicle to vehicle communications," in *Proc. IEEE Conf. Intell. Transp. Syst.*, 1997, pp. 117–122.

[7] S. Kato, S. Tsugawa, K. Tokuda *et al.*, "Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 3, pp. 155–161, Sep. 2002.

[8] A. M. K. Cheng and K. Rajan, "A digital map/GPS based routing and addressing scheme for wireless *ad-hoc* networks," in *Proc. IEEE Intell. Veh. Symp.*, 2003, pp. 17–20.

[9] J.-P. Hubaux, S. Capkun, and J. Luo, "The security and privacy of smart vehicles," *IEEE Security Privacy*, vol. 2, no. 3, pp. 49–55, May–Jun. 2004.

[10] L. Chisalita and N. Shahmehri, "A peer-to-peer approach to vehicular communication for the support of traffic safety applications," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2002, pp. 336–341.

[11] J. Blum, A. Eskandarian, and L. J. Hoffman, "Performance characteristics of inter-vehicle *ad hoc* networks," in *Proc. IEEE Conf. Intell. Transp. Syst.*, 2003, vol. 1, pp. 114–119.

[12] Y. Chang and K. Kurami, "Longitudinal guidance and control for the entry of vehicles onto automated highways," in *Proc. IEEE Conf. Decision and Control*, 1993, vol. 2, pp. 1891–1896.

[13] P. Kachroo and Z. Li, "Vehicle merging control design for an automated highway system," in *Proc. IEEE Conf. Intell. Transp. Syst.*, 1997, pp. 224–229.

[14] J. W. Choi, T. H. Fang, S. Kwong *et al.*, "Remote-controlled platoon merging via coder-estimator sequence algorithm for a communication network," *IEEE Trans. Ind. Electron.*, vol. 50, no. 1, pp. 30–36, Feb. 2003.

[15] S. Joo, X. Y. Lu, and J. K. Hedrick, "Longitudinal maneuver design in coordination layer for automated highway system," in *Proc. Amer. Control Conf.*, 2003, vol. 1, pp. 42–47.

[16] S. Arora, A. K. Raina, and A. K. Mittal, "Collision avoidance among AGVs at junctions," in *Proc. IEEE Intell. Veh. Symp.*, 2000, pp. 585–589.

[17] R. Lachner, "Collision avoidance as a differential game: Real-time approximation of optimal strategies using higher derivatives of the value function," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.—'Comput. Cybern. and Simul.'*, 1997, vol. 3, pp. 2308–2313.

[18] L. Li, J. Song, F.-Y. Wang *et al.*, "New developments and research trends for intelligent vehicles," *IEEE Intell. Syst.*, vol. 20, no. 4, pp. 10–14, Jul./Aug. 2005.

[19] A. Avila, G. Korkmaz, Y. Liu *et al.*, "A complete simulator architecture for inter-vehicle communication based intersection warning systems," in *Proc. IEEE Conf. Intell. Transp. Syst.*, 2005, pp. 461–466.

[20] H. C. Fuerstenberg, "A new European approach for intersection safety—The EC-project INTERSAFE," in *Proc. IEEE Conf. Intell. Transp. Syst.*, 2005, pp. 343–347.

[21] A. Benmimoun, J. Chen, D. Neunzig *et al.*, "Communication-based intersection assistance," in *Proc. IEEE Intell. Veh. Symp.*, 2005, pp. 308–312.

[22] J. Kaltwasser and J. Kassubek, "A new cooperative optimized channel access for inter-vehicle communication," in *Proc. Veh. Navigat. and Inf. Syst. Conf.*, 1994, pp. 145–148.

[23] R. Verdone, "Communication systems at millimeter waves for ITS applications," in *Proc. IEEE Veh. Technol. Conf.*, 1997, vol. 2, pp. 914–918.

[24] K. Tokuda, M. Akiyama, and H. Fujii, "DOLPHIN for inter-vehicle communication system," in *Proc. IEEE Intell. Veh. Symp.*, 2000, pp. 504–509.

[25] Y. Morioka, T. Sota, and M. Nakagawa, "An anti-car collision system using GPS and 5.8 GHz inter-vehicle communication at an off-sight intersection," in *Proc. IEEE Veh. Technol. Conf.*, 2000, vol. 5, pp. 2019–2024.

[26] N. Senda and M. Nakagawa, "Car detection using curved mirrors by automotive millimeter wave radar," in *Proc. Int. Conf. Inf. Commun. and Signal Process.*, 2003, vol. 2, pp. 1114–1118.

[27] A. Dogan, G. Korkmaz, Y. Liu *et al.*, "Evaluation of intersection collision warning system using an inter-vehicle communication simulator," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2004, pp. 1103–1108.

[28] I. R. Porche, K. S. Chang, W. Li *et al.*, "Real time task manager for communication and control in multicar platoons," in *Proc. Intell. Veh. Symp.*, 1992, pp. 409–414.

[29] N. Reijers, R. Cunningham, R. Meier *et al.*, "Using group communication to support mobile augmented reality applications," in *Proc. IEEE Int. Symp. Object-Oriented Real-Time Distrib. Comput.*, 2002, pp. 297–306.

[30] L. Chisalita and N. Shahmehri, "A peer-to-peer approach to vehicular communication for the support of traffic safety applications," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2002, pp. 336–341.

[31] P. Mohapatra, G. Chao, and J. Li, "Group communication in mobile *ad hoc* networks," *Computer*, vol. 37, no. 2, pp. 52–59, Feb. 2004.

[32] J. P. Singh, N. Bambos, B. Srinivasan *et al.*, "Wireless LAN performance under varied stress conditions in vehicular traffic scenarios," in *Proc. IEEE Veh. Technol. Conf.*, 2002, vol. 2, pp. 743–747.

[33] W. Kremer, "Vehicle density and communication load estimation in mobile radio local area networks (MR-LANs)," in *Proc. IEEE Veh. Technol. Conf.*, 1992, vol. 2, pp. 698–704.

[34] H. Fujii, M. Akiyama, and K. Tokuda, "Inter-vehicle communication protocol for group cooperative driving," in *Proc. IEEE Veh. Technol. Conf.*, 1999, vol. 4, pp. 2228–2232.

[35] A. Uno, T. Sakaguchi, and S. Tsugawa, "A merging control algorithm based on inter-vehicle communication," in *Proc. IEEE/IEEJ/JSAI Int. Conf. Intell. Transp. Syst.*, 1999, pp. 783–787.

[36] T. Sakaguchi, A. Uno, S. Kato *et al.*, "Cooperative driving of automated vehicles with inter-vehicle communication," in *Proc. IEEE Intell. Veh. Symp.*, 2000, pp. 516–521.

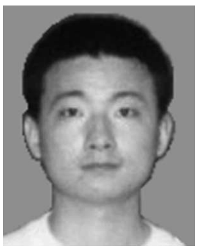
[37] Y. T. Lin, F.-Y. Wang, P. B. Mirchandani *et al.*, "Implementing adaptive driving systems for intelligent vehicles by using neuro-fuzzy networks," *Transp. Res. Rec.*, no. 1774, pp. 98–105, 2001.

[38] F.-Y. Wang, P. B. Mirchandani, and Z. Wang, "The VISTA project and its applications," *IEEE Intell. Syst.*, vol. 17, no. 6, pp. 72–75, Nov./Dec. 2002.

[39] L. Li and F.-Y. Wang, "Cooperative driving and lane changing at blind crossings," in *Proc. IEEE Intell. Veh. Symp.*, 2005, pp. 435–440.

[40] H. Sawant, J. Tan, Q. Yang, *et al.*, "Using Bluetooth and sensor networks for intelligent transportation systems," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2004, pp. 767–772.

[41] B.-W. Chuang, J.-H. Tarng, J. Lin *et al.*, "System development and performance investigation of mobile *ad-hoc* networks in vehicular environments," in *Proc. IEEE Intell. Veh. Symp.*, 2005, pp. 302–307.
[42] K. Tischler and B. Hummel, "Enhanced environmental perception by inter-vehicle data exchange," in *Proc. IEEE Intell. Veh. Symp.*, 2005, pp. 313–318.



Li Li (S'05–M'06) received the B.S. degree in electronics and information engineering from Huazhong University of Science and Technology, Wuhan, China, in 1999 and the M.S. degree in industrial engineering and the Ph.D. degree in systems engineering from the University of Arizona, Tucson, in 2003 and 2005, respectively.

He is now an Adjunct Assistant Professor of systems and industrial engineering with the University of Arizona, working in the fields of nonlinear systems, intelligent control and sensory, optimization and planning, intelligent vehicles, and intelligent transportation systems.

Fei-Yue Wang (S'89–M'90–SM'94–F'04) received the B.S. degree in chemical engineering from Qingdao University of Science and Technology, Qingdao, China, in 1982, the M.S. degree in mechanics from Zhejiang University, Hangzhou, China, in 1984, and the Ph.D. degree in electrical, computer, and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, in 1990.

He joined the University of Arizona in 1990 and became a Full Professor of systems and industrial engineering in 1999 and is currently the Director of the Program in Advanced Research for Complex Systems. In 1999, he found the Intelligent Control and Systems Engineering Center at the Institute of Automation, Chinese Academy of Sciences, Beijing, China, under the support of the Outstanding Oversea Chinese Talents Program. Since 2002, he has been the Director of the Key Laboratory of Complex Systems and Intelligence Science, Chinese Academy of Sciences. His current research interests include modeling, analysis, and control mechanism of complex systems, agent-based control systems, intelligent control systems, real-time embedded systems, application-specific operating systems, applications in intelligent transportation systems, intelligent vehicles and telematics, Web caching and service caching, smart appliances and home systems, and network-based automation systems. He has published more than 200 books, book chapters, and papers in those areas since 1984 and received more than $20M USD and over ¥50M RMB from NSF, DOE, DOT, NNSF, CAS, MOST, Caterpillar, IBM, HP, AT&T, GM, BHP, RVSI, ABB, and Kelon.

Dr. Wang is a member of Sigma Xi, ACM, AMSE, ASEE, and a Fellow of the International Council of Systems Engineering (INCOSE). He received the Caterpillar Research Invention Award with Dr. P. J. A. Lever in 1996 for his work in robotic excavation and the National Outstanding Young Scientist Research Award from the National Natural Science Foundation of China in 2001, as well as various industrial awards for his applied research from major corporations. He was the Editor-in-Chief of the *International Journal of Intelligent Control and Systems* from 1995 to 2000 and the Editor-in-Charge of *Series in Intelligent Control and Intelligent Automation* from 1996 to 2004. He is currently the Editor-in-Charge of *Series in Complex Systems and Intelligence Science*, an Associated Editor and the Editor for the Intelligent Transportation Systems Department of the *IEEE Intelligent Systems*, and an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, and several other international journals. He is an elected member of the IEEE SMC Board of Governors and the AdCom of the IEEE Nanotechnology Council, the President-Elect of the IEEE Intelligent Transportation Systems Society, and the Chair of the Technical Committee on System Complexity of the Chinese Association of Automation. He was the Program Chair of the 1998 IEEE International Symposium on Intelligent Control and the 2001 IEEE International Conference on Systems, Man, and Cybernetics, the Chair for Workshops and Tutorials for the 2002 IEEE International Conference on Decision and Control, the General Chair of the 2003 IEEE International Conference on Intelligent Transportation Systems, and a Co-Program Chair of the 2004 IEEE International Symposium on Intelligent Vehicles and the General Chair for the same conference in 2005, the General Chair of the IEEE 2005 International Conference on Networking, Sensing, and Control, ASME/IEEE International Conference on Mechatronic/Embedded Systems and Applications, and the Program Chair of IEEE International Conference on Vehicular Electronics and Safety. He was the Vice President and one of the major contributors of the American Zhu Kezhen Education Foundation and a Member of the Boards of Directors of five companies in information technology and automation.