Safe Distributed Lane Change Maneuvers for Multiple Autonomous Vehicles Using Buffered Input Cells

Mingyu Wang[1], Zijian Wang[2], Shreyasha Paudel[3], Mac Schwager[2]

***Abstract*— This paper introduces the Buffered Input Cell as a reciprocal collision avoidance method for multiple vehicles with high-order linear dynamics, extending recently proposed methods based on the Buffered Voronoi Cell [1] and generalized Voronoi diagrams [2]. We prove that if each vehicle's control input remains in its Buffered Input Cell at each time step, collisions will be avoided indefinitely. The method is fast, reactive, and only requires that each vehicle measures the relative position of neighboring vehicles. We incorporate this collision avoidance method as one layer of a complete lane change control stack for autonomous cars in a freeway driving scenario. The lane change control stack comprises a decision-making layer, a trajectory planning layer, a trajectory following feedback controller, and the Buffered Input Cell for collision avoidance. We show in simulations that collisions are avoided with multiple vehicles simultaneously changing lanes on a freeway. We also show in simulations that autonomous cars using the BIC method effectively avoid collisions with an aggressive human-driven car.**

I. INTRODUCTION

According to current statistics, autonomous cars are *more likely* to be involved in a collision than human-driven cars per mile, but they are *less-often* found to be at fault [3], [4]. One explanation for this paradoxical phenomenon is that current autonomous driving polices do not adequately predict or correctly account for the future control actions of other vehicles. This hypothesis is also supported in an analysis of a collision during the DARPA urban challenge [5], [6]. In this paper we propose a reciprocal strategy that, when integrated into an autonomous driving control stack, can effectively avoid collisions while taking into account the control actions of other drivers.

The key to the collision avoidance strategy is for each vehicle to compute a set of allowable inputs, called the Buffered Input Cell (BIC), that provides a *reciprocal* guarantee—as long as all vehicles' inputs remain in their BICs at the current time step, there will be no collisions at the next time step. Each vehicle can compute its BIC quickly knowing only the relative positions of neighboring vehicles (unlike other methods that require velocities), and then projects its commanded control input into this set to avoid collisions. Our BIC concept is inspired by the Buffered Voronoi Cell (BVC)

This work was supported in part by the Toyota SAIL Center for AI Research. We are grateful for this support.

[1]Mingyu Wang is with the Department of Mechanical Engineering, Stanford University, Stanford, CA 94305, USA, mingyuw@stanford.edu.

[2]Zijian Wang and Mac Schwager are with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA, {zjwang, schwager}@stanford.edu.

[3] Shreyasha Paudel is with the Ford Research and Innovation Center, Palo Alto, CA 94304, USA, shreyashapdl@gmail.com.

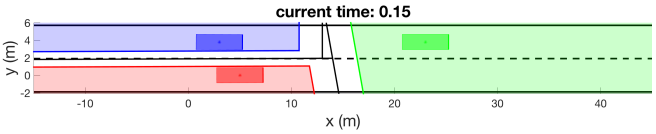

Fig. 1. An example of three autonomous vehicles using Buffered Input Cells (BIC) to avoid collision in a freeway driving scenario. The buffered Voronoi cell (BVC) of the corresponding vehicles are shaded in different colors. The BIC is obtained by mapping the BVC constraint in the position space into the control input space using vehicle dynamics. Each vehicle chooses its control input to be in its own BIC to avoid collision.

[1] for collision avoidance with single integrator robots. Our method generalizes the BVC to vehicles with higher order dynamics and adapts generalized Voronoi diagrams [2] to deal with non-circular geometry, both of which are crucial to freeway driving. We give a proof of collision avoidance with the BIC for vehicles with linear dynamics of any order under a controllability-like assumption. We incorporate the BIC collision avoidance strategy as one layer in a lane change control stack for autonomous freeway driving, and show in simulations that collisions are avoided with multiple vehicles performing simultaneous lane change maneuvers on a freeway. We also compare simulation performance with a naive model predictive control (MPC) lane change controller, and find that the MPC controller sometimes leads to collisions, while our BIC controller always avoids collisions. We also show that the algorithm effectively avoids collisions with an aggressive human driver, even though the human driver violates the reciprocal assumptions of the BIC.

A large body of research in collision avoidance in the driving context is concerned with path planning in the presence of other vehicles and road structures [7]–[9]. A sampling-based approach is found to be effective in real-time motion planning for autonomous vehicles when combined with closed-loop predictions [10]. Some other work also focuses on trajectory planning [11], and the tunability and stability of the planning is presented in [12]. However, most planning algorithms [13] carry the assumption that other vehicles maintain constant speeds and thus fail to account for dynamic traffic environments. One possible solution to this conundrum is to use reciprocal collision avoidance approaches such as Reciprocal Velocity Obstacle (RVO) [14], [15]. In [16], a similar approach to our BIC method is proposed, where the authors extend RVO by projecting position constraints into control space. However, RVO methods require measuring relative position and velocity of other vehicles, while our BIC method only requires position measurements. A nonlinear control theoretic tool called Control Barrier Functions (CBF) is introduced in [17], [18] to prevent collisions in a minimally

invasive fashion by solving a quadratic programming (QP). In contrast, our method uses a reciprocal assumption about the actions of the other agents, and also takes into account the physical geometries of the agents. Additionally, intention models of the traffic participants are shown to be crucial for autonomous driving, and are commonly dealt with by partially observable Markov decision processes (POMDPs) [19], [20], which characterize the uncertainty in knowledge of other drivers' policies. We do not explicitly model other vehicles' policies, but instead focus on provable safety by providing a deterministic method that relies on the BIC reciprocal constraint assumption.

The remainder of this paper is organized as follows. In Sec. II we define the Buffered Input Cell and prove properties about collision avoidance. We describe the lane change control stack with the BIC layer in Sec. III. We demonstrate our method as applied to simulated vehicles and compare performance with an MPC lane change controller in Sec. IV and offer concluding remarks in Sec. V.

II. Buffered Input Cell for Linear Systems

In this section, we describe our core technical contribution, the Buffered Input Cell (BIC), which is developed based on the previously introduced concepts of generalized Voronoi diagrams [2] and the Buffered Voronoi Cell (BVC) [1]. The BIC extends the collision avoidance properties of the BVC to general linear systems by mapping the constraints in position space into the control input space.

A. Collision Avoidance with the Buffered Voronoi Cell

Consider N robots whose physical geometries are given by closed convex sets $\mathcal{O}_i^t$. First define the distance between two convex sets:

***Definition** 1: (Distance Between Two Convex Sets)* Let $\mathcal{A}$ and $\mathcal{B}$ be two closed convex sets, the distance between the two sets is the distance between the closest points in the two sets.

$$d(\mathcal{A},\mathcal{B}) := \min \|a^* - b^*\|, \text{ s.t. } a^* \in \mathcal{A}, b^* \in \mathcal{B} \quad (1)$$

Accordingly, a *collision-free* configuration of all N robots is defined as:

***Definition** 2: (Collision-free Configuration)* A collision-free configuration of a group of N robots is one where

$$d(\mathcal{O}_i,\mathcal{O}_j) > 0, \quad \forall i \neq j$$

Recall in collision-free configurations, the generalized Voronoi cell of robot i is defined as the set of points $\mathbf{q} \in \mathbb{R}^d$ which satisfies:

$$\mathcal{V}_i := \{\mathbf{q} \mid \|\mathbf{q}-\mathbf{p}_i^*\| < \|\mathbf{q}-\mathbf{p}_j^*\|, \|\mathbf{p}_i^*-\mathbf{p}_j^*\| = d(\mathcal{O}_i,\mathcal{O}_j), \mathbf{p}_i^* \in \mathcal{O}_i, \mathbf{p}_j^* \in \mathcal{O}_j \quad \forall j \neq i\}. \quad (2)$$

Note that a robot is contained in its generalized Voronoi cell in a collision-free configuration (see Lemma 1) [2]. The generalized Voronoi cell for robot i is an open set that excludes the generalized Voronoi edge. The generalized Voronoi edge is a function of the geometries and configuration of robot i and its neighbors, and hence changes dynamically as the robots move. In order to avoid cumbersome notation, we denote this dependence by $\mathcal{V}_i^t$. We then define the Buffered Voronoi Cell[1] (BVC) [21] by retracting the boundary of the generalized Voronoi cell by the body of the robot,

$$\bar{\mathcal{V}}_i^t := \mathcal{V}_i^t \ominus (\mathcal{O}_i^t - \mathbf{p}_i^t), \quad (3)$$

where $\mathcal{A} \ominus \mathcal{B}$ denotes the Minkowski difference of two sets $\mathcal{A}$ and $\mathcal{B}$:

$$\mathcal{A} \ominus \mathcal{B} = \{\mathbf{c} \mid \mathbf{c} + \mathcal{B} \subseteq \mathcal{A}\}$$

Note that this is similar to "growing the obstacles" to compute the configuration space for path planning [22], [23].

By definition of the Minkowski difference, the BVC has the property that if the robot center position is within the BVC at time t, then the whole body of the robot is within its Voronoi cell at time t. Under the assumption that the robot orientation does not change, this can be used for reciprocal collision avoidance because if every robot stays inside its current BVC at the next time step, $\mathbf{p}_i^{t+1} \in \bar{\mathcal{V}}_i^t$ for all t, then collisions are prevented indefinitely. A crucial property of the BVC is stated below:

***Lemma** 1: (Property of the BVC)* A group of N robots with center positions $\mathbf{p}_i$ are in a collision-free configuration, if and only if their center positions $\mathbf{p}_i \in \bar{\mathcal{V}}_i^t$, where $\bar{\mathcal{V}}_i^t$s $(i = 1, 2, \ldots, N)$ are the Buffered Voronoi Cells from a collision-free configuration.

Proof: Firstly, by the BVC definition given in (3), it follows directly that $\mathbf{p}_i \in \bar{\mathcal{V}}_i^t \Leftrightarrow \mathcal{O}_i \subseteq \mathcal{V}_i^t$. Thus, to prove the result, we need to show that $\mathcal{O}_i \subseteq \mathcal{V}_j^t, \mathcal{O}_j \subseteq \mathcal{V}_j^t \Leftrightarrow d(\mathcal{O}_i,\mathcal{O}_j) > 0, \forall j \neq i$. We prove $d(\mathcal{O}_i,\mathcal{O}_j) > 0 \Rightarrow \mathcal{O}_i \subseteq \mathcal{V}_i^t$ by contradiction. Consider robots i and j that $d(\mathcal{O}_i,\mathcal{O}_j) = ||\mathbf{p}_i^* - \mathbf{p}_j^*|| > 0$. Assume $\exists\, \mathbf{a} \in \mathcal{O}_i$, s.t. $\mathbf{a} \notin \mathcal{V}_i$, i.e. $\|\mathbf{a} - \mathbf{p}_i^*\| \geq \|\mathbf{a} - \mathbf{p}_j^*\|$. It is obvious that $\mathbf{a} \neq \mathbf{p}_i^*$.

$$\begin{aligned}
&\|\mathbf{a} - \mathbf{p}_i^*\| \geq \|\mathbf{a} - \mathbf{p}_j^*\| \\
\Leftrightarrow &\|\mathbf{a} - \mathbf{p}_i^*\|^2 \geq (\mathbf{a} - \mathbf{p}_j^*)^T(\mathbf{a} - \mathbf{p}_j^*) \\
\Leftrightarrow &\|\mathbf{a} - \mathbf{p}_i^*\|^2 \geq (\mathbf{a} - \mathbf{p}_i^* + \mathbf{p}_i^* - \mathbf{p}_j^*)^T(\mathbf{a} - \mathbf{p}_i^* + \mathbf{p}_i^* - \mathbf{p}_j^*) \\
\Leftrightarrow &2(\mathbf{a} - \mathbf{p}_i^*)^T(\mathbf{p}_i^* - \mathbf{p}_j^*) + (\mathbf{p}_i^* - \mathbf{p}_j^*)^T(\mathbf{p}_i^* - \mathbf{p}_j^*) \leq 0 \\
\Rightarrow &(\mathbf{a} - \mathbf{p}_i^*)^T(\mathbf{p}_j^* - \mathbf{p}_i^*) \geq 0
\end{aligned}$$

Since $\mathcal{O}_i$ is convex, according to Eqn. (1), we have

$$||\alpha\mathbf{a} + (1-\alpha)\mathbf{p}_i^* - \mathbf{p}_j^*|| \geq ||\mathbf{p}_i^* - \mathbf{p}_j^*||, \forall \alpha \in [0,1]\,. \quad (4)$$

Substitute $\alpha = \frac{(\mathbf{a}-\mathbf{p}_i)^T(\mathbf{p}_j-\mathbf{p}_i)}{(\mathbf{a}-\mathbf{p}_i)^T(\mathbf{a}-\mathbf{p}_i)} \geq 0$, Eqn.(4) becomes $\frac{((\mathbf{a}-\mathbf{p}_i)^T(\mathbf{p}_j-\mathbf{p}_i))^2}{(\mathbf{a}-\mathbf{p}_i)^T(\mathbf{a}-\mathbf{p}_i)} \leq 0$, which contradicts $||\mathbf{p}_i^* - \mathbf{p}_j^*|| > 0$ and $\mathbf{a} \neq \mathbf{p}_i^*$. Thus, we have $d(\mathcal{O}_i,\mathcal{O}_j) > 0 \Rightarrow \mathcal{O}_i \subseteq \mathcal{V}_i^t$. The proof of $\mathcal{O}_i \subseteq \mathcal{V}_i^t, \mathcal{V}_j^t, \mathcal{O}_j \subseteq \mathcal{V}_j^t \Rightarrow d(\mathcal{O}_i,\mathcal{O}_j) > 0$. $d(\mathcal{O}_i,\mathcal{O}_j) > 0 \Leftarrow \mathcal{O}_i \subseteq \mathcal{V}_i, \mathcal{V}_j^t, \mathcal{O}_j \subseteq \mathcal{V}_j^t$ follows straightforward from Eqn. (3). Thus, the results follows. ■

For single integrator robots [1], ensuring that $\mathbf{p}_i^{t+1} \in \bar{\mathcal{V}}_i^t$ is straightforward, however for high order robot dynamics, we require a different technique.

[1]The definition we give here is more general than the previous definition from [1] equation (1) and (3). The two definitions are equivalent in the case that $\mathcal{O}_i$ is a ball of radius r.

B. Collision Avoidance with the Buffered Input Cell

We propose the Buffered Input Cell (BIC) to ensure collisions are avoided even when the robots have translational linear dynamics of arbitrary order. High order linear models can capture momentum and other complex dynamical effects that are particularly important in freeway driving. Instead of enforcing the collision-free constraint in the position space, we produce a constraint in the control input space by exploiting the robot dynamics. We follow a procedure similar to the generalized velocity obstacle concept from [16].

Consider the following discrete linear dynamics:

$$\mathbf{x}_i^{t+1} = F_i \mathbf{x}_i^t + G_i \mathbf{u}_i^t, \tag{5}$$

where $\mathbf{x}_i^t \in \mathbb{R}^n$ is the state, $\mathbf{u}_i^t \in \mathbb{R}^m$ denotes the control input, and F_i and G_i are matrices from the robot model that are known by robot i, but not by any other robot. We assume in this linear model that the orientation of the vehicle does not change, which is approximately true for linearized vehicle dynamics at freeways speeds. The position of the ith vehicle $\mathbf{p}_i^t$ is related to the vehicle state by the linear mapping (again only known by robot i),

$$\mathbf{p}_i^t = C_i \mathbf{x}_i^t. \tag{6}$$

Substituting (5) into (6), we obtain

$$\mathbf{p}_i^{t+1} = C_i F_i \mathbf{x}_i^t + C_i G_i \mathbf{u}_i^t. \tag{7}$$

Define a matrix $J_i := C_i G_i$. We require the following assumption related to controllability.

***Assumption** 1: (Strong Controllability)* The robot dynamics are such that $J_i = C_i G_i$ is invertible for all robots i.

Note that a necessary condition for Assumption 1 is that there are an equal number of input dimensions and position dimensions, $m = d$. We find in Sec. III-E that this condition is satisfied for a linearized kinematics bicycle model of a car.

***Definition** 3: (The Buffered Input Cell)* For a robot with linear dynamics (5), position model (6) and invertible J_i, the BIC for the robot at time step t is defined as

$$BIC_i^t = J_i^{-1}(\bar{\mathcal{V}}_i^t \oplus (-C_i F_i \mathbf{x}_i^t)), \tag{8}$$

where $\bar{\mathcal{V}}_i^t$ is the robot's BVC.

Intuitively, the BIC represents the set of one-step control inputs that will result in a robot position inside the *current* BVC at the *next* time step. An example of a BIC is illustrated in Fig. 2.

***Lemma** 2: (One Step Collision Avoidance)* Under Assumption 1, if the robots are collision free at time t, and if each robot's control input lies inside its non-empty BIC at time t, $\mathbf{u}_i^t \in BIC_i^t$, the robots will remain collision free at time $t+1$.

Proof: Firstly, if the robots are collision free at time t, their BVCs at time t are non-empty (see Lemma 1). Therefore by (8), their BICs are also non-empty at time t. Now, from (7), if $u_i^t \in BIC_i^t$, then $p_i^{t+1} \in C_i F_i \mathbf{x}_i^t + C_i G_i J_i^{-1}(\bar{\mathcal{V}}_i^t - C_i F_i \mathbf{x}_i^t) = \bar{\mathcal{V}}_i^t$, meaning the position of each robot at $t+1$ is in its BVC at time t. Therefore, by the definition of the BVC, they remain in a collision-free configuration at time $t+1$. ■

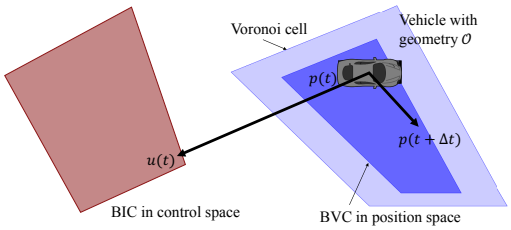

Fig. 2. Example of BVC constraints (in blue) in position space and the corresponding BIC (in red) constraints in control space. Note that the shape is not necessarily preserved during the mapping.

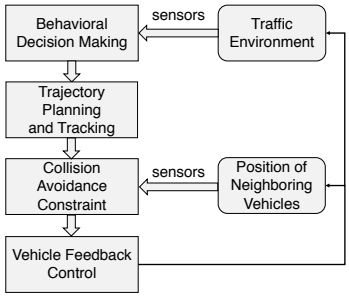

Fig. 3. Architecture for autonomous freeway driving using BICs. The information flow and various layers of control are visualized.

An induction argument on Lemma 2 leads to the main result as follows.

***Theorem** 1: (BIC Collision Avoidance)* Under Assumption 1, if all robots begin in a collision free configuration at $t = 0$, and if their control inputs lie in their BVCs at all time steps, $\mathbf{u}_i^t \in BIC_i^t \ \forall t \geq 0$ and $\forall i$, collisions will be avoided for all time.

Proof: The base case for the induction is that the robots are collision free at $t = 0$. Lemma 2 provides the induction step. This completes the proof. ■

***Remark** 1: (Reciprocity and Robustness)* Theorem 1 requires reciprocity in the sense that all robots obey their own BIC constraint, although they may have entirely different control policies. However, we observe in simulation that even if some vehicles violate this reciprocity assumption (e.g. human driven vehicles), collision can still be avoided. Simulation results in Sec. IV support this assertion.

III. Autonomous Lane Change Control with BICs

In this section, we apply the BIC approach we introduced in the previous section to autonomous freeway driving. We use the BIC constraint to ensure that multiple autonomous vehicles do not collide during simultaneous lane change maneuvers. We implement a control stack for autonomous lane change maneuvers with the BIC constraint to avoid collision, along with other high-level decision making tools, trajectory planning, and differential flatness theory.

A. Overall Architecture

The architecture of the proposed lane change method is illustrated in Fig. 3. As shown in the figure, a lane change maneuver comprises four main steps:

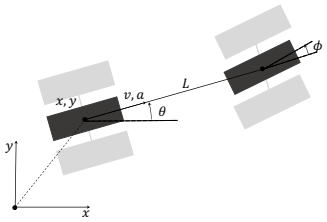

Fig. 4. Illustration of the bicycle model (in dark grey) for a four-wheel vehicle (in light grey). Both wheels in the bicycle model can rotate to move the car in the forward direction. The front wheel can also rotate about its vertical axis, giving a non-holonomic rolling constraint.

1) Behavioral decision making. Based on a map, route and sensor information, this layer selects an appropriate high-level behavior (such as lane change).
2) Trajectory planning and tracking. A reference trajectory and control input can be obtained using differential flatness.
3) Buffered Input Cell constraint. The BIC layer takes the reference control input and constrains it to be inside the current BIC to ensure safety. If the commanded control input exceeds the admissible BIC region, a nearest point on the BIC is chosen.
4) Vehicle Feedback Control. Low-level, closed-loop control to ensure robustness and stability.

B. Behavioral Decision Making Layer

For the determination of high-level commands, we adopt the Multi-Policy Decision Making (MPDM) method proposed in [24]. The MPDM method constantly evaluates and selects the best policy for the commanded vehicle from a set of closed-loop policies. The consequences of the candidate closed-loop policies are predicted by conducting forward simulations. We use the longitudinal distance change during the forward simulation as the score J, i.e.,

$$J = \int_{t_0}^{t_h} \dot{x} dt = x(t_h) - x(t_0),$$

where t_0 and t_h are the start and end time.

C. Vehicle Dynamics Model and Differential Flatness

We use the bicycle model [25], as illustrated in Fig. 4, to characterize the kinematics of the autonomous vehicle. We start with the continuous time model, and then derive a linearized discrete time model to incorporate the BIC constraint. We suppress the vehicle index i in this section to simplify notation. Vehicle states at time t are represented by $\mathbf{x}(t) = [x(t), y(t), \theta(t), v(t)]^T \in \mathbb{R}^4$, describing the position of the rear axle center point $(x(t), y(t))$, the vehicle heading $\theta(t)$, and the magnitude of the velocity $v(t)$. Control inputs of the vehicle are acceleration and steering angle represented by $\mathbf{u}(t) = [a(t), \phi(t)]$. The vehicle kinematics is expressed as

$$\dot{x}(t) = v(t)\cos\theta(t), \qquad \dot{y}(t) = v(t)\sin\theta(t) \tag{9a}$$

$$\dot{\theta}(t) = \frac{v(t)}{L}\tan\phi(t), \qquad \dot{v}(t) = a(t), \tag{9b}$$

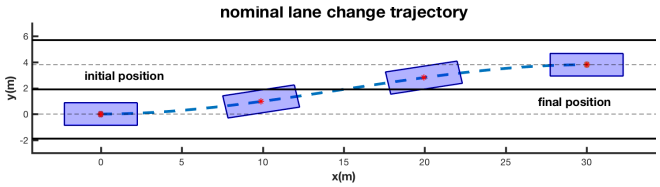

Fig. 5. An example of nominal trajectory during lane change maneuvers. The trajectory is parameterized by time polynomials from the start position (bottom left) to the final position (top right).

where L is the vehicle length from front axle center point to rear axle center point.

Note that it is nontrivial to specify a feasible reference trajectory for the vehicle with nonholonomic constraints (9). Therefore, we employ differential flatness [26] to facilitate the design of a reference trajectory that can always satisfy the differential kinematic constraints. Here we show that the bicycle model is differentially flat and also find the formula from the flat output to the nominal trajectory inputs.

Briefly, a nonlinear system $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ is differentially flat if there exists a smooth function α such that

$$\sigma = \alpha(\mathbf{x}, \mathbf{u}, \dots, \mathbf{u}^{(p)}),$$

for some finite number of time derivatives p, where σ is called the flat output. We can also write the solution of the nonlinear system as a function of σ and its time derivatives

$$\mathbf{x} = \beta(\sigma, \dot{\sigma}, \dots, \sigma^{(p)}), \quad \mathbf{u} = \gamma(\sigma, \dot{\sigma}, \dots, \sigma^{(p)}),$$

where the two functions β, γ are are smooth and are called the endogenous transformation.

The bicycle model in (9) can be shown to be differentially flat with the flat output $\sigma = [\sigma_1\ \sigma_2]^T = [x\ y]^T$ and the endogenous output transformation β, γ are

$$x = \sigma_1, \qquad y = \sigma_2 \tag{10a}$$

$$\theta = \arctan 2(\dot{\sigma_2}, \dot{\sigma_1}), \qquad v = \sqrt{\dot{\sigma_1}^2 + \dot{\sigma_2}^2} \tag{10b}$$

$$a = \frac{\dot{\sigma_1}\ddot{\sigma_1} + \dot{\sigma_2}\ddot{\sigma_2}}{\sqrt{\dot{\sigma_1}^2 + \dot{\sigma_2}^2}}, \qquad \phi = \arctan(\frac{\dot{\sigma_1}\ddot{\sigma_2} - \dot{\sigma_2}\ddot{\sigma_1}}{(\dot{\sigma_1}^2 + \dot{\sigma_2}^2)^{\frac{3}{2}}} l). \tag{10c}$$

Using differential flatness, we can specify a low-dimensional position trajectory $\sigma(t)$ containing only $x(t)$ and $y(t)$, and generate the entire state trajectory $(x(t), y(t), \theta(t), v(t))$ as well as the open-loop control input trajectories $a(t)$ and $\phi(t)$. Applying the input trajectory $(a(t), \phi(t))$ to the nonlinear model without noise or any modeling errors will drive the vehicle's position along the planned trajectory.

D. Trajectory Generation and Tracking

We generate a trajectory for lane changing using polynomial splines in time, similarly to [27]. Consider the scenario where the vehicle decides to change lane from state $\mathbf{x}(0) = [x(0), y(0), \theta(0), v(0)]^T$ to terminal state $\mathbf{x}(T) = [x(T), y(T), \theta(T), v(T)]^T$. These constraints can be imposed in the trajectory by specifying the initial and terminal conditions on σ_1, σ_2, $\dot{\sigma}_1$ and $\dot{\sigma}_2$. A polynomial trajectory can

thus be expressed in the form

$$\sigma(t) = \begin{bmatrix}\sigma_1(t)\\ \sigma_2(t)\end{bmatrix} = \sum_{i=0}^{3} \alpha_i t^i, \tag{11}$$

with constraints

$$\sigma(0) = [x(0)\ y(0)]^T,\ \dot{\sigma}(0) = [\dot{x}(0)\ \dot{y}(0)]^T \tag{12a}$$
$$\sigma(T) = [x(T)\ y(T)]^T,\ \dot{\sigma}(T) = [\dot{x}(T)\ \dot{y}(T)]^T, \tag{12b}$$

where $\alpha_i \in \mathbb{R}^{2\times 1}$. With the eight constraints above, we can obtain a unique solution for the eight unknown parameters α_i, $i = 0, \ldots, 3$ in (11). Fig. 5 shows an example of a generated trajectory. The control inputs and reference trajectory are then used in a feedforward-feedback controller that drives the vehicle along the trajectory despite disturbances and modeling errors.

E. BIC Collision Avoidance Layer

The BIC requires a discrete time, linear dynamical model for the car. Therefore, we linearize the vehicle model at each time step, and discretize in time by computing the matrix exponentials of the matrices from the continuous time model. Specifically, we linearize the model (9) about the current operating state $(x_t, y_t, \theta_t, v_t)$, with $\theta_t = 0$ and $\phi_t = 0$. At freeway speeds heading and steering angle are typically close to zero, so we expect this linearized model to closely match the nonlinear model. The BIC also assumes that the robot orientation does not change during the process, which is approximately true with the small angle assumption. The resulting linearized equations are:

$$\begin{aligned}
\dot{x} &= v_t \cos\theta_t + (v - v_t)\cos\theta_t - v_t(\theta - \theta_t)\sin\theta_t = v,\\
\dot{y} &= v_t \sin\theta_t + (v - v_t)\sin\theta_t + v_t(\theta - \theta_t)\cos\theta_t = v_t\theta\\
\dot{\theta} &= \frac{1}{L}\Big(\frac{v_t(\phi - \phi_t)}{\cos^2\phi_t} + (v - v_t)\tan\phi_t + v_t\tan\phi_t\Big) = \frac{v_t\phi}{L}\\
\dot{v} &= a
\end{aligned}$$

Taking the second-derivative of $x(t)$ and $y(t)$ yields the simple linearized model,

$$\ddot{x} = a, \quad \ddot{y} = \tfrac{{v_t}^2}{L}\phi. \tag{14}$$

Note that the linearized bicycle model is decoupled in x and y directions under this linearization. The linearized model can be expressed in state space form $\dot{\mathbf{x}} = A_t\mathbf{x} + B_t\mathbf{u}$ and $\mathbf{p} = C_t\mathbf{x}$, with matrices

$$A_t = \begin{bmatrix}0&0&0&1\\0&0&v_t&0\\0&0&0&0\\0&0&0&0\end{bmatrix}, B_t = \begin{bmatrix}0&0\\0&0\\0&\frac{v_t}{L}\\1&0\end{bmatrix}, C = \begin{bmatrix}1&0&0&0\\0&1&0&0\end{bmatrix}.$$

To discretize the model in time, we assume constant control inputs in a small time interval, i.e., $u(t) = [a_t, \phi_t]$ from time t to $t + \Delta t$. Given initial state $\mathbf{x}(t) = \mathbf{x}_t$ at time t, we solve $\mathbf{x}(t + \Delta t) = F_t\mathbf{x}_t + G_t\mathbf{u}_t$, where $F_t = e^{A_t\Delta t}$ and $G_t = \int_0^{\Delta t} e^{A_t\tau}B_t d\tau$.

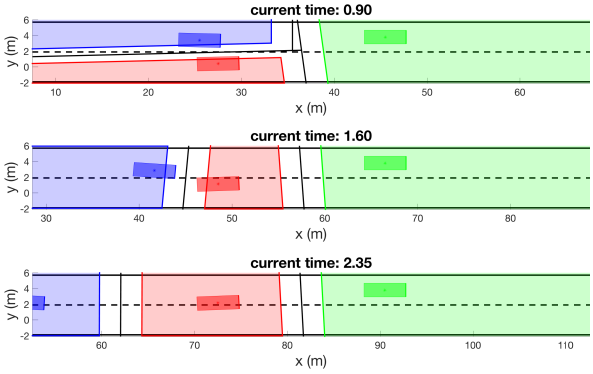

Fig. 6. Three snapshots during simulation (Section IV-A). Vehicles start lane changing in the top figure. Red vehicle enters the target lane in the middle figure, and then finishes lane changing in the bottom figure. No collision occurred during the simulation.

$$F_t = \begin{bmatrix}1&0&0&\Delta t\\0&1&\Delta t v_t&0\\0&0&1&0\\0&0&0&1\end{bmatrix} \text{ and } G_t = \begin{bmatrix}\frac{\Delta t^2}{2}&0\\0&\frac{(\Delta t v_t)^2}{2L}\\0&\frac{\Delta t v_t}{L}\\\Delta t&0\end{bmatrix}. \tag{15}$$

We therefore obtain the vehicle's position as a function of the discrete time state and control input as

$$p_{t+\Delta t} = CF_t\mathbf{x}_t + CG_t\mathbf{u}_t \tag{16}$$

Recall, to apply the BIC, we require from Assumption 1 that the matrix

$$J_t = CG_t = \begin{bmatrix}\frac{\Delta t^2}{2}&0\\0&\frac{(\Delta t v_t)^2}{2L}\end{bmatrix} \tag{17}$$

be invertible. For $v_t > 0$, the matrix J_t is indeed invertible for this linearized vehicle model, and the case where $v_t = 0$ is irrelevant to our high speed freeway driving scenario. With these definitions for F_t, G_t, C and J_t updated by each vehicle at each time step, we project the inputs from the feedback controller into the BIC for collision avoidance, as described in Sec. II.

IV. Simulation

In this section, simulations are conducted to evaluate the performance of our algorithm in a freeway lane change scenario. The first simulation has multiple autonomous vehicles performing conflicting lane changes simultaneously using our BIC algorithm, whereas the second one includes a vehicle controlled by an adversarial human driver in order to test the robustness of our approach. In the third simulation, we assessed and compared our method to MPC method, which is a major field of research on its applications to autonomous vehicles. The autonomous vehicles use the linearized and discretized dynamics for their own BIC computation, however their motion in the simulation is determined by their full nonlinear dynamics, again to verify the robustness of the approach.

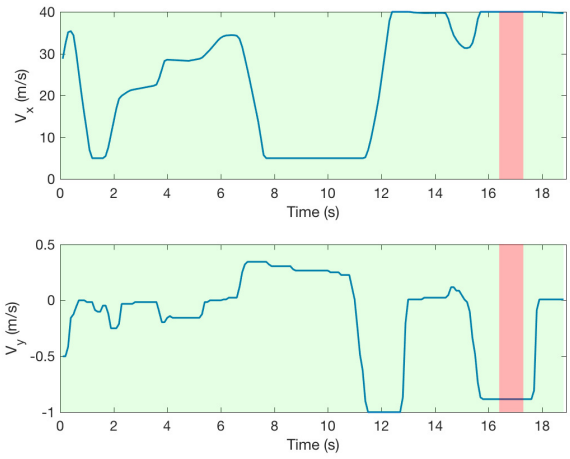

Fig. 7. Velocities in x and y directions of the human-driven vehicle. Green regions represent the time durations when human-operated vehicle was inside its BVC , while red region represents the human-operated vehicle was outside its BVC.

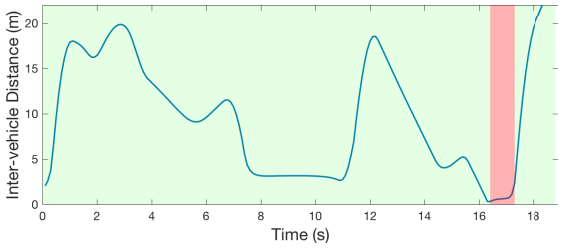

Fig. 8. Distance between the human controlled vehicle and BIC controlled vehicle during the simulation (in blue line). The inter-vehicle distance was always above 0, indicating that no collision occurred.

A. Two Vehicles Changing Lanes Simultaneously

Consider a common maneuver in freeway driving as visualized in Figure 1, where two adjacent vehicles decide to change lanes simultaneously and their planned trajectories intersect. Three vehicles are driving in the same direction (from left to right) on a two-lane, one-way highway. The initial configurations of all vehicles are shown in Fig. 1. The green vehicle travels at a constant and relatively low speed while the red and blue vehicles are trying to change lanes starting from the same time using our proposed lane change maneuver control architecture. The lane change maneuver time is set to $Tc = 4.0s$ for blue vehicle and $Tc = 3.9s$ for the red vehicle. Initial and target speeds for the three vehicles are $v_{\text{blue}} = 35m/s, v_{\text{red}} = 30m/s, v_{\text{green}} = 35m/s$. Three snapshots are shown in Fig. 6 during the simulation at time $0.90s$, $1.60s$ and $2.35s$. In the top figure in Fig. 6, the blue vehicle slowed down thus the red vehicle continues to pass the blue vehicle and change lane. Simulation results show that our algorithm successfully navigates the two vehicles to change lanes without collision.

B. Interaction Between an Autonomous Vehicle and a Human Driver

In this simulation, the blue vehicle is controlled by a human through a joystick. Inputs from the joystick are accelerations in x and y directions (hence the human vehicle is a double-integrator), while the autonomous cars are the same as in the previous simulation. Although the human driver does not necessarily obey the BIC constraint and sometimes exhibits adversarial behavior during the simulation, the autonomous vehicles are still able to avoid collision. The velocity profile of the human controlled vehicle is shown in Fig. 7. The smallest distance among the vehicles is plotted in Fig. 8, which also illustrates if the human driver obeys the BIC constraints. It can be seen from the plot that around 16s to 18s, the human control input violates the BIC constraint, resulting in an instance of small clearance between the vehicles. However, no collision occurred during the simulation since the autonomous vehicle dodged promptly. Also from this simulation, although the vehicles have different dynamics models, this does not change the effectiveness of the BIC algorithm, which allows for collision avoidance for heterogenous robot groups (recall that each vehicle only needs to know its own dynamics, and the dynamics need not be the same among all vehicles).

C. Performance Comparison with MPC

The proposed BVC algorithm is compared to model predictive control (MPC) method concerning the collision avoidance performance. MPC method solves the trajectory planning problem for a short horizon and executes only the control input for the first time step. In the next time step, a new optimal control problem is solved. Our MPC method is implemented in MATLAB with CVX [28] using MOSEK solver. Collision avoidance constraints is imposed using Mixed Integer Programming. We still consider the scenario where the blue vehicle and the red vehicle are changing lanes simultaneously as in Sec. IV-A. The objective function of each vehicle is expressed as

$$\min_{\mathbf{u}} \sum_{i=1}^{N} \lambda^i (\mathbf{p}_i - \mathbf{p}_{\text{target}})^T R (\mathbf{p}_i - \mathbf{p}_{\text{target}}) + \gamma^i {\mathbf{u}_i}^T Q \mathbf{u}_i,$$

where N is the planning steps, $\mathbf{u} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, u_N]$ is control input, and $\mathbf{p}_i$ is vehicle position at step i. Vehicle model given in Eqn. 9 is linearized at every step.

In the first case, both blue and red vehicles were using the MPC controller. Since there was no communication between vehicles, a non-cooperative and oscillation behavior was shown during the simultaneous lane change maneuvers which led to a collision as shown in Fig. 9. In the second case (Fig. 10), the red vehicle is using MPC controller while the blue vehicle is using BIC controller. The BIC control is able to avoid collision, even though the other vehicle is using the MPC controller. The performance comparison between BIC and MPC controllers shows the benefit of our proposed method. The BVC controller is more reactive and robust to vehicles whose intention and thus future trajectory is uncertain for the ego vehicle, which is essential to autonomous driving scenarios.

V. Conclusion and Future Work

In this paper, we proposed a computationally efficient, distributed control algorithm achieving collision free lane

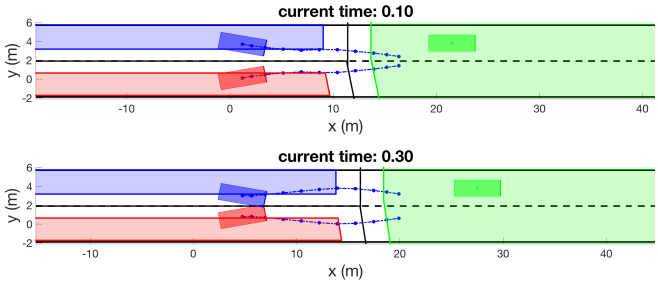

Fig. 9. Two snapshots during the simulation. Both red and blue vehicles are using MPC controller. The blue dash-dot lines are the MPC planned trajectories.

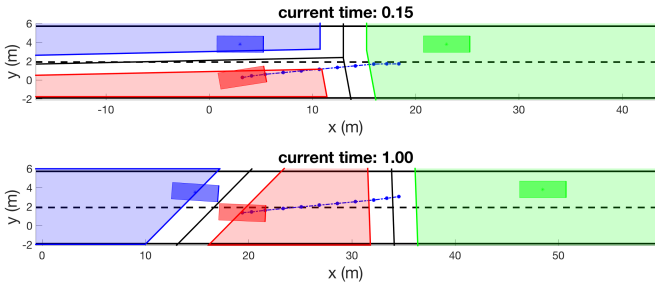

Fig. 10. Two snapshots during the simulation with the same initial configuration as in Fig. 9. The red vehicle was using MPC controller while the blue vehicles was using BIC controller. The blue dash-dot line is the MPC planned trajectory.

change maneuvers for autonomous vehicles, which only relies on relative position measurements. We introduced Buffered Input Cells, which restrict the command control inputs to the allowable control inputs at each time step to guarantee collision avoidance among multiple vehicles. The performance of this algorithm is demonstrated in MATLAB simulations. There are many directions for future work. We are currently formulating a probabilistic version of the BVC and BIC to account for significant sensor noise. We also plan to consider the effects of actuator saturation on the collision avoidance guarantees of the BIC method. We also plan to perform hardware experiments with model-scale autonomous cars and full-sized autonomous vehicles.

References

[1] D. Zhou, Z. Wang, S. Bandyopadhyay, and M. Schwager, "Fast, online collision avoidance for dynamic vehicles using buffered voronoi cells," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1047–1054, 2017.

[2] O. Arslan and D. E. Koditschek, "Sensor-based reactive navigation in unknown convex sphere worlds," in *The 12th International Workshop on the Algorithmic Foundations of Robotics*, 2016.

[3] NHTSA report, *https://static.nhtsa.gov/odi/inv/2016/INCLA-PE16007-7876.PDF*.

[4] B. Schoettle and M. Sivak, "A preliminary analysis of real-world crashes involving self-driving vehicles," *University of Michigan Transportation Research Institute*, 2015.

[5] L. Fletcher, S. Teller, E. Olson, D. Moore, Y. Kuwata, J. How, J. Leonard, I. Miller, M. Campbell, D. Huttenlocher *et al.*, "The mit–cornell collision and why it happened," *Journal of Field Robotics*, vol. 25, no. 10, pp. 775–807, 2008.

[6] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, "Autonomous driving in urban environments: approaches, lessons and challenges," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4649–4672, 2010.

[7] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for berthaa local, continuous method," in *Intelligent Vehicles Symposium Proceedings*, 2014, pp. 450–457.

[8] D. Reichardt and J. Shick, "Collision avoidance in dynamic environments applied to autonomous vehicle guidance on the motorway," in *Intelligent Vehicles Symposium, Proceedings of the*, 1994, pp. 74–78.

[9] O. Arslan, K. Berntorp, and P. Tsiotras, "Sampling-based algorithms for optimal motion planning using closed-loop prediction," *arXiv preprint arXiv:1601.06326*, 2016.

[10] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105–1118, 2009.

[11] G. Schildbach and F. Borrelli, "Scenario model predictive control for lane change assistance on highways," in *Intelligent Vehicles Symposium (IV)*, 2015, pp. 611–616.

[12] T. Gu, J. Atwood, C. Dong, J. M. Dolan, and J.-W. Lee, "Tunable and stable real-time trajectory planning for urban autonomous driving," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, 2015, pp. 250–256.

[13] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[14] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Robotics and Automation (ICRA), IEEE International Conference on*, 2008, pp. 1928–1935.

[15] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*. Springer, 2011, pp. 3–19.

[16] D. Bareiss and J. van den Berg, "Generalized reciprocal collision avoidance," *The International Journal of Robotics Research*, vol. 34, no. 12, pp. 1501–1514, 2015.

[17] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, "Control barrier certificates for safe swarm behavior," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 68–73, 2015.

[18] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.

[19] Z. N. Sunberg, C. Ho, and M. J. Kochenderfer, "The value of inferring the internal state of traffic participants for autonomous freeway driving," in *American Control Conference*, 2017, pp. 3004–3010.

[20] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, "Intention-aware online pomdp planning for autonomous driving in a crowd," in *Robotics and Automation (ICRA), International Conference on*, 2015, pp. 454–460.

[21] H. Hadwiger, "Minkowskische addition und subtraktion beliebiger punktmengen und die theoreme von erhard schmidt," *Mathematische Zeitschrift*, vol. 53, no. 3, pp. 210–218, 1950.

[22] J.-C. Latombe, *Robot motion planning*. Springer Science & Business Media, 2012, vol. 124.

[23] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[24] A. G. Cunningham, E. Galceran, R. M. Eustice, and E. Olson, "MPDM: Multipolicy decision-making in dynamic, uncertain environments for autonomous driving," in *Robotics and Automation (ICRA), IEEE International Conference on*, 2015, pp. 1670–1677.

[25] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[26] M. Van Nieuwstadt, M. Rathinam, and R. Murray, "Differential flatness and absolute equivalence of nonlinear control systems," *SIAM Journal on Control and Optimization*, vol. 36, no. 4, pp. 1225–1239, 1998.

[27] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Robotics and Automation (ICRA), International Conference on*, 2011, pp. 2520–2525.

[28] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.