2019 IEEE 58th Conference on Decision and Control (CDC)
Palais des Congrès et des Expositions Nice Acropolis
Nice, France, December 11-13, 2019

Exploiting Beneficial Information Sharing Among Autonomous Vehicles

Songyang Han Jie Fu Fei Miao

Abstract— **As communication technologies develop, an autonomous vehicle will receive information not only from its own sensing system but also from infrastructures and other vehicles through communication. This paper discusses how to exploit a sequence of future information that is shared among autonomous vehicles, including the planned positions, the velocities and the lane numbers. A hybrid system model is constructed, and a control policy is designed to utilize shared sequence information for making navigation decisions. For the high-level discrete state transitions, the shared information is used to determine when to change lane, if lane changing will bring reward for the autonomous vehicle and there exists a feasible continuous state controller. For the low-level continuous state space controller generation, the shared information can relax the safety interval constraints in the existing model predictive control method. In the system level, the information sharing can increase the traffic flow and improve driving comfort. We demonstrate the advantages of information sharing in control and navigation in simulation.**

Index Terms— **Autonomous vehicle, hybrid system, information sharing, control policy synthesis.**

I. Introduction

The connectivity between autonomous vehicles has the potential to significantly improve the perception systems to have a better sense of traffic. The development of the Dedicated Short-Range Communication (DSRC) technology enables Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications, as stated in the SAE J2735 standard [3]. The U.S. Department of Transportation (DOT) has estimated that V2V communication based on DSRC can address up to 82% of all crashes in the United States involving unimpaired drivers, potentially saving thousands of lives and billions of dollars [7]. The 5G technologies might also enable the access of cloud services and information sharing among vehicles and infrastructures.

However, existing control frameworks for autonomous vehicles or Connected Autonomous Vehicles (CAV) mainly focus on the controller design when a decision about lane changing or keeping has been provided [4]. For instance, Adaptive Cruise Control (ACC) [17] and Cooperative Adaptive Cruise Control (CACC) system have been designed to guarantee string stability of the platoon [18]. Cooperation schemes for two or more scattered vehicles are proposed to form platoons in a fuel-efficient manner [12]. The plug and play Model Predictive Control (MPC) method is proposed for a heavy duty vehicle platoon [5]. Platooning, ACC or CACC algorithms only consider longitudinal control without lateral control for multiple-lanes environment, when a vehicle needs to change lane to merge into or leave the platoon, continuous state space controllers are designed assuming lane changing decisions have already been given [20].

When an autonomous vehicle has additional knowledge, or gets extra knowledge about the environment based on V2V communication, how to make tactical decisions such as whether to change lane or keep lane, what requirements can be satisfied by the autonomous vehicle are still unsolved challenges. Furthermore, existing CAVs coordination and control approaches for scenarios such as cross intersections or merge lane [19], [10], [15] only consider sharing current states or Basic Safety Message (BSM), including current velocities and positions, whether sharing future planed velocity or trajectory can bring benefits remains unclear. Hence, it is critical to design a control policy that uses shared information to enhance operation performance for future CAVs, considers complicated practical environment and safety requirements, and shows benefits of V2V communication.

In this work, we explore the advantages raised by the extended sensing capability of autonomous vehicles through beneficial information sharing. We assume that both current and future planning information can be shared among neighbor autonomous vehicles, and design a tactical decision making rule about when to change lane based on shared information. A hybrid system model with controllable switching is constructed to study both the discrete state transitions and continuous dynamics of an autonomous vehicle. In simulation, we show that the control policy based on beneficial information sharing can help increase traffic flow and provide more comfortable driving experience.

The main contributions of this work are:

- To the best of our knowledge, this is the first work exploiting sharing a sequence of future information of autonomous vehicles' positions, velocities, and lane numbers in control policy synthesis. The information shared among autonomous vehicles can be generated by trajectory planning.
- For the discrete state transitions, the shared information is used to determine whether it is a good choice to change lane or not. For the continuous physical state dynamics, the shared information helps to relax the safety interval constraints and give more freedom to select feasible control inputs.
- The information sharing helps to improve driving comfort and traffic flow, which are analyzed through simulations.

The rest of this paper is organized as follows. In Section II,

This work was supported by NSF 1849246, NSF 1932250 grants.
S. Han and F. Miao are with the Department of Computer Science and Engineering, University of Connecticut, Storrs Mansfield, CT, USA 06268. Email: {songyang.han, fei.miao}@uconn.edu. Jie Fu is with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01604. Email:jfu2@wpi.edu.

we introduce the control and information sharing problem for CAVs considered in this work and a hybrid system model. In Section III, we develop a control policy synthesis algorithm to exploit benefits of information sharing in terms of both discrete state transitions and continuous state controllers. The simulation results are shown in Section IV with regard to traffic flow and driving comfort. The conclusions are given in Section V.

II. Problem Description

The V2V and V2I communications extend the information gathered by a single vehicle further beyond its own sensing system. This work explores how to use shared information to achieve benefits for an ego vehicle (the autonomous vehicle that we can control). The questions that need to be answered along this track include what is the information to be shared, to whom it is shared with, how to use the information, and what possible benefits could generate. The first two questions are addressed in this section, and the rest are addressed in Section III. This section introduces the shared information considered in this work. Afterwards, a hybrid system model is defined to describe the discrete sate transitions and continuous dynamics of an ego vehicle.

A. Information Sharing

Autonomous vehicles are assumed to share information with their ϵ-neighbors defined as follows.

Definition 1 (ϵ-neighbor)**.** One vehicle j is said to be the ϵ-neighbor of a vehicle i if $\mid x_t^i - x_t^j \mid \leq \epsilon$, where i and j are vehicles' indexes, x_t^i and x_t^j represent the longitudinal positions of vehicle i and j at time instant t respectively, and ϵ is a constant parameter. The set that includes all the ϵ-neighbors of the vehicle i except for i itself is denoted as $\mathscr{N}_i(\epsilon)$. This set is called vehicle i's ϵ-neighbors.

In this work, we consider autonomous vehicles driving on a 3-lane highway. Each autonomous vehicle is expected to share its current state and future plan, denoted by a sequence of $[x_t,\ x_{t+1},\ ...,\ x_{t+T}]$ (positions); $[v_t,\ v_{t+1},\ ...,\ v_{t+T}]$ (velocities); $[l_t,\ l_{t+1},\ ...,\ l_{t+T}]$ (lane numbers, labeled as 1, 2, 3) to its ϵ-neighbors. In these sequences, T is an integer parameter representing the time horizon. According to the current development of BSM and DSRC security, the message can be both authenticated and encrypted [7]. Hence, in this work, we assume that vehicular communication is true information that is not manipulated by attackers.

Parameter T determines how much future information is shared. There is a trade-off for selecting T. The larger T means there is more information shared among autonomous vehicles which may bring more benefits. However, the communication cost would increase and the information reliability may also decrease accordingly. There exist some experiments quantifying the communication delay using WiFi, 4G and 3G network [8]. Limited to the scope of this work, we assume that ϵ and T are given based on the communication capability, and how to select a better ϵ and T is one direction for future work.

B. Hybrid System Model

An ego vehicle is modeled as a hybrid system in Fig. 1. The influence of behaviors of other vehicles will trigger different response in the ego vehicle, referred to as different discrete states of operations such as lane keeping and lane changing. Specifically, there are three discrete states considered in this work.

Formally, the model of an ego vehicle is defined below:

Definition 2 (Hybrid System with Controllable Switching (HSCS))**.** A HSCS is a collection $H = (Q, X, U, \text{Init}, Inv, f, G, \delta)$ where

- $Q = \{q_0, q_1, \ldots, q_n\}$ is a finite set of discrete states.
- $X \subseteq \mathbb{R}^N$ is a compact set of continuous states.
- $U \subseteq \mathbb{R}^M$ is a compact set of the control inputs in continuous space.
- $\text{Init} \subseteq Q \times X$ is the set of initial states.
- $Inv : Q \to X$ assigns to each q an invariant set;
- $f : Q \times U \times X \to X$ assigns to each $q \in Q$ a continuous vector field $f(\cdot, q)$, a function from $X \times U$ to X.
- $G_{qq'} : Q \times Q \to X$ assigns each (q, q') a guard. A transition from state q to state q' is triggered when the continuous state is within $G_{qq'}$.
- δ: $X \times Q \to Q$ is a switching controller satisfying the following form:

  $$q(t^+) = \delta(q(t), x(t)), \text{ and } x(t^+) = x(t),$$

  where the switching controller δ assigns a hybrid system $(q(t), x(t))$ into a new discrete state $q(t^+)$ when $x(t) \in G_{q(t),q(t^+)}$.

In each discrete state, the local subsystem is a continuous state space model. The continuous state remains the same given the discrete switching δ. In this system, all switching is controllable.

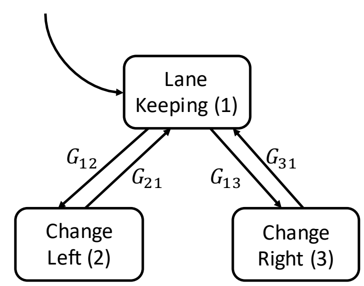


Fig. 1. The hybrid system model for an ego vehicle.

C. Discrete States

1) Lane Keeping: Lane Keeping (LK) is the initial state of each vehicle. Vehicles drive along the current lane in this state. The continuous controller will adjust the speed of the ego vehicle according to the safety interval requirements in this state, which is introduced in Sec. III-B. For instance, when the headway cannot support one vehicle's current speed v_t, it needs to slow down to avoid collision. This could happen when the ego vehicle is following a front vehicle or another vehicle merges in front of it.

2) Change Left & Change Right: In these two states, the ego vehicle changes to a neighbor lane on its left/right. Change Left (CL) is symmetric to Change Right (CR). Once the lane changing decision is made by the decision making algorithm (as introduced in Sec. III-A), this hybrid system enters CL/CR sate. The corresponding vehicle will execute the lane changing maneuver when it enters either state.

D. Continuous States & Control Inputs

The continuous behavior of an ego vehicle is described by a kinematic bicycle model, as shown in Fig. 2. This model can achieve a good balance between accuracy and complexity [6], [9].

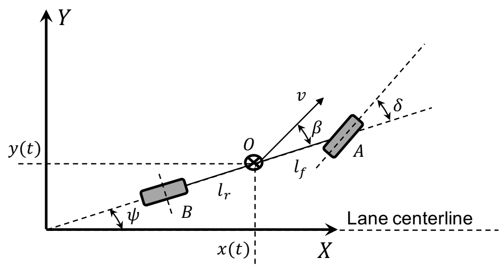


Fig. 2. Kinematic bicycle model attached to the lane center line.

Point A represents the two left and right front wheels, while the rear two wheels are represented by point B. Point O is the Center of Gravity (CoG). The lengths of the line segments OA and OB are represented by l_f and l_r respectively. The δ is the steering angle for the front wheels. The planar motion of this vehicle is described by three coordinates: x, y and ψ. (x, y) is the location of the CoG, and ψ illustrates the orientation of the vehicle. The X-axis represents the lane center line. The v is the velocity at the CoG and the slip angle β denotes its angle with OA.

The discrete-time equations of this model can be obtained by applying an explicit Euler method with a sampling time t_{cs} for continuous states [2]. The control vector for this vehicle is defined as $\mathbf{u}_t \triangleq [\delta_t, a_t]$, where a_t is its acceleration. The state vector is defined as $\mathbf{s}_t \triangleq [x_t, y_t, \psi_t, v_t, l_t]$, where l_t is the current lane number. The detail equations can be found in Appendix. More compactly, the update of the state vector is denoted as $\mathbf{s}_{t+1} = f(\mathbf{s}_t, \mathbf{u}_t)$.

III. Control Policy Synthesis

In this section, we design a control policy based on the HSCS model to exploit beneficial information sharing. For the discrete state transition control, i.e., whether to change lane or keep lane, shared information is utilized to make decisions according to the reward of lane changing and the guard conditions defined in this section. For the low-level continuous state control, there are some existing methods, whose performance can also be improved by information sharing, such as relaxing the constraints in an MPC controller.

A. Discrete State Control

The guard $G_{qq'}$ decides whether the transition from state q to state q' would take place or not. The discrete state transition policy is then proposed based on these guards. The shared information can be used to determine when lane changing is a better decision than lane keeping.

1) Guards: G_{12} specifies when the vehicle should make a decision to change left. The incentive to change lane is to achieve higher speed or avoid obstacle/traffic jam. A quality factor is defined to evaluate the future velocity quality of each lane based on shared velocity information. The factor corresponds to whether a vehicle could achieve higher speed after lane changing is defined as the following.

Definition 3 (Quality factor for the future velocity)**.** The quality factor for the future velocity of lane #l_k is

$$Q_v(l_k) = \frac{1}{N} \sum_{i:i\in\mathcal{N}_k} \sum_{j=1}^{T} \gamma^{(j-1)} v_{t+j}^i, \tag{1}$$

where $\mathcal{N}_k = \{i | i \in \mathcal{N}_i(\epsilon), l_t^i = l_k\}$, $N =| \mathcal{N}_k |$, $l_k \in \{1, 2, 3\}$, γ is a decay coefficient, v_{t+j}^i is the velocity of the vehicle i at time instant $(t + j)$.

As time increases, the larger j is, the less accuracy v_{t+j}^i would have. Therefore, a decay coefficient $\gamma < 1$ is multiplied to penalize future information. This quality factor has a good reflection of the velocity that an ego vehicle can achieve, and helps to avoid unnecessary lane changing. For example, even though at time t the vehicles on the neighbor lane have a higher speed than the ego vehicle, there may be a traffic jam in front of them. The ego vehicle could not observe this traffic jam because it could not see through its neighbors which block its view. In this case, there is no need to change lane.

It may make some passengers uncomfortable with frequent lane changes. Therefore, another quality factor is defined to evaluate the frequency of lane changing.

Definition 4 (Quality factor for the lane changing frequency)**.** The quality factor for the lane changing frequency is

$$Q_f = -\sum_{i=1}^{F} Change(t - i), \tag{2}$$

where t is the current time instant, F is a constant determining the window size of $[t - F, t - 1]$ and

$$Change(i) = \begin{cases} 1, & \text{if lane changing starts from time i;} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The larger this summation is, the less comfort the passengers may feel, which results in a smaller quality factor Q_f. The reward function for CL is defined as a weighted sum of the above two quality factors:

Definition 5 (Reward function for CL)**.** The reward function for CL is

$$r_{CL}(l_k) = w \cdot (Q_v(l_k - 1) - Q_v(l_k)) + Q_f, \tag{4}$$

where w is a weight that trades off two objectives, $Q_v(0)$ is defined to be 0 for convenience.

This weight should be determined by passengers' preference. If passengers do not like frequent lane change, then w is small; if passengers just want to arrive their destination as soon as possible without caring about how often lane changing happens, then w is large.

The ego vehicle would change left only if the corresponding reward is large enough. Therefore, we have

Definition 6 (The guard for the start of CL)**.** The guard for the start of CL is

$$G_{12} : r_{CL}(l_t) \geq \Theta_{CL}, \tag{5}$$

where Θ_{CL} is a predefined threshold.

Whenever the lane changing finishes, the vehicle will turn back to the LK state. The termination of this process can be observed by the lane changing trajectory, therefore,

Definition 7 (The guard for the end of CL)**.** The guard for the end of CL is

$$G_{21} : \mid y_t - y_l \mid \leq \epsilon_{cl}, \tag{6}$$

where y_t is the lateral position of the CoG, y_l is the lateral position of the left lane's center line, and ϵ_{cl} is a predefined small number.

Due to symmetry, $G_{13} : r_{CR}(l_t) \geq \Theta_{CR}$, where $r_{CR}(l_t)$ can be defined similarly with (4); $G_{31} : \mid y_t - y_r \mid \leq \epsilon_{cr}$ where y_r is the lateral position of the right lane's center line.

2) Discrete State Transition Policy: Based on the guards above, the discrete state transition policy of each HSCS can be synthesized as Algorithm 1. The time interval for this high-level controller is denoted by T_{ds}. If current state is LK, this algorithm will check $G_{12} \vee G_{13}$ for every T_{ds} time steps. If this condition holds, it means the reward of lane changing is considerable, then the ego vehicle will enter CL/CR state accordingly.

In most time instants, the ego vehicle stays in the LK state. The "Update continuous state dynamics in the LK state" in Algorithm 1 does not trigger to enter the CL/CR state or exit the LK state behavior, since the ego vehicle only updates continuous state space dynamics. Once the vehicle enters the lane changing state (CL or CR), the time step t and continuous state space dynamics would update according to that state. The discrete states will not be influenced by information sharing or transited until the lane changing maneuver finishes.

When determining lane changing, each vehicle just needs to consider CL/CR excluding changing 2 lanes continuously, e.g., from lane #1 to lane #3. What if the vehicle is in lane #1 with a low speed while lane #3 is totally open? In this case, the vehicles in lane #2 would definitely notice the open lane first and choose to change to lane #3. After they finishes lane changing, vehicles on lane #1 may start to change to lane #2 and may then change to lane #3 according to the corresponding reward. Vehicles in lane #1 do not need to worry about the quality factor on lane #3 at the beginning, because it cannot circumvent lane #2 and jump to lane #3.

Algorithm 1: Discrete state transition synthesis

Initialize the continuous states;
Initialize the discrete state to be the LK state;
for *every T_{ds} time steps* **do**
    if *current in CL / CR state* **then**
        Update in the current state until G_{21} / G_{31}.
    else
        if $G_{12} \vee G_{13}$ **then**
            if $r_{CL}(l_t) > r_{CR}(l_t)$ **then**
                Check the feasibility of CL by the continuous controller; if feasible, enter CL state until it finishes (G_{21}), then enter the LK state.
            else
                Check the feasibility of CR by the continuous controller; if feasible, enter CR state until it finishes (G_{31}), then enter the LK state.
            end
        else
            Update continuous state dynamics in the LK state;
        end
    end
end

In this way, the computation cost would be reduced by only focusing on two neighbor lanes.

Remark. *Algorithm 1 is the discrete state transition synthesis for each individual ego vehicle. If all the autonomous vehicles use this algorithm to make high-level decisions, it is neither practical nor necessary for them to update states synchronously. The random back-off algorithm in communication could be a good reference to generate random update sequence [13]. This can also help to avoid potential conflicts, for example, two vehicles determine to change lane simultaneously.*

B. Continuous State Control

Researchers have proposed various controllers for autonomous vehicles' trajectory following, e.g., Model Predictive Control (MPC) [2], [4], and the control barrier function based program [16], [1]. In this section, the MPC in [2] is taken as an example to show how the shared information helps to relax the constrains in the underlying continuous state controller.

For each T time horizon $\mathscr{T} = [t, t+1, ..., t+T-1]$, the control inputs can be generated by the following MPC:

$$\begin{aligned} \min_{\mathbf{u}_i, \mathbf{s}_{i+1}} \quad & \sum_{i \in \mathscr{T}} (\mathbf{s}_{i+1} - \mathbf{s}_{i+1}^{ref})^\mathsf{T} \mathbf{Q} (\mathbf{s}_{i+1} - \mathbf{s}_{i+1}^{ref}) + \mathbf{u}_i^\mathsf{T} \mathbf{R} \mathbf{u}_i \\ s.t. \quad & \mathbf{s}_{i+1} = f(\mathbf{s}_i, \mathbf{u}_i), \quad \forall i \in \mathscr{T} \\ & \mathbf{u}^{min} \leq \mathbf{u}_i \leq \mathbf{u}^{max}, \quad \forall i \in \mathscr{T} \\ & \dot{\mathbf{u}}^{min} \leq \mathbf{u}_{i+1} - \mathbf{u}_i \leq \dot{\mathbf{u}}^{max}, \ \forall i \in \mathscr{T} \\ & \mathbf{s}_{i+1}^{min} \leq \mathbf{s}_{i+1} \leq \mathbf{s}_{i+1}^{max}, \quad \forall i \in \mathscr{T} \end{aligned} \tag{7}$$

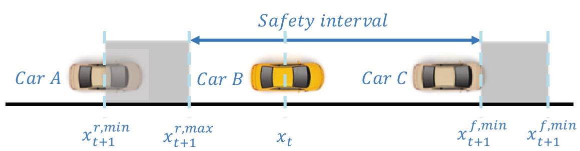
where $\mathbf{Q} \in \mathbb{R}^{5\times5}$, $\mathbf{R} \in \mathbb{R}^{2\times2}$ are positive definite weighting matrices for tuning; $\mathbf{s}_{i+1}^{ref}$ is the reference trajectory, $\mathbf{s}_{i+1}^{min}$ and $\mathbf{s}_{i+1}^{max}$ are the bounds of each state:

$$\mathbf{s}_{i+1}^{min} = [x_{i+1}^{min}, -w_{lane}, -\pi/2, 0, 1]^{\mathsf{T}}, \tag{8}$$

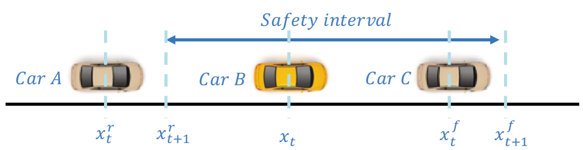
$$\mathbf{s}_{i+1}^{max} = [x_{i+1}^{max}, w_{lane}, \pi/2, v^{max}, 3]^{\mathsf{T}}, \tag{9}$$

where w_{lane} represents the width of each lane.

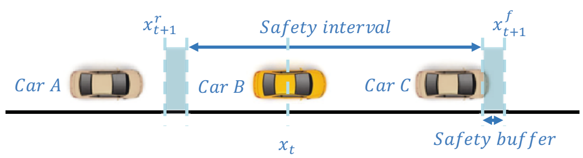
The following two subsections describe how shared information is used to calculate x_{i+1}^{min} and x_{i+1}^{max} in lane keeping and lane changing states respectively.



(a) Based on prediction. The safety interval is obtained based on the position range prediction of car A and car C.



(b) Based on information sharing. The safety interval is obtained based on the trajectory planning of car A and C. Its range is no smaller than the prediction-based safety interval.



(c) Refined by a safety buffer. Both the prediction and shared information are not accurate enough, and there might be a latency caused by communication and control implementation.

Fig. 3. The safety interval for the ego car B is the position bound in Equ. (7). To elaborate the idea of safety interval, each car is treated as a mass point at the CoG in this figure.

1) Lane Keeping: In order to guarantee driving safety, there should not be overlap in positions of all the vehicles. To elaborate the idea of safety interval, we first ignore the length $(l_f + l_r)$ of each vehicle and treat each vehicle as a mass point at the CoG. The state bound $[x_{i+1}^{min}, x_{i+1}^{max}]$ is used to guarantee the planned path locating within the safety interval. As shown in Fig. 3(a), for autonomous car B, x_{t+1} should satisfy $x_{t+1}^{r,max} \leq x_{t+1} \leq x_{t+1}^{f,min}$, where $x_{t+1}^{r,max}$ is the maximum position of the rear car (represented by "r" in the superscript) at time $t+1$ and $x_{t+1}^{f,min}$ is the minimum position of the front car (represented by "f" in the superscript) at time $t+1$. This range is called the safety interval for car B at time $t+1$. The safety interval can ensure there is no position overlap between the ego vehicle and its front/rear neighbor. This safety interval is predicted based on probability density functions in [2].

If the position information $[x_t, x_{t+1}, ..., x_{t+T}]$ is shared among car A, B and C, then the safety interval can be directly obtained as $[x_{i+1}^r, x_{i+1}^f]$ for each i, as shown in Fig. 3(b). No matter what kind of prediction is used, because $x_{t+1}^r \leq x_{t+1}^{r,max} \leq x_{t+1} \leq x_{t+1}^{f,min} \leq x_{t+1}^f$, the information sharing could give a larger safety interval (at lease equal to), like car C in Fig.3(a). A larger safety interval would bring more freedom to the control inputs of the ego vehicle, which would hopefully generate smaller control cost for the optimal solution of (7).

Both prediction and shared information may not be accurate enough. Also, there might be a latency caused by communication and control implementation. Therefore, a small safety buffer is added to improve safety under these uncertainties, as shown in Fig. 3(c).

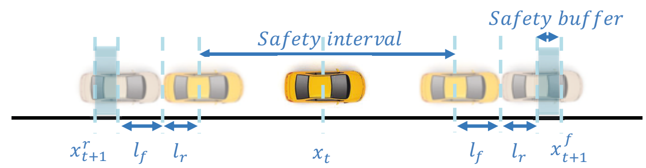


Fig. 4. The safety interval used in Equ. (7) is the information-sharing-based interval refined by safety buffers and cars' length.

Now we add these cars' length back in Fig. 4. The physical positions of l_f and l_r can be found in Fig. 2. Finally, with information sharing, the safety interval is selected as

$$[x_{i+1}^r + l_{bf} + l_f + l_r, x_{i+1}^f - l_{bf} - l_f - l_r],$$

where l_{bf} is the length of the safety buffer.

2) Lane Changing: The MPC program (7) can also be used to validate the feasibility of a lane change at time t, and generate control inputs if there are feasible solutions. Similar to the lane keeping case, the shared information is utilized to refine the safety intervals on both the current lane and target lane.

During lane changing, the continuous state bounds should consider the safety intervals on both the current lane and the target lane. As shown in Fig. 5, the ego car D is going to change left. The safety interval used in Fig. 5 is the intersection of the two intervals on lane #1 and #2, which is $[x_{t+1}^A + l_{bf}, x_{t+1}^B - l_{bf}]$ in this case. If there is another car in parallel to the ego vehicle on the target lane, simply set the safety interval to be $[0, 0]$. Then the MPC program (7) will return an infeasible result to the Algorithm 1.

As long as the two cars have some overlap along the longitudinal axis, they are called in parallel no matter how small it is. When a car is changing lane, it is assumed to be occupying two lanes (the current lane and the target lane) simultaneously until this maneuver finishes. For example, once car D starts to change to lane #1, it is considered to be occupying both lane #1 and #2 during this process. This assumption would influence the information received by other vehicles. When plugging in (7), the length of each car is also added back, similarly with Fig. 4

C. Benefits of Information Sharing

Each autonomous vehicle is expected to share its current and planned positions, velocities, lane numbers in the future

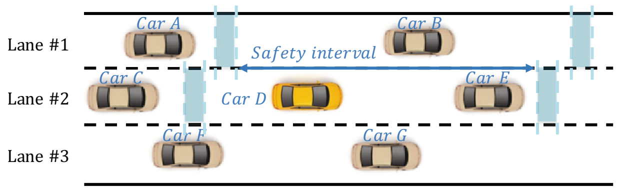


Fig. 5. Car D is changing from lane #2 to lane #1. Its safety interval is the intersection of the safety intervals on both the current lane (#2) and target lane (#1). Here the length of each car is ignored for simplicity.

T time steps. These planned future information can only be obtained through information sharing among vehicles, rather than measuring by sensors. For each individual ego vehicle, the utilization of shared information is:

- In the discrete state transition, shared information is used to calculate quality factor for the future velocity of different lanes, which could help achieve higher velocity and avoid unnecessary lane changing.
- In the continuous state controller calculation, shared information can help to improve the performance of existing methods. For example, the safety interval could be enlarged (or at lease equal to) in MPC.

In the system level, the performance of information sharing can be evaluated by the following criteria:

1) Traffic Flow: The traffic flow can reflect the quality of road throughout with respect to the traffic density. The traffic density ρ is the ratio between the total number of vehicles and the length of the road. The traffic flow is calculated as

$$Q = \rho \times \bar{v}, \tag{10}$$

where $\bar{v}$ is the average velocity of all the vehicles [21].

The quality factor for the future velocity can guide the vehicle to a lane with a higher velocity. The larger safety interval gives more freedom to the controller, which may generate a larger velocity. Therefore, the traffic flow is expected to be larger with information sharing.

2) Average Driving Comfort Cost: Define the driving comfort cost with respect to the vehicle i's acceleration a_{it} at time t as follows:

$$Cost(a_{it}) = \begin{cases} 1, & \text{if } a_{it} < \Theta_a; \\ 2, & \text{if } a_{it} \geq \Theta_a; \\ 3, & \text{lane changing,} \end{cases} \tag{11}$$

where Θ_a is a predefined threshold.

The Average Driving Comfort Cost (ADCC) is the average driving comfort costs of all the autonomous vehicles on the road and all the time instants. It is represented by

$$\frac{1}{N^{total} \cdot t^{max}} \sum_i \sum_t Cost(a_{it}), \tag{12}$$

where N^{total} is the total number of all the vehicles on the road. This ADCC is modified based on the effort definition in [11] by adding a term to deal with lane changing. The quality factor for the future velocity is used to avoid unnecessary lane changing. The quality factor for the lane changing frequency is further used to improve the driving comfort. Therefore, the ADCC is expected to be smaller with information sharing.

IV. SIMULATION

The benefits of information sharing for individual vehicle can be explicitly found from the definition of the reward function (4) and the MPC program (7). The benefits in system level is shown by simulation in this section. In our simulation, the initial positions of all the vehicles are randomly scattered on different lanes. The simulation platform is MATLAB, and there are some standard solvers for nonlinear programming [14]. The length of the highway is 1000. When a vehicle reaches the end of this highway, it will continue driving from the start. It works like there is a portal at the end of the road that sends all the vehicles to the start of this road. The total number of vehicles ranges from 100 to 900. For different traffic densities, simulation will run for 2000 time steps. The two criteria defined in Sec. IV are calculated based on the statistics in the last 1000 time steps. Also, the simulation runs 30 times under different initialization for each traffic density.

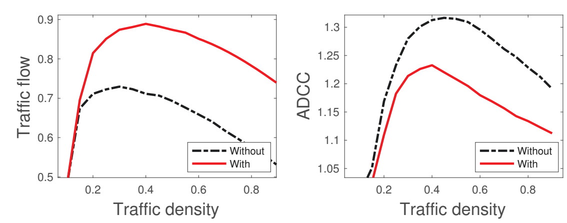


Fig. 6. The comparison between with and without information sharing. As the traffic density grows, the information sharing gives larger traffic flow and better driving comfort (a smaller ADCC value).

As shown in Fig. 6, the two criteria are compared between with and without information sharing. With information sharing among autonomous vehicles, the discrete state transition is based on Algorithm 1, where the reward function is used; the MPC program is using the safety interval generated by shared information. Without information sharing, the lane changing is determined by the velocities of the neighbors; the safety interval is constructed based on predictions. In Fig. 6, when the traffic density ρ is small, there is small difference between the two methods, because all the vehicles have enough space to drive at a high speed and there is almost no need to change lane. As ρ grows, the information sharing could give a larger traffic flow and smaller ADCC, which satisfies the analysis in Sec. III-C.

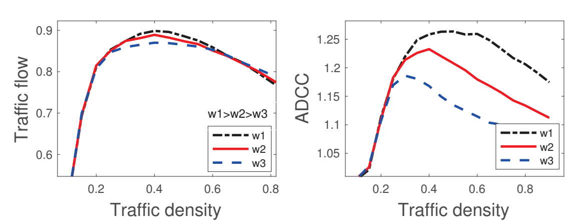


Fig. 7. The comparison between different weights. The weight w trades off two objectives in reward function (4). The larger w_1 gives a larger traffic flow. The smaller w_3 gives better driving comfort or a smaller ADCC value.

The performance of the control policy is influenced by the trade-off weight w in the reward function (4). Fig. 7 compares traffic flow and ADCC under different weights. Similarly, there is small difference when ρ is small. As ρ grows, the larger w gives a larger traffic flow but with a larger ADCC. Because the larger w means the vehicle would be more willing to change lane when it could reach a higher speed. The smaller w considers more about the lane change frequency, therefore lead to a better ADCC while sacrificing the traffic flow.

V. CONCLUSION

This paper explores the control policy and benefits when sharing information among autonomous vehicles. We propose to share a T-time-step future information among autonomous vehicles' ϵ-neighbors besides current state informaton. A HSCS is defined to study the discrete and dynamic behavior of an autonomous vehicle. In the high-level discrete state transition control, the shared information is used to evaluate the reward of lane changing. In the low-level continuous state control, the shared information can relax the state bounds determined by the safety interval. In simulation, the information sharing can improve the traffic flow and driving comfort when the traffic density is large enough. Therefore, the T-time-step future information sharing among autonomous vehicles and the control policy developed in this work shows advantages compared with existing frameworks. Formal method-based and reachability analysis-based approach could be used to validate the safety of this hybrid system in future work.

REFERENCES

[1] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2017.

[2] G. Cesari, G. Schildbach, A. Carvalho, and F. Borrelli. Scenario model predictive control for lane change assistance and autonomous driving on highways. *IEEE Intelligent Transportation Systems Magazine*, 9(3):23–35, 2017.

[3] D. Committee et al. Dedicated short range communications (dsrc) message set dictionary. *SAE Standard J*, 2735:2015, 2009.

[4] J. Guanetti, Y. Kim, and F. Borrelli. Control of connected and automated vehicles: State of the art and future challenges. *Annual Reviews in Control*, 45:18 – 40, 2018.

[5] H. Hu, Y. Pu, M. Chen, and C. J. Tomlin. Plug and play distributed model predictive control for heavy duty vehicle platooning and interaction with passenger vehicles. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 2803–2809, Dec 2018.

[6] C. M. Kang, S.-H. Lee, and C. C. Chung. Comparative evaluation of dynamic and kinematic vehicle models. In *2014 IEEE Conference on Decision and Control (CDC)*, pages 648–653. IEEE, 2014.

[7] J. B. Kenney. Dedicated short-range communications (dsrc) standards in the united states. *Proceedings of the IEEE*, 99(7):1162–1182, July 2011.

[8] S.-W. Kim, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus. The impact of cooperative perception on decision making and planning of autonomous vehicles. *IEEE Intelligent Transportation Systems Magazine*, 7(3):39–50, 2015.

[9] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli. Autonomous driving using model predictive control and a kinematic bicycle vehicle model. In *Intelligent Vehicles Symposium, Seoul, Korea*, 2015.

[10] J. Lee and B. Park. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):81–90, March 2012.

[11] N. Li, D. W. Oyler, M. Zhang, Y. Yildiz, I. Kolmanovsky, and A. R. Girard. Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems. *IEEE Transactions on control systems technology*, 26(5):1782–1797, 2018.

[12] K. Y. Liang, J. Mrtensson, and K. H. Johansson. Heavy-duty vehicle platoon formation for fuel efficiency. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1051–1061, April 2016.

[13] Y. Liang, X. Li, J. Zhang, and Z. Ding. Non-orthogonal random access for 5g networks. *IEEE Transactions on Wireless Communications*, 16(7):4817–4831, 2017.

[14] D. G. Luenberger and Y. Ye. *Linear and nonlinear programming*. Springer, 2015.

[15] X. Meng and C. G. Cassandras. Optimal control of autonomous vehicles for non-stop signalized intersection crossing. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6988–6993, Dec 2018.

[16] Q. Nguyen and K. Sreenath. Exponential control barrier functions for enforcing high relative-degree safety-critical constraints. In *2016 American Control Conference (ACC)*, pages 322–328. IEEE, 2016.

[17] P. Nilsson, O. Hussien, A. Balkan, Y. Chen, A. D. Ames, J. W. Grizzle, N. Ozay, H. Peng, and P. Tabuada. Correct-by-construction adaptive cruise control: Two approaches. *IEEE Transactions on Control Systems Technology*, 24(4):1294–1307, July 2016.

[18] J. Ploeg, D. P. Shukla, N. van de Wouw, and H. Nijmeijer. Controller synthesis for string stability of vehicle platoons. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):854–865, April 2014.

[19] J. Rios-Torres and A. A. Malikopoulos. A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1066–1077, May 2017.

[20] G. Schildbach and F. Borrelli. Scenario model predictive control for lane change assistance on highways. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 611–616, June 2015.

[21] W.-X. Zhu and H. Zhang. Analysis of mixed traffic flow with human-driving and autonomous cars based on car-following model. *Physica A: Statistical Mechanics and its Applications*, 496:274–285, 2018.

APPENDIX

Kinematic Bicycle Model

The control vector for this vehicle is defined as $\mathbf{u}_t \triangleq [\delta_t, a_t]$, where a_t is its acceleration. The state vector is defined as $\mathbf{s}_t \triangleq [x_t, y_t, \psi_t, v_t, l_t]$, where l_t is the current lane number. The detail equations can be found in [2] as follows:

$$\begin{cases} \dot{x}_t & = v_t cos(\psi_t + \beta_t), \\ x_{t+1} & = x_t + \dot{x}_t t_{cs}, \\ \dot{y}_t & = v_t sin(\psi_t + \beta_t), \\ \tilde{y}_{t+1} & = y_t + \dot{y}_t t_{cs}, \\ y_{t+1} & = \tilde{y}_{t+1} + \begin{cases} -w_{lane}, & \text{if } \tilde{y}_{t+1} \geq \frac{w_{lane}}{2}; \\ w_{lane}, & \text{if } \tilde{y}_{t+1} < -\frac{w_{lane}}{2}; \\ 0, & \text{otherwise.} \end{cases} \\ \psi_{t+1} & = \psi_t + \frac{v_t cos\beta_t tan\delta_t}{l_f + l_r} \cdot t_{cs}, \\ v_{t+1} & = v_t + a_t t_{cs}, \\ l_{t+1} & = l_t + \begin{cases} 1, & \text{if } \tilde{y}_{t+1} \geq \frac{w_{lane}}{2}; \\ -1, & \text{if } \tilde{y}_{t+1} < -\frac{w_{lane}}{2}; \\ 0, & \text{otherwise,} \end{cases} \end{cases} \tag{13}$$

where β_t is the slip angle. Once the control variable δ_t is determined, this angle can be updated as $\beta_t = tan^{-1}\left(\frac{l_r}{l_f + l_r} \cdot tan\delta_t\right)$. More compactly, the update of the state vector is denoted as $\mathbf{s}_{t+1} = f(\mathbf{s}_t, \mathbf{u}_t)$ in the MPC program (7).