IEEE Control Systems Letters paper presented at
2023 American Control Conference (ACC)
San Diego, CA, USA. May 31 - June 2, 2023

Distributed formation trajectory planning for multi-vehicle systems

Binh Nguyen[1], Truong Nghiem[2], Linh Nguyen[3],
Tung Nguyen[4], Hung La[5], Mehdi Sookhak[6], Thang Nguyen[1⋆]

***Abstract*— This paper addresses the problem of distributed formation trajectory planning for multi-vehicle systems with collision avoidance among vehicles. Unlike some previous distributed formation trajectory planning methods, our proposed approach offers great flexibility in handling computational tasks for each vehicle when the global formation of all the vehicles changes. It affords the system the ability to adapt to the computational capabilities of the vehicles. Furthermore, global formation constraints can be handled at any selected vehicles. Thus, any formation change can be effectively updated without recomputing all local formations at all the vehicles. To guarantee the above features, we first formulate a dynamic consensus-based optimization problem to achieve desired formations while guaranteeing collision avoidance among vehicles. Then, the optimization problem is effectively solved by ADMM-based or alternating projection-based algorithms, which are also presented. Theoretical analysis is provided not only to ensure the convergence of our method but also to show that the proposed algorithm can surely be implemented in a fully distributed manner. The effectiveness of the proposed method is illustrated by a numerical example of a 9-vehicle system.**

I. Introduction

Over the past decade, real-time trajectory planning plays a crucial role in motion planning for autonomous vehicles in unknown environments. Especially for sophisticated cooperation, the planning problem has become more challenging due to a large number of vehicles operating in a narrow space and with limited computational resources. For this case, collision avoidance and distributed computational framework are a prerequisite for designing goal-oriented trajectories regarding vehicle dynamics, neighbors, or dynamic obstacles [1]–[4].

There are fruitful approaches devoted to dealing with the problems of trajectory planning for multiple vehicles. Among them, the optimization-based approach where trajectories are obtained by solutions to designed optimization problems has gained much attention due to its solid mathematical foundation. The studies in [5], [6] formulated the planning problem as mixed-integer linear and quadratic programming (MILP and MIQP) to handle collision avoidance by mixed-integer box constraints. The MILP and MIQP techniques face significant challenges in online distributed implementation due to their computational burdens. Recently, Sequential Convex Programming (SCP) [7], and Distributed Model Predictive Control (DMPC) [8] have succeeded in solving point-to-point trajectory generation for multiple vehicles. More recently, formation trajectory planning has been investigated in [9], [10] to address both target point tracking and formation preserving problems.

In a typical distributed formation trajectory planning method [9], [10], each vehicle solves a local problem that involves its local dynamics and local formation constraints with its neighboring vehicles. The neighborhood of a vehicle usually determines both the communication topology and the local formation maintained by the vehicle. While such a method has computational and communication benefits, it also has several drawbacks. First, as the vehicles move, the communication topology may change, which varies the neighborhood of each vehicle and therefore its local formation constraints must be adjusted in real time. This is difficult because each vehicle must then store its relative formation to each and every other vehicle so that it can adjust its local formation constraints based on which vehicles are its neighbors at the current time. Second, the computation distribution between vehicle agents is inflexible because each must solve a local problem consisting of all local dynamics and constraints. In practice, vehicles may have different computational capabilities, thus it is beneficial to be able to flexibly distribute computations among them according to their capabilities. Finally, it is difficult to change the global formation of all vehicles during operation since the local formation of each vehicle agent must be updated in real time. These drawbacks stem from the rigid structure of the communication and computation of the vehicle network.

This paper proposes a novel approach to overcome the above challenges, which formulates a dynamic consensus-based optimization problem to achieve desired formations while guaranteeing collision avoidance among vehicles. Our method allows the communication topology to vary in run time without affecting the success of the algorithm, as long as the resulting communication graph is strongly connected. More importantly, our method allows the computation to be freely distributed among the vehicle agents according to their computational capabilities. This flexibility even enables the computation tasks to be migrated or re-balanced during run

⋆Corresponding author: thang.nguyen@tamucc.edu

This work was funded by the U.S. National Science Foundation (NSF) under grants NSF-CAREER: 1846513 and NSF-PFI-TT: 1919127.

[1]Binh T. Nguyen and Thang Nguyen are with the Department of Engineering, Texas A&M University–Corpus Christi, Corpus Christi, TX 78412, USA

[2]Truong Nghiem is with School of Informatics, Computing, and Cyber Systems, Northern Arizona University, Flagstaff, AZ 86011, USA

[3]Linh Nguyen is with School of Engineering, Information Technology and Physical Sciences, Federation University Australia, Churchill 3842, VIC, Australia

[4]Tung Nguyen is with the Department of Information Technology, Uppsala University, PO Box 337, SE-75105, Uppsala, Sweden

[5]Hung La is with the Advanced Robotics and Automation (ARA) Lab, Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557, USA.

[6]Mehdi Sookhak is with the Department of Computing Sciences, Texas A&M University–Corpus Christi, Corpus Christi, TX 78412, USA

time, for example when a vehicle suddenly has computation or energy issues. Moreover, because in our method the global formation constraints can be handled at any selected vehicle agents, any formation change can be easily and quickly updated at those agents without having to recompute all local formations at all agents.

Notation: $\mathbf{C}_R$ stands for a circle with radius R, i.e., $\mathbf{C}_R = \{x \in \mathbb{R}^2 |\ \|x\|_2 \leq R\}$; $\oplus$ and $\mathbf{1}$ represent Minkowski sum and 1-element vector; with the index set $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$, denote $[\mathbf{v}_s]_{s\in\mathcal{S}} = \left[\mathbf{v}_{s_1}^\top,\ \mathbf{v}_{s_2}^T, \ldots, \mathbf{v}_{s_n}^\top\right]^\top$. For two vectors $x = [x_i]$ and $y = [y_i]$ with the same dimension, we denote $x \preceq y$ when $x_i \leq y_i,\ \forall i$.

II. Preliminaries

A. Description of vehicles and formation

Consider a group of N_v vehicles in a two-dimensional (2D) space. Let us define a bounded convex set $\mathbf{S} \in \mathbb{R}^2$ as the space where the vehicles operate. Each vehicle is considered a circle with radius R centered at $p_i(t) \in \mathbb{R}^2$, for $i \in \mathcal{V} = \{1, 2, \ldots, N_v\}$. Here, the variable $t \in \mathbb{N}$ is the discrete-time instant. Among the group of vehicles, let us select a leader indexed $\ell \in \mathcal{V}$, and the others are followers. The leader must reach its destination point while the group of vehicles achieves a given formation. We assume that each vehicle can make bi-directional communication with its neighbors and the communication network of the vehicles forms a strongly connected graph $\mathcal{G}(t)$ at all times. Note that the communication graph $\mathcal{G}(t)$ may vary over time t but is always strongly connected. At time t, let $\mathcal{N}_i(t)$ be the set of neighbors of vehicle i, $|\mathcal{N}_i(t)|$ denote the number of elements in $\mathcal{N}_i(t)$, $\bar{\mathcal{N}_i}(t) = \mathcal{N}_i(t) \cup \{i\}$ and $\mathcal{E}(t)$ be the set of (i, j) such that vehicle i has a communication channel with vehicle j (i.e., $\mathcal{E}(t)$ is the set of edges of $\mathcal{G}(t)$).

This work uses the definition of the displacement-based formation [11] to define the desired formation in a group of vehicles via their relative positions. Define the index set as $\mathcal{F} = \{(i, j) \,|\, i \in \mathcal{V}, j \in \mathcal{V}, i < j\}$ such that for each $(i, j) \in \mathcal{F}$, let $F_{ij}(t) \in \mathbb{R}^2$ be the relative position between vehicle i and vehicle j in the desired formation. Differing from other distributed algorithms where each vehicle knows its relative positions to other vehicles [12], [13] or references [14], this paper proposes a distributed algorithm for leader-follower formation trajectory planning, in which each follower is not required to know its relative position in the formation. This feature overcomes a challenge of conventional methods that, when a change of formation occurs, will require updating all relative positions for all vehicles, which takes at least N_v communication channels.

B. Problem formulation

This section presents an online distributed trajectory planning problem for multiple vehicles. The sampling-based motion planning approach [15] is used to design the reference trajectory of each vehicle in the group with a small sampling time $\tau > 0$. Fig. 1 illustrates the motion planning approach. Here, the reference trajectories are replanned every $N_p \geq 1$ sampling steps, the discrete time steps t_{k-1}, t_k, and t_{k+1}

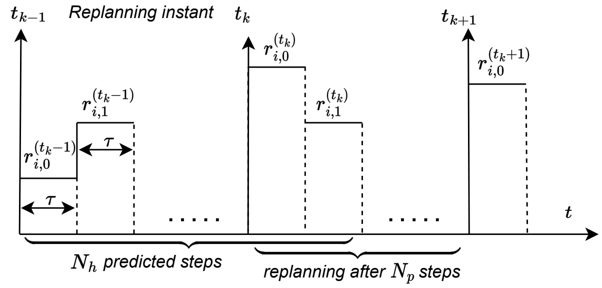

Fig. 1: Illustration of the MPC-based trajectory planning.

are consecutive replanning instants (where $t_k - t_{k-1} = N_p$), and between these instants the reference trajectories are fixed. See [16], [17] for methods for determining replanning period N_p and replanning instants. At each replanning instant t_k, we define the sampled trajectory planning as a sequence $\{r_{i,h}^{(t_k)} = r_i(t_k + h)\}_{h=1,2,\ldots,N_h}$, where $N_h \geq N_p$ is the number of predictive steps of the planned trajectory. In this work, we assume that N_p and N_h are predetermined and constant for all times.

For smoothness of the planned trajectories, we aim to design $r_{i,h}^{(t_k)}$ that follows the discrete-time linear dynamics:

$$z_{i,h+1}^{(t_k)} = A_{d,i}^{(t_k)} z_{i,h}^{(t_k)} + B_{d,i}^{(t_k)} u_{i,h}^{(t_k)}, \tag{1}$$

where $z_{i,h}^{(t_k)} = \left[r_{i,h}^{(t_k)},\ v_{i,h}^{(t_k)}\right]^\top \in \mathbb{R}^4$, $A_{d,i}^{(t_k)} \in \mathbb{R}^{4\times4}$, $B_{d,i}^{(t_k)} \in \mathbb{R}^{4\times2}$; $v_{i,h}^{(t_k)}$ and $u_{i,h}^{(t_k)} \in \mathbb{R}^2$ denotes the designed velocities and navigation input, respectively. Particularly, the linear model (1) can be considered as the simplified or linearized dynamics of vehicles where the matrices $A_{d,i}^{(t_k)}$ and $B_{d,i}^{(t_k)}$ can be obtained, for example, by linearizing nonlinear vehicle dynamics at replanning instant t_k.

With the initial conditions $r_{i,0}^{(t_k)} = p_i(t_k)$, we assume that the each vehicle can track its reference trajectory perfectly ($r_{i,h}^{(t_k)} = p_i(t_k + h)$ for $h = 1, 2, \ldots, N_p$) and the planned trajectories should meet the following requirements:

- (i) vehicles i and j do not intersect, i.e, $\|r_{i,h}^{(t_k)} - r_{j,h}^{(t_k)}\|_2 \geq 2R$ for all $(i, j) \in \mathcal{V} \times \mathcal{V} \setminus \{i\}$, $h \in [0,\ H)$
- (ii) $r_i^{(t_k)}$ is bounded and differentiable, i.e., $\|r_{i,h}^{(t_k)}\|_2$ and $\|v_{i,h}^{(t_k)}\|_2$ are bounded by geometrical limitations.
- (iii) The group of vehicles reaches its desired formation.
- (iv) The leader reaches its destination r_F as t_k increases.

Further, we consider the following geometrical and physical limitations on the discrete model (1) as follow:

$$r_{i,h}^{(t_k)} \in \mathcal{S},\ |v_{i,h}^{(t_k)}| \preceq \bar{v}_i\mathbf{1},\ |u_{i,h}^{(t_k)}| \preceq \bar{u}_i\mathbf{1},\ (\bar{v}_i, \bar{u}_i > 0). \tag{2}$$

III. Method

This section presents the MPC approach to designing the trajectory $r_{i,h}^{(t_k)}$ for $h = 1, \ldots, N_h$ without collisions among the group of vehicles where N_h is the predictive horizon. For each vehicle, let $\xi_{ij,h}^{(t_k)} \in \mathbb{R}^2$ be a predicted (or virtual) position of vehicle j at predictive step $h \in \mathcal{H} = \{0, 1, 2, \ldots, N_h\}$ calculated by vehicle i for $(i, j) \in \mathcal{V} \times \mathcal{V}$. Accordingly, let $\xi_{ii,0}^{(t_k)} = r_{i,0}^{(t_k)} = p_i(t_k)$ and $\xi_{ii,h}^{(t_k)} = r_{i,h}^{(t_k)}$. By taking advantage of the consensus principle, the vehicle i can create a replica of the predicted trajectory of vehicle j by

$\xi_{ij}^{(t_k)}$ even when there is no communication between them. The advantages can guarantee collision avoidance since the connection between two close vehicles is lost.

A. Collision avoidance between two vehicles

The collision avoidance requirement (i) can be rewritten as $r_{i,h}^{(t_k)} - r_{j,h}^{(t_k)} \notin \mathbf{C}_{2R}$. It should be noted that collision avoidance results in non-convex constraints. Thus, the obtained optimization problem is not solved effectively. With the help of the above-defined virtual position, we are concerned with a relaxed version of collision avoidance by the following lemma:

Lemma 1: For vector functions $f_1, f_2 : t \to \mathbb{R}^2$, if there exists a unit vector $\mathbf{e} : t \to \mathbb{R}^2$ ($\|\mathbf{e}\|_2 = 1$) such that

$$(f_1 - f_2)^\top \mathbf{e} \geq R. \tag{3}$$

Then, one has $\|f_1 - f_2\|_2 \geq R$ for all t.

Proof: The proof of Lemma 1 is obtained by applying the Cauchy-Schwarz inequality to the inequality (3). ∎

Assumption 1: No collision occurs in the group of vehicles at the initial time.

With the help of Lemma 1, let us propose the collision avoidance condition between vehicles i and j as follows:

$$\left(r_{i,h}^{(t_k)} - r_{j,h}^{(t_k)}\right)^\top \mathbf{e}_{ij,h}^{(t_{k-1})} \geq 2R \tag{4}$$

where the unit vector $\mathbf{e}_{ij,h+1}^{(t_{k-1})}$ is calculated from predictive positions of previous step:

$$\mathbf{e}_{ij,h}^{(t_{k-1})} = \frac{r_{i,h}^{(t_{k-1})} - r_{j,h}^{(t_{k-1})}}{\left\|r_{i,h}^{(t_{k-1})} - r_{j,h}^{(t_{k-1})}\right\|_2},\ h = 1, 2, \ldots, N_h \tag{5}$$

and $r_{i,h}^{(t_{k-1})}$ is the h-step predicted trajectory of vehicle i in the previous time step $t_k - 1$.

Remark 1: Since $r_{j,h}^{(t_k)} = \xi_{ij,h}^{(t_k)}$ for all $h \in \mathcal{H}$, the condition (4) does not require position information from vehicle j. Hence, vehicle i does not need to detect or establish communication with vehicle j if $j \notin \mathcal{N}_i^{(t_k)}$.

In light of the collision avoidance condition (4), we will propose a distributed planning algorithm for each vehicle. This algorithm not only ensures that the leader can reach its given destination with formation achievement but also guarantees a collision-free trajectories in the future.

B. Planning Algorithm

As mentioned above, our distributed scheme is applicable even if only the leader knows the formation. We design the local cost function for the leader as follows:

$$f_\ell^{(t_k)} = f_{\ell,\text{track}}^{(t_k)} + f_{\ell,\text{form}}^{(t_k)} + \beta, \tag{6}$$

where $f_{\ell,\text{track}}^{(t_k)} = \sum_{h=1}^{N_h} \left\|r_{\ell,h}^{(t_k)} - r_F\right\|_2^2 = \sum_{h=1}^{N_h} \left\|\xi_{\ell\ell,h}^{(t_k)} - r_F\right\|_2^2$, $f_{\ell,\text{form}}^{(t_k)} = \sum_{h=1}^{N_h} \sum_{(i,j)\in\mathcal{F}} \alpha \left\|\xi_{\ell i,h}^{(t_k)} - \xi_{\ell j,h}^{(t_k)} - F_{ij,h}^{(t_k)}\right\|_2^2$, and β is the attenuation factor. Here, $f_{\ell,\text{track}}^{(t_k)}$ stands for the tracking action of the leader to its pre-planning trajectory, $f_{\ell,\text{form}}^{(t_k)}$ represents an attempt to reach the desired formation, $\alpha > 0$ is a weighting factor, and $F_{ij,h}^{(t_k)} = F_{ij}(t_k + h)$. Furthermore, the local cost functions of the followers are zero $f_i^{(t_k)} = 0$. At each time step t_k, let us consider the following distributed optimization problem

$$\underset{\xi_{ij,h}^{(t_k)}}{\text{minimize}}\ f^{(t_k)} = \underset{\xi_{ij,h}^{(t_k)}}{\text{minimize}} \sum_{i=1}^{N_v} f_i^{(t_k)}, \tag{7a}$$

$$\text{s.t. } \xi_{1j,h}^{(t_k)} = \xi_{2j,h}^{(t_k)} = \cdots = \xi_{Nj,h}^{(t_k)},\ (j,h) \in \mathcal{V} \times \mathcal{H}, \tag{7b}$$

$$\text{(1) and (2) with } \xi_{ii,0}^{(t_k)} = p_i(t_k),\ v_{i,0}^{(t_k)} = \dot{p}_i(t_k), \tag{7c}$$

$$\left(\xi_{ii,h}^{(t_k)} - \xi_{ij,h}^{(t_k)}\right)^\top \mathbf{e}_{ij,h}^{(t_{k-1})} \geq 2R + \varepsilon, (j < i), \tag{7d}$$

$$f_{\ell,\text{track}}^{(t_k)} + f_{\ell,\text{form}}^{(t_k)} \leq \gamma \left(f_{\ell,\text{track}}^{(t_{k-1})} + f_{\ell,\text{form}}^{(t_{k-1})}\right) + \beta\mu^{t_k} \tag{7e}$$

where $\mathbf{e}_{ij,h}^{(t_{k-1})}$ is given by (5) with $r_{j,h}^{(t_k)} = \xi_{ij,h}^{(t_k)}$; ε is a safe distance between two vehicles; $\gamma \in (0,1)$, $\mu \in (0,1]$ are given damping coefficients. The idea of using β is from [18], [19] to relax the condition (7e) as well as maximize the decrease of leader cost function (6).

Based on a solution of constrained convex optimization problems given in [20], we take advantages of ADMM to solve the optimization (7a). To begin with, let us denote the following local constraint sets:

$$\begin{aligned} \mathcal{C}_i = \Big\{ &\mathbf{col}(q_1, \ldots, q_{N_v}) \Big| q_i \in \mathbb{R}^{2(H+1)}, q_i = [q_{i,h}]_{h\in\mathcal{H}}; \\ &\exists v_h, u_h : \left[q_{i,h+1}^\top, v_{h+1}^\top\right]^\top = A_{d,i}\left[q_{i,h}^\top, v_h^\top\right]^\top + B_{d,i} u_h, \\ &q_{i,0} = p_i(t_k),\ v_0 = \dot{p}_i(t_k); \\ &\forall j \neq i : (q_{i,h} - q_{j,h})^\top \mathbf{e}_{ij,h}^{(t_{k-1})} \geq 2R + \varepsilon; \text{ and (2)} \Big\}. \end{aligned} \tag{8}$$

Based on the optimization problem (7a), the path-planning algorithm 1 shows the steps of the online formation trajectory planning for multiple vehicles with collision avoidance.

Algorithm 1: Planning Algorithm

Input: Number of vehicles N_v, radius R, formation $F_{ij}^{(t_k)}$, predictive horizon N_h and current position $r_{i,0}^{(t_k)} = p_i(t_k)$.

Output: $r_{i,h}^{(t_k)}$, $h = 1, 2, \ldots, N_h$.

1: **Initiate:** at $k = 1$, set $\xi_{ij,h}^{(t_{k-1})} = \xi_{ij,0}^{(0)} = r_{i,0}^{(t_k)}$.

2: Calculate matrices $A_{d,i}^{(t_k)}$ and $B_{d,i}^{(t_k)}$ in (1) by vehicles dynamics at each replanning instant t_k.

3: In vehicle i, at each time instant t_k, solve the optimization problem (7a) to obtain $\xi_{ij,h}^{(t_k)}$, $j = 1, 2, \ldots, N_v$.

4: Let $r_{i,h}^{(t_k)} = \xi_{ii,h}^{(t_k)}$.

C. Algorithm analysis

To analyze the convergence of Algorithm 1, we assume that each vehicle can track its reference, i.e. $p_i(t_k + h) = r_{i,h}^{(t_k)}$ for all $h = 1, 2, \ldots, N_p$, which implies that the end of predicted trajectory in previous replanning step is coinciding with the beginning of the current step $r_{i,N_p}^{(t_{k-1})} = r_{i,0}^{(t_k)}$.

Proposition 1: For $\mu \in (0,1)$, suppose that the optimization problem (7a) is feasible for all t_k. Then, the leader ℓ reaches its destination r_F and the group of vehicles achieves the desired formation $\mathcal{F}$ as t_k increases.

Proof: Denote $J_k = f_{\ell,\text{track}}^{(t_k)} + f_{\ell,\text{form}}^{(t_k)}$. From $(7e)$, we have $J_k \leq \gamma J_{k-1} + \beta_k \mu^k$ where β_k is a scalar solution to the optimization problem (7a). Taking a summation of both sides of the inequality and noting that β_k is upper bounded by $\bar{\beta}$, one has $(1-\gamma)\sum_{m=0}^{k} J_k \leq J_0+\bar{\beta}\frac{1-\mu^k}{1-\mu}$. Through the limits, we have $\sum_{m=0}^{\infty} J_m \leq \frac{J_0}{1-\gamma}+\frac{\bar{\beta}}{(1-\gamma)(1-\mu)}$, which implies that $\lim_{k\to\infty} J_k = 0$, $\lim_{t_k\to\infty}\|r_{\ell,0}^{(t_k)} - r_F\|_2 = 0$, and $\lim_{t_k\to\infty}\|r_{i,0}^{(t_k)} - r_{j,0}^{(t_k)} - F_{ij}^{(t_k)}\|_2 = 0$ for all $(i,j)\in\mathcal{F}$. ■

Unlike the other works on online distributed trajectory planning [10], [17], our paper discusses the convergence analysis of the designed planning algorithm by introducing (7e). The presence of the term $\beta\mu^k$ plays an important role as a relaxation for the constraint $J_k \leq \gamma J_{k-1}$. Moreover, for the case $r_{i,N_p}^{(t_{k-1})} \neq r_{i,0}^{(t_k)}$ the asymptotic convergence of J_k is not guaranteed. Hence, let $\mu = 1$ to keep J_k bounded. It should be note that β_k is only available in the leader.

Finding a solution to the optimization problem (7a) plays the most important role in Algorithm 1. We will present how to solve such a problem in a distributed manner in the next section. The superscript $^{(t_k)}$ will be removed to lighten the notations afterward.

IV. Distributed computation framework

This section presents two algorithms to solve the optimization problem (7a) in a fully distributed way. The advantages of both algorithms are analyzed in communication and computation views.

A. ADMM-based Algorithm

At the beginning, (7a) is rewritten in the consensus form

$$\text{minimize} \sum_{i=1}^{N_v} f_i(\boldsymbol{\xi}_i) + \mathbb{I}_{\mathcal{C}_i}(\mathbf{w}_i) \tag{9}$$
$$\text{subject to } \boldsymbol{\xi}_i - \mathbf{w}_j = 0,\ \forall j \in \bar{\mathcal{N}}_i^{(t_k)},$$

where $\mathbb{I}_{\mathcal{C}}$ is the indicator function of $\mathcal{C}$; $\mathbf{w}_i$ stands for constrained variables; and $(\bullet)_i = [(\bullet)_{ij}]_{j\in\mathcal{V}} = \left[[\bullet_{ij,h}]_{h\in\mathcal{H}}\right]_{j\in\mathcal{V}}$ for $(\bullet) \in \{\boldsymbol{\xi}, \mathbf{w}\}$. Then, the augmented Lagrangian of the optimization problem (9) is given by: $\mathcal{L} = \sum_{i=1}^{N_v}\mathcal{L}_i,\ \mathcal{L}_i = f_i(\boldsymbol{\xi}_i) + \mathbb{I}_{\mathcal{C}_i}(\mathbf{w}_i) + \sum_{j\in\bar{\mathcal{N}}_i^{(t_k)}}\boldsymbol{\lambda}_{ij}^{\top}(\boldsymbol{\xi}_i - \mathbf{w}_j) + \frac{\rho}{2}\|\boldsymbol{\xi}_i - \mathbf{w}_j\|_2^2$. Thus, the ADMM-based algorithm is formulated in

$$\boldsymbol{\xi}^{n+1} = \underset{\boldsymbol{\xi}}{\text{argmin}}\, \mathcal{L}(\boldsymbol{\xi}, \mathbf{w}^n, \boldsymbol{\lambda}^n), \tag{10a}$$

$$\mathbf{w}^{n+1} = \underset{\mathbf{w}}{\text{argmin}}\, \mathcal{L}(\boldsymbol{\xi}^{n+1}, \mathbf{w}, \boldsymbol{\lambda}^n), \tag{10b}$$

$$\boldsymbol{\lambda}_{ij}^{n+1} = \boldsymbol{\lambda}_{ij}^{n} + \rho\left(\boldsymbol{\xi}_j^{n+1} - \mathbf{w}_i^{n+1}\right),\ j \in \bar{\mathcal{N}}_i^{(t_k)}, \tag{10c}$$

where $\boldsymbol{\lambda}_{ij} = [\lambda_{ij,h}]_{h\in\mathcal{H}}$. Note that problem (10a) can be solved in a parallel manner by unconstrained quadratic programming

$$\boldsymbol{\xi}_i^{n+1} = \underset{\boldsymbol{\xi}_i}{\text{argmin}}\, f_i(\boldsymbol{\xi}_i) + \sum_{j\in\bar{\mathcal{N}}_i^{(t_k)}} \frac{\rho}{2}\left\|\boldsymbol{\xi}_i - \mathbf{w}_j^n + \frac{1}{\rho}\boldsymbol{\lambda}_{ij}^n\right\|_2^2. \tag{11}$$

Apart from this, (10b) is simplified by the following least square problem

$$\mathbf{w}_i^{n+1} = \underset{\mathbf{w}_i\in\mathcal{C}_i}{\text{argmin}} \sum_{j\in\bar{\mathcal{N}}_i^{(t_k)}} \left\|\boldsymbol{\xi}_j^{n+1} - \mathbf{w}_i + \frac{1}{\rho}\boldsymbol{\lambda}_{ji}^n\right\|_2^2. \tag{12}$$

The solving process is summarized in Algorithm 2. As can be seen in Algorithm 2 that each vehicle requires both constraint update $\mathbf{w}_j^n$ and dual update $\boldsymbol{\lambda}_{ij}^n$ for its computations.

Algorithm 2: ADMM-based algorithm

Input: Number of vehicles N_v, radius R; primal residual error ϵ_{pri}, penalty parameter ρ, maximum ADMM iteration number $n_{\max}$, and $\xi_{ii,0}$.
Output: $\xi_{1j,h}, \xi_{2j,h}, \ldots, \xi_{N_v j,h}$.

```
Initiate: λ^0 = 0, w_i^0 ∈ C_i.
for n = 1, 2, ..., n_max do
    Vehicle i sends w_i^n to j and gathers w_j^n from its neighbors, solve (11) to obtain ξ_i^{n+1}.
    Vehicle i sends (1/ρ)λ_ij^n + ξ_i^{n+1} to j and gathers (1/ρ)λ_ji^n + ξ_j^{n+1} from its neighbors, calculate w_i^{n+1} by (12).
    if ||ξ_i^{n+1} − w_i^{n+1}|| < ε_pri then
        return w_i^{n+1}.
    else
        Calculate λ_ij^{n+1} by (10c).
    end if
end for
```

Here, each robot is set up with the same ρ initially. The computations seem to be the weakness of Algorithm 2. To deal with such an issue, we then introduce an alternative algorithm to avoid this drawback.

B. Alternating projection-based algorithm

The alternating projection is a simple method for computing a point at the intersection of some convex sets. In this case, our work is to find a point in intersection $\cap_{i=1}^{N_v}\mathcal{C}_i$ corresponding to the cost function $\sum_{i}^{N_v} f_i^{(t_k)}$. From the literature on the dynamic consensus-based optimization problem [21], optimization problem (7a) can be solved by

$$\boldsymbol{\xi}_i^{n+1} = \frac{1}{|\bar{\mathcal{N}}_i^{(t_k)}|}\sum_{j\in\bar{\mathcal{N}}_i^{(t_k)}} \mathbf{w}_j^n, \tag{13}$$

$$\mathbf{w}_i^{n+1} = \textbf{Proj}_{\mathcal{C}_i}\left(\boldsymbol{\xi}_i^{n+1} - \gamma_n \nabla f_i(\boldsymbol{\xi}_i^{n+1})\right), \tag{14}$$

where the positive sequence $\{\gamma^n\}_{n\geq 0}$ is chosen such that $\sum_{n=0}^{\infty}\gamma_n = \infty$ and $\sum_{n=0}^{\infty}\gamma_n^2 < \infty$. Then, the alternating projection-based algorithm is summarized in Algorithm 3.

Remark 2: Compared to the Algorithm 2, Algorithm 3 is quite simpler, i.e. it does not require the additional dual update $\boldsymbol{\lambda}_{ij}^n$ (10c). Thus, it can reduce at least twice the communication burdens in the vehicle network. However, Algorithm 2 converges faster than Algorithm 3 in general and can work with various types of cost functions; for example, we can add energy cost as $f_i^{(t_k)} = \sum_{h=0}^{H-1}\|u_{i,h}^{(t_k)}\|_2^2$.

C. Convergence analysis

Intuitively, the local constraint set $\mathcal{C}_i$ is nonempty and convex at all time steps, and the local cost function $f_i^{(t_k)}$ is formulated in terms of a quadratic form. Consequently, the optimization (7a) is strictly convex. Then, for both algorithms, their convergences are ensured by preceding work [20], [21] if the optimization problem is feasible. For the fast convergence, ρ can be chosen increasingly as of [22].

Algorithm 3: Alternating projection-based algorithm

Input: N_v, R, ϵ_{pri}, maximum iteration $n_{\max}$, and $\xi_{ii,0}$.
Output: $\xi_{1j,h}, \xi_{2j,h}, \ldots, \xi_{N_v j,h}$.
Initiate: $\mathbf{w}_i^0 \in \mathcal{C}_i$.
for $n = 1, 2, \ldots, n_{\max}$ **do**
  Gather $\mathbf{w}_j^n$ and calculate $\mathbf{w}_i^{n+1}$ by (13).
  Obtain $\boldsymbol{\xi}_i^{n+1}$ by (14).
  if $\|\boldsymbol{\xi}_i^{n+1} - \mathbf{w}_i^{n+1}\| < \epsilon_{\text{pri}}$ **then**
    return $\mathbf{w}_i^{n+1}$.
  end if
end for

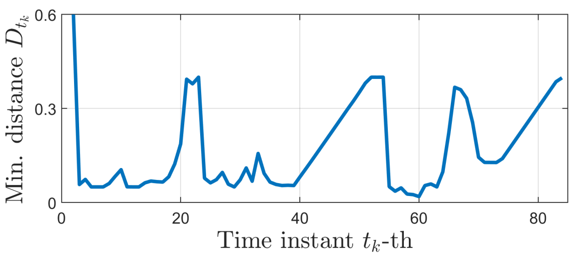

Fig. 2: Minimum distance of every pair of two vehicles

V. Numerical Examples

To demonstrate the effectiveness of the proposed algorithm 1, we present a numerical example of a 9-vehicle system that consists of a leader and eight followers moving in the interest space $\mathbf{S} = \{(x, y) \in \mathbb{R}^2 | 0 \le x, y \le 15\}$. We consider that each vehicle i shares the same linear dynamics with the trajectory generator by the double integral linear dynamics by $\ddot{r}_i(t) = u_i(t)$ or $z_{i,h+1}^{(t_k)} = Az_{i,h}^{(t_k)} + Bu_{i,h}^{(t_k)}$ as in (1). With the sampling time $\tau = 0.2[s]$, we proceed to replan trajectories after each time step $N_p = 1$. The other parameters are selected as follows: prediction horizon $H = 5$, radius of a vehicle $R = 0.3$, safety distance $\varepsilon = 0.05$.

The system is required to consecutively reach three different formations at three different locations in the interest space $\mathbf{S}$ without any collisions among the vehicles. To clearly represent the collision avoidance of the group, we define the minimum distance of every pair of vehicles as follows $D_{t_k} = \min_{(i,j)\in\mathcal{V}\times\mathcal{V}} \|r_{i,0}^{(t_k)} - r_{j,0}^{(t_k)}\|_2 - 2R$. Figs. 3a–3o illustrate snapshots of the movements of the vehicles in 84 time instants in which the leader is highlighted by a red circle and the others are followers. At the initial time instant $t_k = 1$, the vehicles are spotted at unorganized positions that highly probably result in collisions among them when moving. The first required formation is a diamond shape where the desired destination of the leader is given as $r_F = [2,\ 8]^\top$. Figs. 3a–3e show how the group of the vehicles fulfils the first requirement without any collisions. The achievement of collision avoidance is represented by checking minimum distances of every pair of vehicles (see Fig. 2 from 1-st to 22-nd time instant). In this mission, we set the parameters $\alpha = 0.1$ in the optimization problem (6) that prioritizes steering the group of vehicles to the destination point r_F rather than forming the given diamond. The second required formation is a hexagon shape where the desired destination of the leader is given as $r_F = [9,\ 9]^\top$ (see Figs. 3f–3j). It is worth noting that positions of all the vehicles in the formation are reorganized, e.g., the leader is no longer located at the center of the desired formation as the first diamond shape. This arrangement causes collisions among the vehicles. Thanks to Algorithm 1, no collisions occur (see Fig. 2). The last required formation is a square shape where the desired destination of the leader is given as $r_F = [9,\ 2]^\top$ (see Figs. 3k–3o). Once again, positions of all the vehicles in the formation are reorganized, which might cause collisions among the vehicles. However, collisions do not still occur (see Fig. 2 from 53-rd to 84-th time instant). In the last two missions, we set $\alpha = 1$ in the optimization (6) to prioritize that the group of vehicles forms the given formation shapes rather than going to the destination points. This tuning costs more time instants for the group to complete the given missions than the first mission.

VI. Concluding Remarks

This paper has addressed the problem of online distributed leader-follower formation trajectory planning for multiple vehicles with regard to collision avoidance. Our proposed method took the advantage of MPC-based motion planning approaches to design reference trajectories at each replanning instant. With the help of the global consensus scheme, each vehicle has replicas of its neighbors by which the collision avoidance among all the vehicles is established even for disconnected vehicles. Moreover, the key achievement of this paper is the formation preservation since the network topology is time-varying. The formation information is distributed to selective vehicles or only the leader while the group of vehicles maintains its desired formation and reaches the leader's destination. The validation of the planning algorithm was verified by the numerical simulation with the three different formation shapes and the leader's destinations.

References

[1] N. T. Binh, P. D. Dai, N. H. Quang, N. T. Ty, and N. M. Hung, "Flocking control for two-dimensional multiple agents with limited communication ranges," *Int. J Control*, vol. 94, no. 9, pp. 2411–2418, 2021.

[2] A. D. Dang, H. M. La, T. Nguyen, and J. Horn, "Formation control for autonomous robots with collision and obstacle avoidance using a rotational and repulsive force–based approach," *Int. J Advanced Rob. Syst.*, vol. 16, no. 3, p. 1729881419847897, 2019.

[3] T. B. Nguyen and S. H. Kim, "Distributed flocking bounded control of second-order dynamic multiple polygonal agents," *IEEE Access*, vol. 8, pp. 200 170–200 179, 2020.

[4] T. Nguyen, T.-T. Han, and H. M. La, "Distributed flocking control of mobile robots by bounded feedback," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2016, pp. 563–568.

[5] B. Alrifaee, M. G. Mamaghani, and D. Abel, "Centralized non-convex model predictive control for cooperative collision avoidance of networked vehicles," in *2014 IEEE Int. Symp. Intell. Control (ISIC)*. IEEE, 2014, pp. 1583–1588.

[6] V.-A. Le, L. Nguyen, and T. X. Nghiem, "Admm-based adaptive sampling strategy for nonholonomic mobile robotic sensor networks," *IEEE Sensors Journal*, vol. 21, no. 13, pp. 15 369–15 378, 2021.

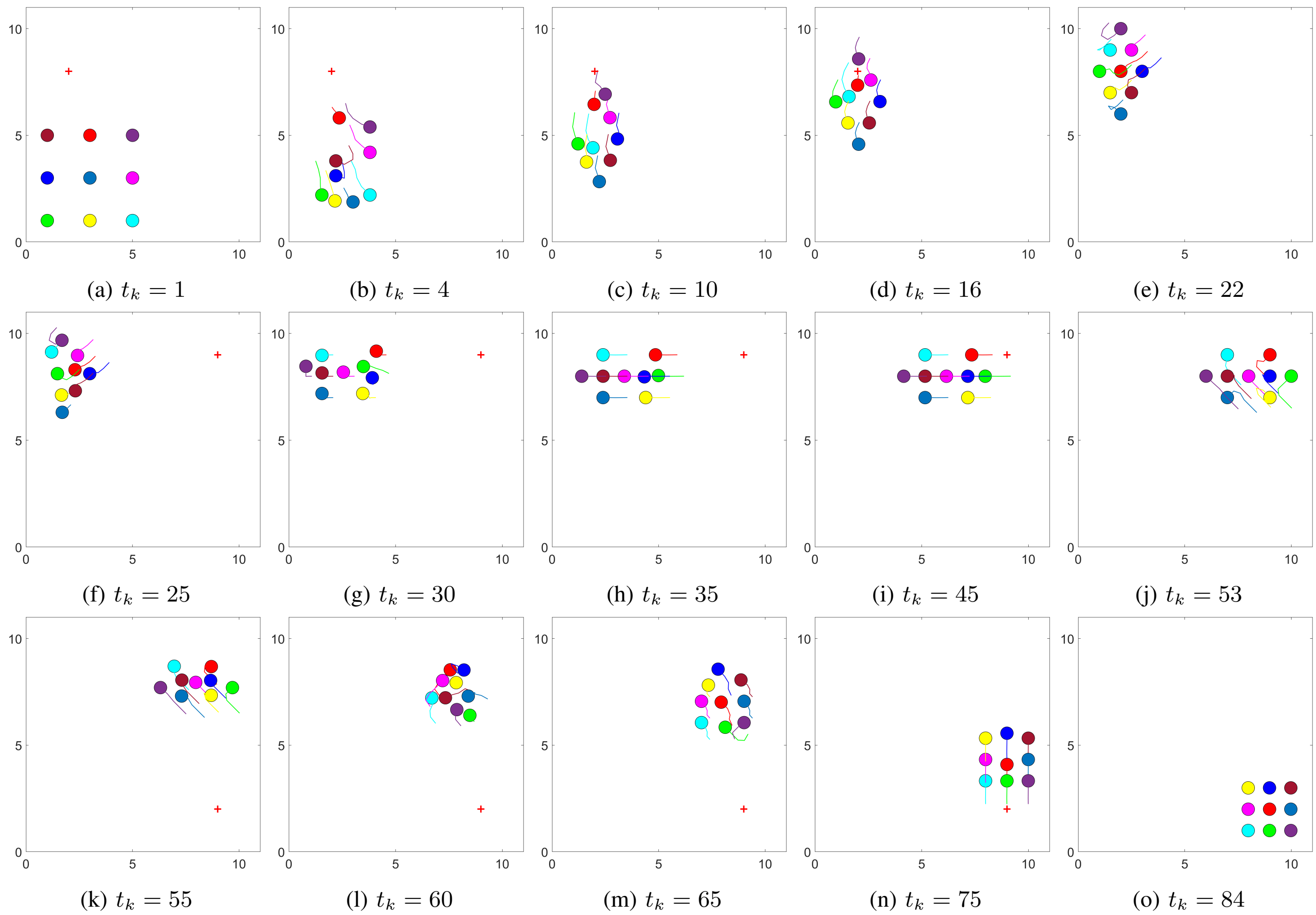

Fig. 3: Snapshots of trajectories of all the vehicles where the leader is denoted as a red circle. The group of vehicles is required to consecutively reach three different formations at three different location: (a)–(e) a diamond shape with a desired destination of the leader at $r_F = [2,\ 8]^\top$ denoted as a red plus sign; (e)–(j) a hexagon shape with a desired destination of the leader at $r_F = [9,\ 9]^\top$ denoted as a red plus sign; (j)–(o) a square shape with a desired destination of the leader at $r_F = [9,\ 2]^\top$ denoted as a red plus sign.

[7] B. Alrifaee, J. Maczijewski, and D. Abel, "Sequential convex programming mpc for dynamic vehicle collision avoidance," in *2017 IEEE Conf. Control Technology and Applications (CCTA)*. IEEE, 2017, pp. 2202–2207.

[8] C. E. Luis and A. P. Schoellig, "Trajectory generation for multiagent point-to-point transitions via distributed model predictive control," *IEEE Rob. Autom. Lett.*, vol. 4, no. 2, pp. 375–382, 2019.

[9] S. Novoth, Q. Zhang, K. Ji, and D. Yu, "Distributed formation control for multi-vehicle systems with splitting and merging capability," *IEEE Control Syst. Lett.*, vol. 5, no. 1, pp. 355–360, 2020.

[10] R. Van Parys and G. Pipeleers, "Distributed mpc for multi-vehicle systems moving in formation," *Rob. Auton. Syst.*, vol. 97, pp. 144–152, 2017.

[11] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[12] T. Nguyen, H. M. La, T. D. Le, and M. Jafari, "Formation control and obstacle avoidance of multiple rectangular agents with limited communication ranges," *IEEE Trans. Control Network Syst.*, vol. 4, no. 4, pp. 680–691, Dec 2017.

[13] R. Van Parys and G. Pipeleers, "Online distributed motion planning for multi-vehicle systems," in *2016 European Control Conf. (ECC)*. IEEE, 2016, pp. 1580–1585.

[14] A. T. Nguyen, T. B. Nguyen, and S. K. Hong, "Dynamic event-triggered time-varying formation control of second-order dynamic agents: Application to multiple quadcopters systems," *Applied Sciences*, vol. 10, no. 8, p. 2814, 2020.

[15] S. M. LaValle, *Planning algorithms*. Cambridge University Press, 2006.

[16] C. E. Luis, M. Vukosavljev, and A. P. Schoellig, "Online trajectory generation with distributed model predictive control for multi-robot motion planning," *IEEE Rob. Autom. Lett.*, vol. 5, no. 2, pp. 604–611, 2020.

[17] J. Park, D. Kim, G. C. Kim, D. Oh, and H. J. Kim, "Online distributed trajectory planning for quadrotor swarm with feasibility guarantee using linear safe corridor," *IEEE Rob. Autom. Lett.*, vol. 7, no. 2, pp. 4869–4876, 2022.

[18] Q. Nguyen and K. Sreenath, "Robust safety-critical control for dynamic robotics," *EEE Trans. Autom. Control*, vol. 67, no. 3, pp. 1073–1088, 2021.

[19] K. Garg, E. Arabi, and D. Panagou, "Fixed-time control under spatiotemporal and input constraints: A quadratic programming based approach," *Automatica*, vol. 141, p. 110314, 2022.

[20] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[21] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Auto. Control*, vol. 55, no. 4, pp. 922–938, 2010.

[22] T. B. Nguyen, T. Nguyen, T. Nghiem, L. Nguyen, J. Baca, P. Rangel, and H.-K. Song, "Collision-free minimum-time trajectory planning for multiple vehicles based on admm," in *2022 IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*. IEEE, 2022, pp. 13 785–13 790.