Distributed motion coordination using convex feasible set algorithm

Hongyu Zhou[1] and Changliu Liu[2]

Abstract— **This paper presents a distributed approach for multi-vehicle motion coordination with application to autonomous driving based on model predictive control (MPC) and convex feasible set (CFS) algorithm. Starting from a centralized approach of the motion coordination problem, we exploit the problem structure and vehicle to vehicle (V2V) communication to decompose the centralized problem. By using CFS to convexify the collision avoidance constraints, the collision-free trajectories can be computed in real time. We analyze the deadlock situation and show that a deadlock can be avoided by changing vehicles' desired speed. Finally, numerical simulations are conducted to verify the proposed distributed method and show the computation efficiency compared to a centralized method. The proposed approach is robust to the tracking errors, which can decrease optimality of the planned trajectories. The results show that our distributed design can lead to safe multi-vehicle coordination.**

I. INTRODUCTION

The research on autonomous vehicles is gaining increasing attention in the robotics community. It is a promising area because of its potentials to improve mobility and bring benefits to our lives, such as reducing crashes, congestion, and emissions. To exploit these benefits, safe and efficient methods to control and coordinate connected autonomous vehicles are important. Each vehicle interacts with its surrounding vehicles and it must avoid collisions for safety concerns. Hence, a key challenge is to efficiently generate a safe motion coordination strategy for a multi-vehicle system.

Expanded from the results of a single-agent system, classical methods for multi-agent coordination include the cell decomposition (e.g., [1], [2]), potential field (e.g., [3], [4]), and roadmap approaches (e.g., [5], [6]). These methods reduce the continuous motion planning problem to a discrete graph search problem and their computational complexity depends on the discretization resolution [7]. Their drawback is that discretization usually results in non-smooth trajectories and suboptimal solutions.

Optimization-based methods which plan in continuous space can generate smoother trajectory. In addition, it is able to take into account the interaction among robots by formulating constraints properly. However, since the optimization problem is usually nonlinear and non-convex, these methods are not computationally efficient for real time implementation. A popular method is the sequential quadratic programming (SQP) [8], [9], [10], which transforms the non-convex optimization problem into a sequence of QP problems and solves them iteratively. However, SQP is not fast enough for real time applications.

Convex feasible set (CFS) algorithm is proposed for real time motion planning [11], [12], [13]. It handles optimization problems with convex objective function and non-convex equality (e.g. robot dynamics) and inequality (e.g. collision avoidance) constraints. Like SQP, CFS algorithm also approximates the original problem as a sequence of convex sub-problems. However, by exploiting the unique geometric structure of the motion planning problems, CFS algorithm is an order of magnitude faster than SQP. Based on CFS algorithm, a fast optimization-based autonomous driving motion planner (FOAD) in [14] is proposed. A centralized planner called multi-car convex feasible set (MCCFS) algorithm is developed for multi-vehicle motion coordination in [15]. To avoid inter-vehicle collision, MCCFS algorithm formulates not only distance constraints, but also priority constraints which prevents two vehicles crossing between two consecutive time steps.

The motion coordination problem can be solved by either a centralized or a distributed approach. The centralized approach requires a central control agent that has information about the workspace and all agents, and that is able to communication with all agents and share the information. However, this approach is ineffectual for a large number of agents and it strongly relies on the central control station which can produce a vulnerable system [7]. On the contrary, all agents are independent and equal with respect to control, and there is no central control agent in the distributed approach, at the cost of a more complex structure and organization. Compared to the centralized approach, the distributed approach is more reliable, adaptable and robust in the presence of many physical constraints, such as limited resources and energy, short wireless communication ranges, and narrow bandwidths [16].

There are numerous distributed approaches to motion coordination. A scheduling design is proposed for intersection management in [17]. This method includes a high-level decision maker for passing order based on conflict zones and a low-level speed profile planner. [18] propose a reactive strategy called reciprocal velocity obstacles (RVO) for multi-agent coordination. Distributed MPC designs have been proposed in, e.g., [19], [20], [21], [22]. [19] formulates collision avoidance as a dual optimization problem and proposes a bi-level distributed MPC scheme. [20] uses invariant-set theory and mix-integer linear programming (MILP) to formulate the distributed motion coordination problem. [21] and [22] rely

[1]Hongyu Zhou is with the Department of Marine Technology, Noweigian University of Science and Technology, NO-7491 Trondheim, Norway hongyuz@alumni.ntnu.no

[2]Changliu Liu is with the The Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA, 15213, USA cliu6@andrew.cmu.edu

on the alternating direction method of multipliers (ADMM) to decompose the coordination problem. [21] uses separating hyperplanes and [22] formulates non-convex constraints for collision avoidance.

In this paper, we focus on efficient, safe and coordinated multi-vehicle motion planning with application to autonomous driving. We exploit the structure of the coordination problem to formulate the distributed approach, with the assumption that vehicle to vehicle (V2V) communication is available. Our method does not rely on MILP or ADMM which can be computationally expensive. We do not formulate conflict zones either. Instead, we represent vehicles by convex geometric shape and leverage the CFS algorithm to convexify the collision avoidance constraints. This also avoids solving an additional optimization problem due to collision avoidance as in [19]. We analyze the features of the deadlock situation and propose a solution by changing vehicles' desired speed. We validate our method by simulation, comparing its performance with a centralized implementation, a scheduling design, and RVO.

The rest of the paper is organized as follows. Section II presents the CFS algorithm and multi-vehicle motion coordination problem. Section III formulates the distributed motion planning problem. Section IV presents the numerical results. Section V concludes the paper with directions for future work.

II. PRELIMINARIES

A. Convex Feasible Set (CFS) Algorithm

The CFS algorithm is proposed in [13] to solve the following non convex optimization problem iteratively:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\Gamma} J(\mathbf{x}), \tag{1}$$

where $\mathbf{x}$ is the state variable, $J(\mathbf{x})$ is a smooth and convex objective function, Γ is the state space constraint which can be non-convex, and $\mathbf{x}^*$ is the optimal solution.

To solve the optimization problem (1) using CFS algorithm, there are three steps:

1) Step 1: Initialize the state variable $\mathbf{x}^{(0)}$.
2) Step 2: Using the state variable of the last iteration $\mathbf{x}^{(k)}$, calculate the convex feasible set $\mathcal{F}(\mathbf{x}^{(k)}) \subset \Gamma$. For simplicity, we assume that Γ is the intersection of collision avoidance constraints, i.e., $\Gamma = \cap_i \Gamma_i$, where $\Gamma_i = \{\mathbf{x} : \phi_i \geq 0\}$ is the space outside of the ith obstacle, and ϕ_i is the signed distance function to the ith obstacle.
   In the cases where obstacles' shape can be considered convex [12], the convex feasible set corresponding to the ith obstacle is

$$\mathcal{F}_i(\mathbf{x}^{(k)}) = \{\mathbf{x} : \phi_i(\mathbf{x}^{(k)}) + \nabla\phi_i(\mathbf{x}^{(k)})(\mathbf{x}-\mathbf{x}^{(k)}) \geq 0\}, \tag{2}$$

   where ∇ is the gradient operator. At a point where ϕ_i is not differentiable, $\nabla\phi_i$ is a sub-gradient such that the convex feasible set $\mathcal{F}$ always includes the steepest descent direction of $J(\mathbf{x})$ in the set Γ [12].
3) Step 3: Given the convex feasible set $\mathcal{F}(\mathbf{x}^{(k)})$, obtain new solution $\mathbf{x}^{(k)}$ by solving

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}\in\mathcal{F}(\mathbf{x}^{(k)})} J(\mathbf{x}). \tag{3}$$

The CFS algorithm starts with step 1, then applies steps 2 and 3 iteratively. It is proved that the sequence $\mathbf{x}^{(k)}$ generated by step 3 will converge to a local optimum $\mathbf{x}^*$ of the optimization problem (1).

B. Multi-vehicle Motion Coordination

The multi-vehicle motion coordination can be formulated as a centralized MPC which computes the collision-free trajectories for all vehicles simultaneously. In this subsection, the centralized MPC is introduced.

1) Configuration Space: In the autonomous driving application, the configuration is the 2D position of a vehicle, denoted as $x \in \mathbb{R}^2$. Then the trajectory of the ith vehicle can be denoted as $\mathbf{x}_i = \left[x_i^1; x_i^2; ...; x_i^H\right] \in \mathbb{R}^{2H}$, where H is the planning horizon, $i \in \mathcal{V}$ is the vehicle index, and $\mathcal{V} := \{1, 2, ..., N\}$ is the set of N vehicles.

2) Objective Function: We formulate the objective function $J_i(\mathbf{x}_i, s_i)$ for the ith vehicle to be quadratic. It is defined as $J_i(\mathbf{x}_i, s_i) = J_i^o(\mathbf{x}_i) + J_i^a(\mathbf{x}_i) + J_i^s(s_i)$. The three terms are explained as follows.

- $J_i^o(\mathbf{x}_i)$ penalizes the difference between the planning trajectory $\mathbf{x}_i$ and the reference trajectory $\mathbf{x}_i^{ref} \in \mathbb{R}^{2H}$, which is given by

$$\begin{aligned} J_i^o(\mathbf{x}_i) &= \frac{1}{2} c_o (\mathbf{x}_i - \mathbf{x}_i^{ref})^\top (\mathbf{x}_i - \mathbf{x}_i^{ref}) \\ &= \frac{1}{2} c_o \mathbf{x}_i^\top \mathbf{x}_i - c_o \mathbf{x}_i^\top \mathbf{x}_i^{ref} + \frac{1}{2} c_o (\mathbf{x}_i^{ref})^\top \mathbf{x}_i^{ref}, \end{aligned} \tag{4}$$

  where c_o is a weighting parameter. Note that since $\frac{1}{2} c_o (\mathbf{x}_i^{ref})^\top \mathbf{x}_i^{ref}$ is constant, this term can be neglected in the objective function.
- $J_i^a(\mathbf{x}_i)$ penalizes the magnitude of lateral and longitudinal accelerations, which is defined as

$$\begin{aligned} J_i^a(\mathbf{x}_i) &= \frac{1}{2} c_a (\mathbf{A}_i \mathbf{x}_i)^\top (\mathbf{A}_i \mathbf{x}_i) \\ &= \frac{1}{2} c_a \mathbf{x}_i^\top \mathbf{A}_i^\top \mathbf{A}_i \mathbf{x}_i, \end{aligned} \tag{5}$$

  where c_a is a weighting parameter and $\mathbf{A}_i$ is the acceleration matrix which calculates the acceleration of the planning trajectory $\mathbf{x}_i$:

$$\mathbf{A}_i = \frac{1}{T_s^2} \begin{bmatrix} I & -2I & I & 0 & \cdots & 0 \\ 0 & I & -2I & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & I & -2I & I \end{bmatrix}, \tag{6}$$

  and T_s is the sampling time.
- $J_i^s(s_i) = c_s \|s_i\|^2$ penalizes the magnitude of the slack variable $s_i \in \mathbb{R}^2$, where c_s is a weighting parameter.

3) Constraints: In this paper, the collision avoidance constraints force each vehicle pair (i, j), where $i, j \in \mathcal{V}$ and $i \neq j$, to maintain safety distance at each time step:

$$\phi(x_i^h, x_j^h) = d(x_i^h, x_j^h) - d \geq 0, \tag{7}$$

where $h \in \{1, 2, ..., H\}$, $d(x_i^h, x_j^h)$ is the distance between x_i^h and x_j^h, and d is the safety margin between two vehicles, which can take into account the model uncertainty and measurement errors.

Besides, we want the planned trajectories start from vehicles' current positions, i.e., $x_i^1 = x_i^c$. However, when tracking errors exist, this constraint can give an infeasible initial condition. Hence, we modify this constraint as

$$x_i^1 = x_i^c + s_i, \tag{8}$$

where x_i^c represents the current position of the ith vehicle and s_i is a slack variable.

The resulting optimization problem is summarized as

$$\min_{\mathbf{x}_i, s_i} \quad \sum_{i=1}^{N} J_i(\mathbf{x}_i, s_i) \tag{9a}$$

$$\text{s.t.} \quad \phi(x_i^h, x_j^h) = d(x_i^h, x_j^h) - d \geq 0, \tag{9b}$$

$$x_i^1 = x_i^c + s_i, \tag{9c}$$

$$\forall i \in \mathcal{V}, j \in \mathcal{V}\backslash\{i\}, h \in \{1, 2, ..., H\}.$$

III. DISTRIBUTED MOTION COORDINATION

The optimization problem (9) calculates the optimal trajectories for all vehicles simultaneously. The number of decision variables $\mathbf{x}_i$, for all $i \in \mathcal{V}$, is proportional to the number of vehicles, which makes the centralized MPC not scalable to a larger team. In this section, we present the distributed approach which allows each vehicle to solve a sub-problem in parallel. Based on CFS algorithm, the non-convex sub-problem is transformed into a QP problem, which can be solved efficiently. We analyze the situations where a deadlock happens and propose a solution.

The overall algorithm for vehicle i is presented in Algorithm 1. The trajectory is replanned every T_r. First, vehicle i communicate with all surrounding vehicles (line 3). Then vehicle i detects if a deadlock occurs, and if so, it changes its desired speed (lines 4-8). If the deadlock is avoided and vehicle i reaches its reference trajectory, then it is allowed to move at the original speed (lines 9-13). Finally, $\mathbf{x}_i^{ref}$ should be modified according to its current position x_i^c and desired speed, and the optimal trajectory for vehicle i can be obtained by solving a sub-problem (lines 15-17).

Algorithm 1: Distributed MPC for vehicle i

Input: x_i^c, $\mathbf{x}_j$, $\forall j \in \mathcal{V}\backslash\{i\}$
Parameter: c_o, c_a, c_s, T_r, T_s, H, l, w, r, n, ϵ_1, ϵ_2
Output: $\mathbf{x}_i$

1 Initialize $\mathbf{x}_i$, $\mathbf{x}_i^{ref}$, $Deadlock\_Flag \leftarrow False$;
2 **for** $t = 0, T_r, 2T_r, ..., \infty$ **do**
3 Communication with vehicle j, $\forall j \in \mathcal{V}\backslash\{i\}$: send $\mathbf{x}_i$ and receive $\mathbf{x}_j$;
4 **if** $Deadlock\_Flag$ *is* $False$ **then**
5 **if** *Equation* (14) *is satisfied* **then**
6 $Deadlock\_Flag \leftarrow True$;
7 Change desired speed;
8 **end**
9 **else**
10 **if** *Vehicle* i *converges to* $\mathbf{x}_i^{ref}$ **then**
11 $Deadlock\_Flag \leftarrow False$;
12 Change speed to original desired speed;
13 **end**
14 **end**
15 Modify $\mathbf{x}_i^{ref}$ according to x_i^c and desired speed;
16 $\mathbf{x}_i^{(0)} \leftarrow \mathbf{x}_i$;
17 Solve optimization problem (13) for $\mathbf{x}_i$
18 **end**

A. Distributed MPC Formulation

The optimization problem (9) has a coupling among vehicles due to collision avoidance constraints (9b). To break the coupling and present a distributed approach, we assume vehicles are able to send and receive the trajectories through V2V communication. In the sub-problem solved by the ith vehicle, we keep $\mathbf{x}_j$ fixed, denoted by $\bar{\mathbf{x}}_j = \left[\bar{x}_i^1; \bar{x}_i^2; ...; \bar{x}_i^H\right]$. Now for the ith vehicle, the MPC formulation becomes

$$\min_{\mathbf{x}_i, s_i} \quad J_i(\mathbf{x}_i, s_i) \tag{10a}$$

$$\text{s.t.} \quad \phi(x_i^h, \bar{x}_j^h) = d(x_i^h, \bar{x}_j^h) - d \geq 0, \tag{10b}$$

$$x_i^1 = x_i^c + s_i, \tag{10c}$$

$$\forall j \in \mathcal{V}\backslash\{i\}, h \in \{1, 2, ..., H\}.$$

B. Vehicle Representation

The geometric representation for vehicles from the perspective of vehicle i is shown in Fig. 1. Vehicle i itself is represented by a circle with radius r, and the surrounding vehicles are represented by a rectangle with length $2l$ and width $2w$. We use $\mathcal{O}$ to denote the rectangle and modify the collision avoidance constraints (10b) as

$$\phi(x_i^h, \bar{\mathcal{O}}_j^h) = d(x_i^h, \bar{\mathcal{O}}_j^h) - r \geq 0, \tag{11}$$

where $d(x_i^h, \bar{\mathcal{O}}_j^h)$ is the signed distance function from x_i^h to $\bar{\mathcal{O}}_j^h$.

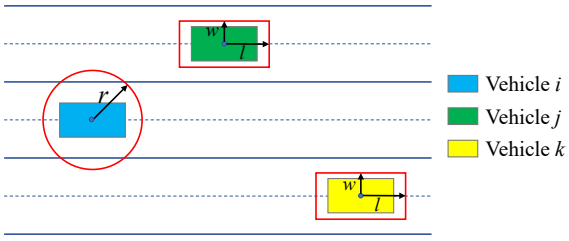
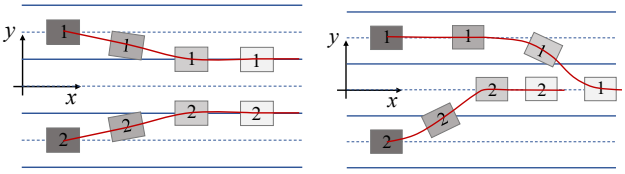


Fig. 1. Vehicles' representation for collision avoidance from the perspective of vehicle i.



(a) Two vehicles in a deadlock situation. (b) Two vehicles forming a platoon without a deadlock.

Fig. 2. Demonstration of deadlock breaking. The shallower color represents the future time steps and the red line is the planned trajectory.

Since the rectangular shape is convex, Equation (2) can be used to find the convex feasible set, given by

$$\phi + \nabla\phi \ (x_i^h - x_i^{h(k)}) \geq 0, \tag{12}$$

where ϕ means $\phi(x_i^{h(k)}, \bar{\mathcal{O}}_j^h)$ for simplicity and k is the index of iteration.

Replacing constraint (10b) by Equation (12), the CFS-based distributed MPC is given by

$$\begin{aligned} \min_{\mathbf{x}_i, s_i} \quad & J_i(\mathbf{x}_i, s_i) && \text{(13a)} \\ \text{s.t.} \quad & \phi + \nabla\phi \ (x_i^h - x_i^{h(k)}) \geq 0 && \text{(13b)} \\ & x_i^1 = x_i^c + s_i, && \text{(13c)} \\ & \forall j \in \mathcal{V}\backslash\{i\}, h \in \{1, 2, ..., H\}. \end{aligned}$$

Note that optimization problem (13) is indeed a QP problem.

C. Deadlock Breaking

A deadlock occurs when multiple vehicles have the same reference trajectory. An example of two vehicles stuck in a deadlock is given in Fig. 2(a). Vehicle 1 and vehicle 2 have symmetric positions about x-axis. They have the same desired speed, and therefore, the same reference trajectory, which is a sequence of points on $y = 0$. Instead of forming a platoon shown in Fig. 2(b), two vehicles' planned trajectories both maintain a distance to the reference trajectory and they will move forward in parallel.

Based on this observation, we summarize the features of vehicle i stuck in a deadlock is

$$\begin{aligned} \left|\max\{d(\mathbf{x}_i^{-n}, \mathbf{x}_i^{ref})\} - \min\{d(\mathbf{x}_i^{-n}, \mathbf{x}_i^{ref})\}\right| \leq \epsilon_1, \\ \left|\mathrm{mean}\{\mathrm{d}(\mathbf{x}_\mathrm{i}^{-\mathrm{n}}, \mathbf{x}_\mathrm{i}^{\mathrm{ref}})\}\right| \geq \epsilon_2, \end{aligned} \tag{14}$$

where $\mathbf{x}_i^{-n} = \left[x_i^{H-n+1}; x_i^{H-n+2}; ...; x_i^H\right]$ is the last n points of the planned trajectory, $d(\mathbf{x}_i^{-n}, \mathbf{x}_i^{ref}) \in \mathbb{R}^n$ is the distance from these points to the reference trajectory, and ϵ_1 and ϵ_2 are tuning parameters.

To break a deadlock, a solution is to change these vehicles' desired speeds such that they are different from each other. We specify the front vehicle who has smaller $\mathrm{mean}\{\mathrm{d}(\mathbf{x}_\mathrm{i}^{-\mathrm{n}}, \mathbf{x}_\mathrm{i}^{\mathrm{ref}})\}$ will be assigned a larger speed. If two vehicles are symmetric about the reference trajectory, as shown in Fig. 2(a), the vehicle merging from the left has priority. In Fig. 2(b), vehicle 1 is assigned a larger speed than vehicle 2, and the deadlock is avoided. Once they converge to the reference trajectory, they can move at original desired speed.

This strategy of deadlock breaking assigns different speeds to vehicles and make any two vehicles have different reference trajectories. Besides, different speeds prevent multiple vehicles having the same x-position as shown in Fig. 2(a). Therefore, this strategy breaks the symmetry between vehicles that leads to a deadlock. Compared to the priority constraints proposed in MCCFS algorithm [15], our method do not increase the number of constraints. In the optimization problem (13), only the reference trajectory $\mathbf{x}_i^{ref}$ is changed according to the desired speed and the constraints are not affected.

IV. NUMERICAL RESULTS

In this section, we present simulation results of six scenarios, i.e., the unstructured road, intersection, crossing, platoon formation, merging, and overtaking scenarios. The distributed design, centralized method, and RVO are tested on a laptop with 2.60GHz Intel Core i7-9750H in Python script. Section IV-A describes the simulation setup. Section IV-B presents the simulation results. Section IV-C analyzes the performance of the proposed method.

A. Simulation Setup

1) Road Environment: The road environment is shown in Fig. 3. The solid line is the boundary of lane, and the dash line is the centerline. In the highway environment, there are three lanes with width of $4m$. Note that the x-axis aligns with the centerline of $Lane\ 1$. In the intersection environment, there are four lanes, of which the centerline is $2m$ away from the lane boundary. The y-axis aligns with the boundary of $Lane\ 0$ and $Lane\ 1$.

2) Vehicle Model: The vehicle bicycle kinematic model is applied for vehicle modeling, shown in Fig. 4. A vehicle's state includes the position (x_0, y_0), the velocity v_0, and the heading θ_0. The control input is the acceleration a and the steering angle δ. L is the wheel base. Assuming that the steering angle δ is constant during a replanning time T_r, the vehicle rotates around the instant center O with a rotation radius R. In one replanning time T_r, the distance that the

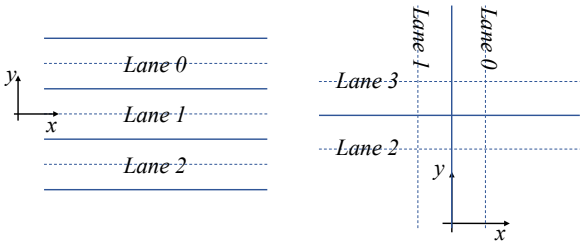


Fig. 3. The road environment. Left: highway. Right: intersection.

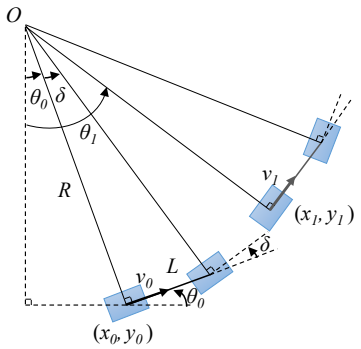


Fig. 4. The vehicle bicycle kinematic model.

vehicle travels is $L_r = v_0 T_r + \frac{1}{2} a T_r^2$ and the curvature is $\kappa = \frac{\tan\delta}{L_r}$. Then the state is updated as follows:

$$
\begin{aligned}
v_1 &= v_o + aT_r \\
\theta_1 &= \theta_o + \int_0^{L_r} \kappa \mathrm{d}s = \theta_o + \kappa L_r \\
x_1 &= x_o + \int_0^{L_r} \cos(\theta_o + \kappa s)\mathrm{d}s = x_o + \frac{\sin(\theta_o + \kappa L_r) - \sin(\theta_0)}{\kappa} \\
y_1 &= y_o + \int_0^{L_r} \sin(\theta_o + \kappa s)\mathrm{d}s = y_o + \frac{\cos(\theta_0) - \cos(\theta_o + \kappa L_r)}{\kappa}
\end{aligned}
\tag{15}
$$

3) Low-level Controller: A feedback controller is implemented to achieve low-level tracking control. The control input a and δ is a function of the difference to the planned trajectory, the desired speed, and the desired angle with respect to the planned trajectory. It is calculated after motion planning is completed, and it remains constant in a replanning time T_r. The constraints on input are $-5m/s^2 \leq a \leq 5m/s^2$ and $-45° \leq \delta \leq 45°$.

4) Overview of System Architecture: The system architecture, presented in Fig. 5, is a communication-planning-control scheme. First, trajectories are shared among vehicles. Each vehicle sends its planned trajectory and receives surrounding vehicles' information. Based on this information, its current state, and its reference trajectory, the vehicle plans the optimal trajectory. The low-level controller then computes the control command to achieve path following. Then vehicle dynamics model takes the control command and updates the vehicle's state. This process iterates every replanning time T_r.

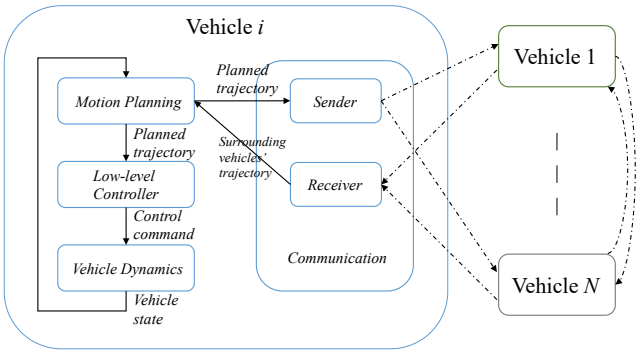


Fig. 5. System architecture.

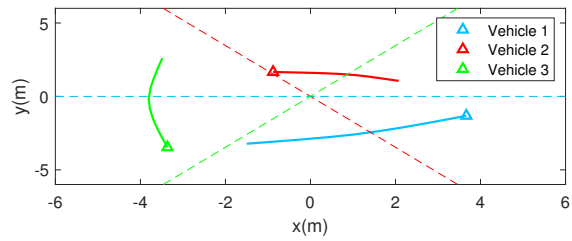
B. Simulation Results

For vehicle representation, we choose $r = 3$, $l = 1.9$, and $w = 1$. In the intersection secnario, r is changed to 2.5. In the unstructured road and intersection scenarios, we only consider trajectory planning and assume the vehicle can reach the planned position exactly at the next time step. We choose the replanning time $T_r = T_s$. In the crossing, platoon formation, merging, and overtaking scenarios, we choose $T_r = 0.02s$, and we use the low-level controller to perform tracking control. The reference trajectory of each vehicle is set as the centerline of its target lane.

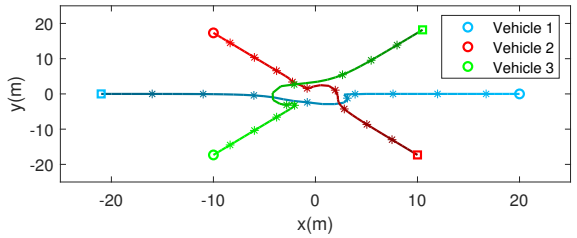
1) Unstructured Road Scenario: In this scenario, we test the distributed MPC on a unstructured road. Three vehicles are evenly distributed on a circle with radius $20m$. The reference trajectories are diameters of the circle with desired speed $10m/s$. We choose the planning horizon $H = 10$ and the sampling time $T_s = 0.1s$. The result is shown in Fig. 6. When $t = 24T_r$, presented in Fig. 6(a), the strategy to safely coordinate the vehicles is to rotate around point $(0, 0)$. The consensus is reached in $6T_r$.

2) Intersection Scenario: In this scenario, four vehicles travel at four-way-stop intersection with speed $10m/s$ and they are not allowed to deviate from the centerline. Their initial positions are $(2, 0)$, $(-2, 50)$, $(-25, 23)$, and $(25, 27)$, respectively. We choose the planning horizon $H = 10$ and the sampling time $T_s = 0.1s$. The result is given in Fig. 7. At $t = 30T_r$, all vehicles arrive at the intersection, then vehicles 3 and 4 retreat such that vehicles 1 and 2 can pass the intersection without violating collision avoidance constraints. In Fig. 7(a), the trajectories planned at $t = 36T_r$ show the consensus among vehicles. This safe coordination strategy is developed in $4T_r$.

3) Crossing Scenario: In this scenario, two vehicles with desired speed $10m/s$ start at $(0, -4)$ and $(0, 4)$ respectively. Vehicle 1 moves from $Lane\ 2$ to $Lane\ 0$, and vehicle 2 from $Lane\ 0$ to $Lane\ 2$, which creates a crossing scenario. We choose the planning horizon $H = 20$ and the sampling time $T_s = 0.1s$. The result is shown in Fig. 8. When changing lane, two vehicles negotiate about which vehicle should wait and which vehicle should cross first through trajectory sharing. Then they reach a consensus in $4T_r$ such that vehicle 2 goes and vehicle 1 waits.

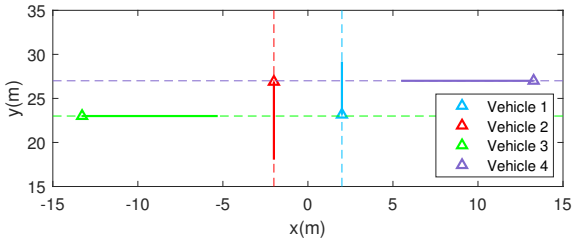


(a) The planned trajectories at $t = 24T_r$.

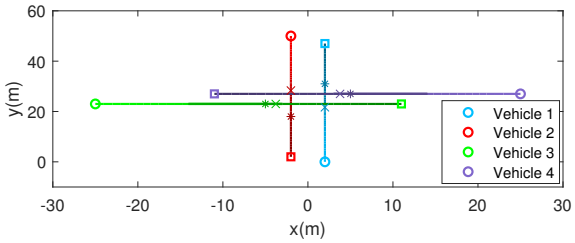


(b) The simulation result.

Fig. 6. Unstructured road situation. The triangle represents the vehicle's position when replanning, the dash line represents the original trajectory without surrounding vehicles, the circle represents the vehicle's initial position, the square represents the vehicle's end position, and the star marks the vehicle's position every $5T_r$.



(a) The planned trajectories at $t = 36T_r$.



(b) The simulation result.

Fig. 7. Intersection situation. The circle represents the vehicle's initial position, the square represents the vehicle's end position, the cross marks the vehicle's position at $t = 30T_r$, and the star marks the vehicle's position at $t = 45T_r$.

4) Platoon Formation: In this scenario, four vehicles traveling at either $Lane\ 0$ or $Lane\ 2$ with desired speed $20m/s$ are forming a platoon at $Lane\ 1$. Their initial positions are $(0, -4)$, $(6, 4)$, $(12, -4)$, and $(18, 4)$, respectively. We choose the planning horizon $H = 20$ and the sampling time $T_s = 0.1s$. The result is given in Fig. 9. Four vehicles successfully form a train-like platoon, and maintain the safe distance with each other. The consensus is reached in $1T_r$.

5) Merging Scenario: In this scenario, two vehicles travels at $Lane\ 1$ with desired speed $10m/s$, and another two vehicles at $Lane\ 0$ with the same desired speed intend to merge into $Lane\ 1$. We choose the planning horizon $H = 25$, the sampling time $T_s = 0.1s$, $n = 5$, $\epsilon_1 = 0.01$, and $\epsilon_2 = 0.2$. The result is presented in Fig. 10. Unlike the platoon formation scenario presented above, four vehicles can be stuck in a deadlock and travel in parallel. When a deadlock is detected, vehicle 4 changes its speed to $25m/s$ and vehicle 3 speeds up to $20m/s$. Note that vehicles 1 and 2 maintain their speeds since they are moving on the centerline of $Lane\ 1$. Since vehicles 4 and 3 are faster, they merge into $Lane\ 1$ in sequence in front of vehicles 1 and 2. The safe coordination is achieved in $3T_r$.

6) Overtaking Scenario: In this scenario, vehicle 1 traveling at $50m/s$ overtakes other three vehicles traveling at $10m/s$. Their initial positions are $(0, 0)$, $(15, 0)$, $(20, -4)$ and $(25, 0)$, respectively. We choose the planning horizon $H = 30$ and the sampling time $T_s = 0.2s$. The result is shown in Fig. 11. Vehicle 1 changes lane in order to avoid collision with vehicles being overtaken, and other vehicles can follow their reference trajectories. The consensus among vehicles is reached in $8T_r$.

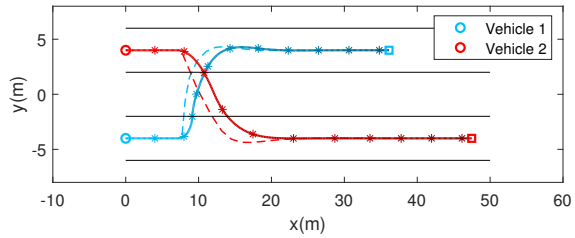
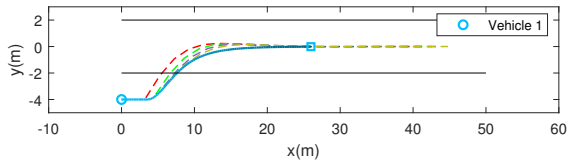


Fig. 8. Crossing situation. The circle represents the vehicle's initial position, the square represents the vehicle's end position, and the star marks the vehicle's position every $25T_r$. The dash line is the planned trajectory when vehicles reach a consensus.

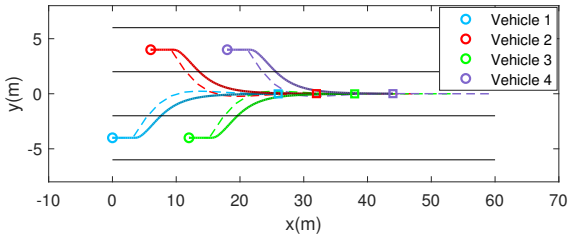
C. Performance Analysis

1) Distributed Versus Centralized Design: The proposed CFS-based distributed MPC approach is compared to a centralized approach, MCCFS [15], in terms of computation time and the sum of optimal objective function. We simulate the formation of two to five vehicles. We choose $H = 20$, $T_r = 0.02s$, $T_s = 0.1s$, and total simulation time $t = 100T_r$ in the cases with tracking control. The parameters are $H = 20$, $T_r = T_s = 0.1s$, and $t = 30T_r$ for the cases without a controller. The result is given in Table I and Table II.

Both methods ensure safe motion coordination. However, Table I shows that the distributed approach outperforms the MCCFS by more than an order of magnitude in terms of total average and maximum time. Besides, since each vehicle only solves a QP problem in the distributed approach, the computation time is short and reasonable for online implementation. In Table II, the centralized approach gives lower value of the sum of optimal objective function. The result in Table I and Table II shows the trade-off between computation time and the optimality of the planned trajectory.



(a) The planned trajectories every $5T_r$ and the executed trajectory of vehicle 1.



(b) The simulation result.

Fig. 9. Platoon formation. The circle represents the vehicle's initial position and the square represents the vehicle's end position. The dash line is the planned trajectory.

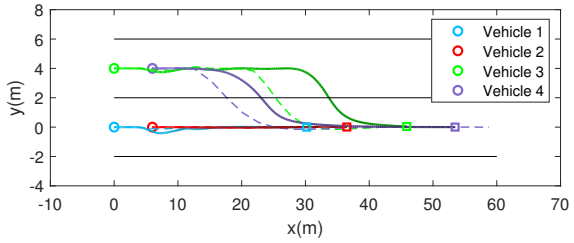


Fig. 10. Merging situation. The circle represents the vehicle's initial position and the square represents the vehicle's end position. The dash line is the planned trajectory when vehicles reach a consensus.

2) Tracking Errors: The vehicle dynamics and control input saturation are not taken into account in the MPC design. The planned trajectory then can violate these constraints and the vehicle cannot track the planned trajectory perfectly. The tracking errors jeopardize the optimality of the executed trajectory. Take platoon formation as an example. The comparison between planned trajectories and the executed trajectory of vehicle 1 is presented in Fig. 9(a). During lane changing, vehicle 1 has a mean cross-track error of $0.023m$. It tends to react slowly and deviates from the optimal trajectory given by the motion planner. Compared to the planned trajectory, the resulting trajectory take more time and distance to converge to the reference trajectory. The sum of optimal objective function during the platoon formation with and without tracking errors is presented in Table III. Relevant parameters are $H = 20$, $T_r = T_s = 0.02s$, and $t = 60T_r$. The tracking errors result in loss of around $25 - 40\%$ optimality.

In the presence of tracking errors, the proposed method is able to safely and efficiently coordinate multiple vehicles in various scenarios, which demonstrate its robustness.

3) Optimization-based Versus Scheduling Design: To solve the coordination problem at an intersection, [17] uses a decision maker to decide passing order and then a motion planner to adjust the speed. The optimality and computational complexity of this method rely on conflict zone resolution. On the contrary, the CFS-based distributed MPC do not decouple decision making and motion planning, and do not formulate conflict zones explicitly. The proposed method take into account the interaction among vehicles and allow vehicles to reach a consensus by V2V communication. Therefore, a decision maker for a scheduling is not necessary in our system. However, we observe that some vehicles need to move backwards to satisfy collision avoidance constraints. This might cause deadlock when the traffic density goes up.

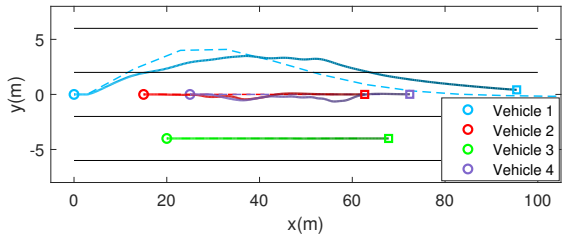


Fig. 11. Overtaking situation. The circle represents the vehicle's initial position and the square represents the vehicle's end position. The dash line is the planned trajectory when vehicles reach a consensus.

TABLE I

COMPUTATION TIME (IN SECOND) FOR CENTRALIZED AND DISTRIBUTED APPROACHES.

Number	Centralized		Distributed		
	Avg.	Max.	Avg. (Each)	Avg. (Total)	Max. (Total)
2	0.1853	0.2293	0.0048	0.0096	0.0124
3	0.4313	0.4726	0.0091	0.0272	0.0381
4	0.7768	1.1257	0.0129	0.0514	0.0700
5	1.2780	1.7829	0.0183	0.0913	0.1667

4) Distributed MPC Versus RVO: We simulate the unstructured road scenario to compare the MPC design with RVO, in terms of average trajectory length of each vehicle, time duration to reach goal, and computation time. The result is summarized in Table IV. The performance of our method is comparable to RVO in terms of trajectory length and time duration. The computation time of RVO is two to three times faster than the MPC. However, with increasing number of agents (e.g. larger than 10), fine tuning is required to obtain a good result. Besides, mean length and time duration of MPC design is significantly larger than that of RVO. Therefore, the scalability of our method in such tight environments is a problem.

V. CONCLUSIONS

This paper proposed CFS-based distributed MPC approach for multi-vehicle motion coordination. Assuming that vehicles are ablem to communicate with each other, the coupling in the centralized problem is removed to formulate a distributed MPC. The non-convex collision avoidance constraints are convexified using CFS algorithm, and the subproblem solved by each vehicle is transformed into a QP

TABLE II

THE SUM OF OPTIMAL OBJECTIVE FUNCTION FOR BOTH CENTRALIZED AND DISTRIBUTED APPROACHES WITH AND WITHOUT TRACKING ERRORS.

Number	w/		w/o	
	Centralized	Distributed	Centralized	Distributed
2	-19.4137	-16.3195	-18.2504	-9.1265
3	-34.9888	-28.2059	-30.3456	-15.1708
4	-55.4399	-43.0571	-44.7087	-22.3435
5	-81.4872	-61.2330	-61.5559	-30.7508

TABLE III

THE SUM OF OPTIMAL OBJECTIVE FUNCTION FOR THE DISTRIBUTED APPROACH WITH AND WITHOUT TRACKING ERRORS.

Number	2	3	4	5
w/	-11.2205	-25.8210	-49.2955	-83.8040
w/o	-18.1184	-37.9775	-67.9167	-110.0959
percentage	61.93%	67.99%	72.58%	76.12%

problem. We showed how to detect a deadlock and escape it by changing vehicles' desired speed. The proposed approach was validated through motion coordination of autonomous vehicles in unstructured road, intersection, crossing, platoon formation, merging, and overtaking scenarios. We showed the distributed approach outperforms the MCCFS by more than an order of magnitude. and it is computationally efficient for real time implementation. We showed that the tracking errors cause the loss of optimality and that a decision maker for a passing order is not necessary in our system. We showed that our method is comparable to RVO, but its drawback is the scalability in tight environments.

In the future work, we will test with real vehicle experiment and in more complex scenarios (e.g. intersection with larger networks of vehicles). We plan to investigate the robustness of the proposed approach in the presence of communication delays, measurement noise, etc.

REFERENCES

[1] Y. Guo and L. E. Parker, "A distributed and optimal motion planning approach for multiple mobile robots," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 3. IEEE, 2002, pp. 2612–2619.

[2] M. Bennewitz, W. Burgard, and S. Thrun, "Optimizing schedules for prioritized path planning of multi-robot systems," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, vol. 1. IEEE, 2001, pp. 271–276.

[3] L. Wachter, J. Murphy, and L. Ray, "Potential function control for multiple high-speed nonholonomic robots," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 1781–1782.

[4] H. G. Tanner and A. Kumar, "Towards decentralization of multi-robot navigation functions," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 4132–4137.

[5] J. Bruce and M. M. Veloso, "Real-time randomized path planning for robot navigation," in *Robot Soccer World Cup*. Springer, 2002, pp. 288–295.

TABLE IV

COMPARISON BETWEEN RVO AND DISTRIBUTED MPC APPROACH.

Number	Avg. Length (m)		Time Duration (s)		Avg. Computation Time (s)	
	RVO	MPC	RVO	MPC	RVO (Each)	MPC (Each)
2	42.57	41.61	5.5	5.3	0.0015	0.0022
4	44.85	48.63	8.2	6.4	0.0022	0.0052
6	43.97	45.85	7.2	5.7	0.0027	0.0083

[6] K. M. Wurm, C. Stachniss, and W. Burgard, "Coordinated multi-robot exploration using a segmentation of the environment," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 1160–1165.

[7] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *International Journal of Advanced Robotic Systems*, vol. 10, no. 12, p. 399, 2013.

[8] T. A. Johansen, T. I. Fossen, and S. P. Berge, "Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 1, pp. 211–216, 2004.

[9] J. Schulman, J. Ho, A. X. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization." in *Robotics: science and systems*, vol. 9, no. 1. Citeseer, 2013, pp. 1–10.

[10] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for bertha a local, continuous method," in *2014 IEEE intelligent vehicles symposium proceedings*. IEEE, 2014, pp. 450–457.

[11] C. Liu and M. Tomizuka, "Real time trajectory optimization for nonlinear robotic systems: Relaxation and convexification," *Systems & Control Letters*, vol. 108, pp. 56–63, 2017.

[12] C. Liu, C.-Y. Lin, Y. Wang, and M. Tomizuka, "Convex feasible set algorithm for constrained trajectory smoothing," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 4177–4182.

[13] C. Liu, C.-Y. Lin, and M. Tomizuka, "The convex feasible set algorithm for real time optimization in motion planning," *SIAM Journal on Control and optimization*, vol. 56, no. 4, pp. 2712–2733, 2018.

[14] J. Chen, C. Liu, and M. Tomizuka, "Foad: Fast optimization-based autonomous driving motion planner," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 4725–4732.

[15] J. Huang and C. Liu, "Multi-car convex feasible set algorithm in trajectory planning," in *Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2020.

[16] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Transactions on Industrial informatics*, vol. 9, no. 1, pp. 427–438, 2012.

[17] C. Liu, C.-W. Lin, S. Shiraishi, and M. Tomizuka, "Distributed conflict resolution for connected autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 1, pp. 18–29, 2017.

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

[19] R. Firoozi, L. Ferranti, X. Zhang, S. Nejadnik, and F. Borrelli, "A distributed multi-robot coordination algorithm for navigation in tight environments," *arXiv preprint arXiv:2006.11492*, 2020.

[20] T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G. J. Balas, "Decentralized receding horizon control and coordination of autonomous vehicle formations," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 1, pp. 19–33, 2007.

[21] F. Rey, Z. Pan, A. Hauswirth, and J. Lygeros, "Fully decentralized admm for coordination and collision avoidance," in *2018 European Control Conference (ECC)*. IEEE, 2018, pp. 825–830.

[22] L. Ferranti, R. R. Negenborn, T. Keviczky, and J. Alonso-Mora, "Coordination of multiple vessels via distributed nonlinear model predictive control," in *2018 European Control Conference (ECC)*. IEEE, 2018, pp. 2523–2528.