Formation and Reconfiguration of Tight Multi-Lane Platoons

Roya Firoozi, Xiaojing Zhang, Francesco Borrelli

Abstract—Advances in vehicular communication technologies are expected to facilitate cooperative driving in the future. Connected and Automated Vehicles (CAVs) are able to collaboratively plan and execute driving maneuvers by sharing their perceptual knowledge and future plans. In this paper, we present an architecture for autonomous navigation of tight multi-lane platoons travelling on public roads. Using the proposed approach, CAVs are able to form single or multi-lane platoons of various geometrical configurations. They are able to reshape and adjust their configurations according to changes in the environment. The proposed architecture consists of three main components: an online decision-maker, an offline motion planner and an online path-follower. The decision-maker selects the desired platoon configuration based on real-time information about the surrounding traffic. The motion planner uses an optimization-based approach for cooperative formation and reconfiguration in tight spaces. The motion planner uses a Model Predictive Control scheme to plan smooth, dynamically feasible and collision-free trajectories for all the vehicles within the platoon. The paper addresses online computation limitations by employing a family of maneuvers precomputed offline and stored on the vehicles’ control units to be executed by a low-level path-following feedback controller in real-time based on the selected desired configuration. We demonstrate the effectiveness of our approach through simulations of three case studies: 1) formation reconfiguration 2) obstacle avoidance, and 3) benchmarking against behavior-based planning in which the desired formation is achieved using a sequence of motion primitives. Videos and software can be found online here [https://github.com/RoyaFiroozi/Centralized-Planning].

I. INTRODUCTION

Vehicular wireless communication systems including vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) enhance cooperative driving by providing a communication network for information exchange between the vehicles to coordinate and plan conflict-free trajectories [1], [2]. Grouping multiple cooperative vehicles into single-lane or multi-lane formation is referred to as platooning. Using communication technologies, connected vehicles within the platoon can navigate in close proximity of each other, self-organize themselves to form certain configurations, keep tight formations and transit from one formation to another. Platooning improves traffic congestion, energy efficiency and safety [3], [4]. It increases road traffic throughput by allowing small inter-vehicle distances. Furthermore, moving with close spacing reduces aerodynamic drag and thus contributes to energy efficiency.

Platooning in classical setting refers to a group of vehicles that form a road train in a single lane [5], [6]. Single-lane platooning study and demonstrations date back to the ’80s [7], [8]. The main drawback of forming a single-lane platoon is that a long train-like platoon may prevent other vehicles to change lane and consequently affect the traffic flow and reduce the mobility. Also in case of presence of obstacles on the road it might be impossible for a long platoon to find enough gap to change lane. Platoon formation in multiple lanes incorporates the advantages of platooning described earlier and at the same time is shape-reconfigurable and is able to facilitate lane change maneuvers as needed. We refer to multi-lane platoon with small number of interconnected vehicles (three up to ten) as mini-platoon. Adding another degree of freedom in multi-lane platoon increases structure flexibility and can further improve mobility, the traffic network throughput, energy efficiency and safety compared to single-lane platoon. For example, in terms of energy efficiency, when there is slow traffic ahead in one lane, multi-lane platoon can reconfigure its shape and perform opportunistic lane change to save the energy consumption by avoiding braking and changing the lane to a faster lane [9]. In terms of safety, once an obstacle is detected in one lane, the multi-lane platoon can reconfigure and accommodate the vehicles in the blocked lane to merge into another lane to avoid the obstacle and minimize the risk of possible collision.

Although single-lane platooning is well studied in the literature, literature on multi-lane platoons is limited and reviewed in the next section. The focus of this paper is to present a general architecture for autonomous navigation of tight multi-lane platoons. The contributions are summarized as follows.

1) An architecture for autonomous navigation of multi-lane platoons on public roads is presented. It comprises a decision-maker, a motion planner and a path-follower. The decision-maker operates in real-time and is responsible for interactions of the platoon with environmental traffic. It selects the desired platoon configuration among all the predefined configurations. The offline motion planner uses an optimization-based algorithm to create various maneuvers that allow smooth transitioning between different configurations. The path-follower executes the pre-stored trajectories in real-time.

2) We identify a set of formation patterns also referred to as single-lane and multi-lane platoons and introduce a finite state machine to capture the transitions between these configurations. Each state of the finite state machine corresponds to a different configuration of the platoon. The decision-maker operates based on the
proposed state machine. The continuous maneuvers for transitions among modes are precomputed by motion planner. The guards for switching condition between the modes are defined as a function of the upcoming traffic.

3) We model the vehicles’ shape as polytopic sets and reformulate the collision avoidance constraints among them into a set of smooth constraints using strong duality theory. These smooth constraints can be handled efficiently by standard non-linear solvers. This approach allows navigation through tight spaces at highway speed.

4) Compared to existing literature, the three novel contributions discussed above address real-time implementation, tight maneuvering and hard constraint satisfaction. Uncertainty is not addressed in this work and is topic of ongoing research.

The remainder of the paper is structured as follows. Section II provides a literature review about multi-vehicle formation. Section III describes preliminaries. Section IV presents the proposed motion planning approach, and describes the decision-making and planning scheme structure. Section V introduces motion planning using sequence of motion primitives, which we will use as a benchmark to compare our proposed planning approach against. Section VI presents simulation results and Section VII concludes the paper and presents future research directions.

II. LITERATURE REVIEW

Coordinated formation methods for multiple autonomous vehicles are well-studied in the literature and can be categorized in three main approaches: Leader-follower, virtual structure, and behavior-based approach. In leader-follower approach the follower agents track the coordinates of the leader [10], [11]. This method is effective for conventional single-lane train-like platoons, but since the follower must follow the same reference trajectory as the leader, it is not applicable to reconfigurable multi-lane platoons, in which the planned motions for the vehicles are not the same. In virtual structure method, the formation is represented as a virtual rigid structure. Each robot is considered as a node in the rigid structure [12]. The main drawback of this method is that, the formation as a rigid structure is not flexible and reshappable.

Behavior-based approaches include methodologies such as flocking and particle swarm optimization algorithms, artificial potential fields, and sequence of motion primitives. Most of the studies on flocking algorithms consider the agents as a group of particles that interact with each other based on Reynolds heuristic rules of cohesion, separation and alignment [13]. Cohesion enforces the particles to stay together and separation penalizes the collision between the particles. In artificial potential field method, potential fields are built so that the robot is attracted by the goal region and repelled by the obstacle region. In formation control, in addition to goal and obstacle potential fields, a swarm attractive field is introduced to achieve the desired formation pattern. The potential-based planning does not impose hard constraint on collision avoidance and cannot guarantee collision avoidance with constrained control input. In addition, all these particle-based methods model the vehicles as particles with radial gap among them and do not take the actual size of the vehicles into account. Furthermore, the dynamic model is considered to be the particle’s dynamic with first, second or third-order point-mass models, which are not the representation of the actual nonlinear dynamics of the vehicles.

Another behavior-based method is to construct the formation maneuvers as sequences of motion primitives [14]. Motion primitives are identified as various behaviors such as lane change and obstacle avoidance. Among all the described formation approaches, this method is more effective for multi-lane platooning, but its disadvantage is that it is difficult to mathematically analyze and solve for sequence of motion primitives.

Combinations of the aforementioned approaches have also been studied. In [15], for example, the authors use the Reynolds rules to define the potential forces between the agents. Cohesion and separation are modeled as pairwise attractive and repulsive potential forces between the particle, respectively and a multi-objective cost function is constructed to satisfy all the rules simultaneously. In [16], the authors propose virtual leader approach with attractive potential field to track a desired path and achieve a desired formation and repulsive potential fields to avoid agents collisions. Also a Lyapunov function is constructed to prove the closed-loop stability. In [17], the authors use a similar approach for flocking of multiple non-holonomic vehicles and prove the convergence using LaSalle’s invariant principle.

III. PRELIMINARIES

A. Vehicle Model

The vehicles set composing the platoon is defined as $\mathcal{V}$. We consider $N_v$ vehicles and identify each vehicle through its index $i \in \mathcal{V} := \{1, 2, ..., N_v\}$. The nonlinear behavior of every vehicle $i$ within the set is modeled by the vehicle kinematic bicycle model, which is a common modeling approach in path planning. In this model, the $i$th vehicle state vector is $z^i = [x^i, y^i, \psi^i, v^i]^\top$, where $x^i$ and $y^i$ represent longitudinal and lateral positions of the vehicle, respectively, $\psi^i$ is the heading angle and $v^i$ denotes the velocity at center of gravity (C.G.) of the vehicle, as seen in Fig. 1. The control input vector is defined as $u^i = [a^i, \delta^i]^\top$, where $a^i$ is the acceleration and $\delta^i$ is the steering angle. The vehicle
dynamics is given as follows
\[
\begin{align*}
\dot{x}^i &= v^i \cos(\psi^i + \beta^i), \\
\dot{y}^i &= v^i \sin(\psi^i + \beta^i), \\
\dot{\psi}^i &= v^i \cos \beta^i, \\
\dot{\beta}^i &= \frac{β_l^i + β_r^i}{l_f^i + l_r^i} (\tan δ^i),
\end{align*}
\]  
(1)

where $\beta^i = \arctan \left( \frac{\tan δ^i (l_f^i + l_r^i)}{l_f^i + l_r^i} \right)$ is the side slip angle, $l_f^i$ and $l_r^i$ are the distance from the center of gravity to the front and rear axles, respectively. Superscript $i$ in this paper denotes the $i$th vehicle in the platoon. Using Euler discretization, we discretize (1) as follows
\[
\begin{align*}
x^i(t + 1) &= x^i(t) + \Delta t \ v^i(t) \cos(\psi^i(t) + \beta^i(t)), \\
y^i(t + 1) &= y^i(t) + \Delta t \ v^i(t) \sin(\psi^i(t) + \beta^i(t)), \\
ψ^i(t + 1) &= ψ^i(t) + \frac{v^i(t) \cos β^i(t)}{l_f^i + l_r^i} \tan δ^i(t), \\
\dot{β}^i &= \frac{β_l^i + β_r^i}{l_f^i + l_r^i} \Delta t
\end{align*}
\]  
(2)

where $\Delta t$ is the sampling time.

### B. Platoon Configuration

Various platoon formation patterns or configurations are considered in this work, including one-lane (train-like) and multi-lane (rectangle, diamond, wedge shape, etc.), as shown in Fig. 2. The platoon configuration $C$ is parameterized as $C(n_v, l, p)$, where $n_v \in \mathbb{Z}$ is the maximum number of vehicles in each lane within the platoon, $l \in \{0, 1\}^n_l$ is an indicator vector that specifies which lanes are occupied, $n_l$ is the maximum number of lanes within the platoon. The $j$th element of $l$ is defined as
\[
l(j) = \begin{cases} 
0 & \text{if no vehicle is in } j\text{th lane} \\
1 & \text{if at least one vehicle is in } j\text{th lane}
\end{cases}
\]
where $j$ denotes the lane index. The parameter matrix $p \in \mathbb{R}^{n_l \times n_v}$ represents the platoon geometrical pattern specified as the relative distances between the vehicles. Every $j$th row of matrix $p$ is defined as $p(j) = [d_{j,\text{shift}}, d_{j1}, ..., d_{j(n_v-1)}]$, where $d_{j1}, ..., d_{j(n_v-1)}$ denote the horizontal inter-vehicle distances at $j$th lane as shown in Fig. 2(b) and $d_{j,\text{shift}}$ is the horizontal shifting distance of the front-most vehicle at each lane with respect to the front end of the reference vehicle. The right-most lane in direction of travel is the reference lane for $j$th lane, as shown in Fig. 2(b) and the reference vehicle is the front-most vehicle at reference lane. For the cars ahead of the reference vehicle, $d_{j,\text{shift}}$ is considered as negative. The values of $d_{j1}, ..., d_{j(n_v-1)}$ and $d_{j,\text{shift}}$ are design parameters and might be chosen as different values for each lane. For example, the platoon configuration in Fig. 2(b) is defined as
\[
C = C(3, [1, 1, 1], p), \quad p = \begin{bmatrix} 0 & 1 & 1 \\ 2 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix},
\]

where $d_{1,\text{shift}}, d_{2,\text{shift}}$ and $d_{3,\text{shift}}$ associated with 1st, 2nd and 3rd lanes are $0m$, $2m$ and $1m$, respectively. Also $d_{11}, d_{12}, d_{21}, d_{31}$ and $d_{32}$ are all $1m$ in this configuration.

For trajectory optimization purposes, it is convenient to convert the configuration $C$ to position coordinates $(x, y)^i$ of each vehicle $i$ within the platoon. The function
\[
g : C(n_v, l, p) \rightarrow ((x, y)^1, \ldots, (x, y)^{n_v}),
\]
gets the configuration $C$ as input and outputs the position coordinates $(x, y)$ for all the vehicles. The origin $O$, as shown in Fig. 2, is defined as the position of the rear-most vehicle at the right-most lane of the platoon configuration and all the coordinates are determined with respect to that origin.

### C. Simple Reference Generator Model

We define a simple integrator function which we will use in Section IV to generate the reference trajectories for each vehicle. The function $h : \mathbb{R}^3 \rightarrow \mathbb{R}^T$, is defined as
\[
h : (x(0), v_{\max}, T) \rightarrow x_{\text{Ref}} = [x(0), x(1), \ldots, x(T)],
\]
which determines $x_{\text{Ref}}$ for all the vehicles within the platoon. The trajectory is obtained by $x(t + 1) = x(t) + v_{\max} \Delta t$, $\forall t \in \{0, 1, ..., T\}$, where $x(t)$ is the vehicle longitudinal position at time $t$, $v_{\max}$ is the maximum speed limit of the road, $T$ is the final time of simulation and $\Delta t$ is the simulation sampling time.

### IV. Architecture

Our proposed architecture for cooperative multi-vehicle systems is composed of three modules: decision making, motion planning and path following. Fig. 3 shows the architecture. The decision maker receives information about the surrounding environment from a perception unit and selects a desired platoon configuration $C_{\text{des}}$ accordingly, in real
time. A finite state machine captures different configurations and possible transitions among them. These transitions are precomputed offline by the motion planner and stored in a look-up table. This module uses an optimization-based approach for cooperative formation and reconfiguration. Once $C_{\text{des}}$ is selected by the decision maker, its corresponding trajectories in the look-up table are executed by the path-follower controller on each vehicle in real-time. We make the following assumptions:

(A1) The vehicles are fully autonomous and connected through vehicle-to-vehicle (V2V) communication.

(A2) Uncertainty due to communication delay or model mismatch is not considered; perfect knowledge of the states for all the vehicles is assumed.

To be more specific, we provide examples of the above quantities. Input $u$ in (5) is the signal contains the traffic information which can be obtained by the following examples

- average of forecast of traffic flow from the scene
- information of traffic flow ahead using V2V communication, cloud, etc.
- estimation of traffic flow ahead using radar.

Exogenous variable (exo.vars.) is detection of an obstacle on the road or slow upcoming traffic. Transitions $T_r$ look-up table consists of pre-stored trajectories. The condition $e_f$ indicates if the desired platoon configuration is attained. An example of these states and transitions in given next.

An example of platoon reconfiguration is shown in Fig. 4(b). The top snapshot shows a multi-lane platoon with initial configuration $C_i$, going forward at steady state (right-headed arrow shows the direction of motion). At specific instance $T_r$ mode is triggered (due to events $e_r$ like obstacle detection or slow traffic in the right lane) and the cars change their lane as shown in the middle snapshot. Whenever the transition maneuver is completed, another configuration $C_{f}$, which in this example is single-lane platoon, is achieved as shown in the bottom snapshot. In this example the guard $e_f$ allows the transition to the final configuration $C_{f}$ once it verifies that $\dot{y}^i \leq \epsilon_y$, $\psi^i \leq \epsilon_\psi$, $v^i = v_{\text{max}}$, where $\dot{y}^i$, $\psi^i$ and $v^i$ are the states defined in (1), $v_{\text{max}}$ is the maximum speed limit of the road and $\epsilon_y$ and $\epsilon_\psi$ are thresholds for changes in $y$ and $\psi$, respectively. In $C_i$ and $C_f$ modes the platoon is at steady state, so for these modes the controllers are lane keeping and Adaptive Cruise Control (ACC). In $T_r$ mode the controller is a path-follower.

The finite state machine $F(\cdot)$ is replicated and synchronized in each car. However, during a reconfiguration only one of the cars is the leader. The front-most car in the reference lane, is chosen as platoon leader on which the decision-making module operates to select the desired configuration. Ones the desired configuration is chosen, it will be broadcasted through V2V network to all the vehicles and each vehicle looks up in the stored library of the trajectories and find the trajectory associated with the selected desired configuration and executes its trajectory via path-following controller.

### A. Decision-Making

A general finite state machine is introduced and illustrated in Fig. 3(a) and characterized by $F(C_i, C_f, e_t, T_r, e_f)$, where

- $C_i, C_f$ are discrete states that are instances of platoon configurations. $C_i$ is the initial configuration and $C_f$ is the final configuration.
- $e_t$ is the event-triggered switch which is defined as a binary function $f_t : (u, G, \text{exo.vars.}) \rightarrow \{0, 1\}$,
  \[
  e_t = f_t(u, G, \text{exo.vars.}),
  \]
  where $u$ is the input, $G$ is the state guard and exo.vars are the exogenous variables.
- $T_r$ represents a look-up table of collision-free precomputed reference trajectories $\{z_{\text{ref}}(0), z_{\text{ref}}(1), z_{\text{ref}}(2), ..., z_{\text{ref}}(T)\}$ for all the vehicles within the platoon, from initial time 0 to final time $T$.
- $e_f$ presents the final event in which the platoon steady state is achieved and configuration $C_f$ is realized.

### B. Motion Planning

The motion planning is performed offline and the computed trajectories are stored in a look-up table. The motion planning module has a hierarchical structure. At the high level, reference trajectories $z_{\text{ref}}$ for each of the vehicles are generated, based on the selected desired configuration. These trajectories can cause collisions, which are resolved by a low level planner. At the low level, a trajectory optimization is formulated as Model Predictive Control (MPC) problem to plan smooth, dynamically feasible and collision-free trajectories for all the vehicles in a centralized optimization problem. This module incorporates the collision avoidance between the vehicles as constraints of optimization problem.
and obtains longitudinal $a^i$ and lateral $\delta^i$ control inputs for all the vehicles.

1) High-Level Reference Generation: The reference trajectory for the $i$th vehicle is denoted as $z^i_{\text{ref}} = [x^i_{\text{ref}}, y^i_{\text{ref}}, \psi^i_{\text{ref}}, v^i_{\text{ref}}]$ and is defined for the interval $[0, 1, 2, \ldots, T]$, initial time 0 until the final maneuver time $T$. \{ $z^i_{\text{ref}}(0), z^i_{\text{ref}}(1), z^i_{\text{ref}}(2), \ldots, z^i_{\text{ref}}(T)$ \} is obtained based on initial $C_i$ and final $C_f$ configurations of the platoon. First, the position coordinate of all the vehicles and the rest are defined in Section III. Then, the longitudinal position reference trajectory \{ $x^i_{\text{ref}}(0), \ldots, x^i_{\text{ref}}(T)$ \} is generated using the integrator model (4).

\begin{equation}
{x^i_{\text{ref}}(0), \ldots, x^i_{\text{ref}}(T)} = h(x^i(0), v^{\text{max}}_{\text{max}}), \tag{6}
\end{equation}

The lateral position reference trajectory $y^i_{\text{ref}}$ is the $y$ coordinate of the road centerline for each vehicle. For the first portion of simulation $(0, \ldots, \rho T)$, $y^i_{\text{ref}}$ is obtained from initial configuration $C_i$ and the rest $(\rho T + 1, \ldots, T)$ is determined by final configuration $C_f$, $g(C_f(n_v, l, p)) = (x^i(T), y^i(T)) \ \forall i \in \mathcal{V}$, which is previously defined in Section III. Then, the longitudinal position reference trajectory \{ $x^i_{\text{ref}}(0), \ldots, x^i_{\text{ref}}(T)$ \} is generated using the integrator model (4).

\begin{equation}
{y^i_{\text{ref}}(0), \ldots, y^i_{\text{ref}}(\rho T)} = y^i(0), \tag{7}
\end{equation}

\begin{equation}
{y^i_{\text{ref}}(\rho T + 1), \ldots, y^i_{\text{ref}}(T)} = y^i(T), \tag{8}
\end{equation}

the parameter $\rho \in [0, 1]$ is a design parameter. $\psi^i_{\text{ref}}$ is zero

\begin{equation}
{\psi^i_{\text{ref}}(0), \ldots, \psi^i_{\text{ref}}(T)} = 0, \tag{9}
\end{equation}

since we assume straight roads and $v^i_{\text{ref}}$ is set as maximum speed limit of the road or average traffic flow $v_{\text{max}}$.

\begin{equation}
{v^i_{\text{ref}}(0), \ldots, v^i_{\text{ref}}(T)} = v_{\text{max}}. \tag{10}
\end{equation}

The reference trajectory $z^i_{\text{ref}}$ for the $i$th vehicle is defined using (6), (7), (9), and (10). The generated trajectory is a naive initialization that might collide with obstacles. The low-level MPC planner ensures collision avoidance among the vehicles.

2) Low-Level Collision Avoidance: multi-vehicle motion planning problem is formulated as a centralized MPC optimization problem that computes conflict-free trajectories for all the vehicles in the platoon simultaneously. MPC scheme uses a receding horizon fashion. At each time step it solves an optimization problem and obtains the control input based on dynamic model predictions over a time horizon and applies the first control input solution. At the next time step, the horizon is shifted forward and the procedure is repeated. The maneuvers are computed by closed-loop simulation of MPC optimization (11) with dynamic model (1).

The objective function penalizes the deviation of each individual vehicle from the reference trajectory generated at the high level and the collision avoidance constraint is incorporated as hard constraint to guarantee safety. The optimization problem is formulated in the MPC framework as follows

\begin{equation}
\min_{u^i(t)} \sum_{i=1}^{N_v} \sum_{k=t}^{t+N-1} \left( \|Q_z(z^i(k)|t) - z^i_{\text{ref}}(k)|t)\|_2^2 + \|Q_u(u^i(k)|t)\|_2^2 + \|Q_{\Delta u}(\Delta u^i(k)|t)\|_2^2 \right) \tag{11a}
\end{equation}

subject to

\begin{equation}
z^i(k+1)|t) = f(z^i(k)|t), u^i(k)|t), \tag{11b}
\end{equation}

\begin{equation}
z^i(0)|t) = z^i(0), \tag{11c}
\end{equation}

\begin{equation}
z_{\text{min}} \leq z^i(k)|t) \leq z_{\text{max}}, \tag{11d}
\end{equation}

\begin{equation}
u_{\text{min}} \leq u^i(k)|t) \leq u_{\text{max}}, \tag{11e}
\end{equation}

\begin{equation}
\mathcal{P}(z^i(k)|t) \cap \mathcal{O} = \emptyset, r \in \mathcal{S}, \tag{11f}
\end{equation}

\begin{equation}
\mathcal{P}(z^i(k)|t) \cap \mathcal{P}(z^j(k)|t) = \emptyset, i \neq j \tag{11g}
\end{equation}

where $u^i(t) = \{u^i(t)|t), \ldots, u^i(t + N - 1)|t)\}$ denotes the sequence of control inputs over the MPC planning horizon $N$ for the $i$th vehicle. The optimal solution is $U^*(t) = \{u^i(t)|t), \ldots, u^i(t + N - 1)|t)\}$, and the receding horizon control law is obtained by applying the first control input $u_{\text{MPC}}(t) = u^i(t)$. Superscript $i$ denotes the $i$th vehicle, $N_v$ is the total number of vehicles in the platoon, $z^i(k)$ and $u^i(k)$ are the state variable and control input of $i$th vehicle at step $k$ predicted at time $t$, respectively. The weight factors $Q_z$, $Q_u$, and $Q_{\Delta u}$ are positive semidefinite matrices. The function $f(\cdot)$ in (11b) represents the vehicle kinematic bicycle model (1), which is discretized using Euler discretization. The reference trajectory obtained from the high level planner is denoted as $z^i_{\text{ref}}$ and $z_{\text{min}}$ and $z_{\text{max}}$ are the state limits and $u_{\text{min}}$ and $u_{\text{max}}$ are the input limits. $\mathcal{P}(z^i(k)|t)$ represents i-th vehicle polytope as the road area occupied by the vehicle and $\mathcal{P}(z^j(k)|t)$ represents the other vehicle polytopes as moving obstacles for $i$th vehicle. $\mathcal{O}$ represents the static obstacles and superscript $r$ denotes the $r$th static obstacle, the set $\mathcal{S} = \{1, \ldots, n\}$ represents the set of static obstacles and $n$ is the total number of static obstacles. The set of neighbors $\mathcal{N}_i$ is the set of all the vehicles within the platoon except $i$th vehicle and is defined as $\mathcal{N}_i = \mathcal{V} \setminus i$. In order to
guarantee collision avoidance, the vehicles and all the static and dynamic obstacles are modeled as polytopic sets that not only each set has empty intersection with all the other sets, but also each set keeps a minimum distance from the other sets. The collision avoidance between the \(i\)th vehicle and the static obstacles \(O\) is expressed in (11), where \(\mathcal{P}(z')\) is the polytopic set that represents the road area occupied by the \(i\)th vehicle, \(O\) denotes the static obstacles, superscript \(r\) is the \(r\)th static obstacle and \(n\) is the total number of static obstacles. The collision avoidance between the \(i\)th vehicle and all the other vehicles (neighbors) is formulated in (11g), where \(\mathcal{P}(z')\) are the polytopic sets that represent all neighbor vehicles.

The remaining of this section is devoted to detailed description and reformulation of the constraints, (11) and (11g). The computed trajectories from closed simulation of MPC optimization (11) with dynamic model (1) are stored in a look-up table and will be executed in real-time by a path-follower which is a feedback controller.

C. Representation of the Road Area Occupied by the Vehicle

As discussed platooing is maintaining close inter-vehicular distance within a group of vehicles. In tightplatooing, both road geometry (lane width) and platoon geometry (longitudinal and lateral inter-vehicle spacing) restrict the motion of the vehicles within the platoon and results in creating a tight environment. To allow navigation at tight spaces, it is essential to model the road structure and the vehicles dimensions as exact sizes with no approximation or enlargement. The vehicle pose or the corresponding road region occupied by the vehicle is defined by a two-dimensional convex polytope \(\mathcal{P}\), as seen in Fig. [5] The initial pose of the vehicle is represented as \(\mathcal{P}_0\). As the vehicle travels along the road, \(\mathcal{P}_0\) undergoes affine transformations including rotation and translation. Hence \(\mathcal{P}(z(k)) = \mathcal{P}(z(k))\mathcal{P}_0 + t(z(k))\), where \(z(k)\) represents the vehicle state at \(k\)th step, \(\mathcal{P}(z(k))\) is the vehicle occupied region as a function of the state \(z(k)\), and dimensions including length \(h\) and width \(w\) and is defined as a set of linear inequalities. \(\mathbf{R} : \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_z \times n}\) is an orthogonal rotation matrix and \(t : \mathbb{R}^{n_z} \rightarrow \mathbb{R}^n\) is the translation vector. \(n_z\) is the dimension of \(z\) and \(n\) is two, since the transformation is occurring in two-dimensional space \(\mathbb{R}^2\). The rotation matrix \(\mathbf{R}(\cdot)\) is a function of the vehicle heading angle \(\psi(k)\) and the translation vector \(t(\cdot)\) is a function of the longitudinal \(x(k)\) and lateral \(y(k)\) positions of the vehicle. So the transformed polytope is defined as \(\mathcal{P}(z(k)) = \{(p_x, p_y)\in \mathbb{R}^2 | \mathbf{A}(z(k)) [p_x, p_y]^\top \leq b(z(k))\}\), where \(p_x\) and \(p_y\) are the coordinates of points in two-dimensional space which are representation of the polytope and \(\mathbf{A}(z(k))\) and \(b(z(k))\) are defined as \(\mathbf{A}(z(k)) = \begin{bmatrix} \mathbf{R}(z(k))^\top \mathbf{R}(z(k)) \end{bmatrix}^\top\), \(\mathbf{R}(z(k)) = \begin{bmatrix} \cos(\psi(k)) & -\sin(\psi(k)) \\ \sin(\psi(k)) & \cos(\psi(k)) \end{bmatrix}\), \(b(z(k)) = [h/2, w/2, h/2, w/2]^\top + \mathbf{A}(z(k))[x(k), y(k)]^\top\).

For coordination of multiple vehicles, each vehicle’s occupied area is modeled as a time-varying polytope and at each time step, we re-plan such that no intersection occurs between the polytopic sets.

D. Collision Avoidance Reformulation

The distance between two polytopic sets \(\mathcal{P}_1\) and \(\mathcal{P}_2\) is defined as
\[
\text{dist}(\mathcal{P}_1, \mathcal{P}_2) = \min_{x, y} \|x - y\|_2 | A_1 x \leq b_1, A_2 y \leq b_2, \]
where \(\mathcal{P}_1\) and \(\mathcal{P}_2\) are described as \(A_1 x \leq b_1\) and \(A_2 y \leq b_2\), respectively. The two sets do not intersect if \(\text{dist}(\mathcal{P}_1, \mathcal{P}_2) > 0\). However, for autonomous driving applications, since the vehicles must keep a minimum safe distance \(d_{\text{min}}\) from each other and from the obstacles, the distance between their polytopic sets should be larger than a predefined minimum distance, \(\text{dist}(\mathcal{P}_1, \mathcal{P}_2) \geq d_{\text{min}}\).

In the motion planning optimization problem (11), the collision avoidance is imposed as constraint. However, the collision avoidance formulated in (12) is itself an optimization problem. Hence, we have to solve an optimization problem as the constraint of another optimization problem. To deal with this issue, as explained in [18], the dual problem can be solved instead of the primal problem (12) based on strong duality theory. The dual problem is expressed as
\[
\max_{\lambda, \mu} -b_1^\top \lambda - b_2^\top \mu: A_1^\top \lambda + s = 0, A_2^\top \mu - s = 0, \|s\| \leq 1, \lambda \geq 0, \mu \geq 0,\]
where \(\lambda\), \(\mu\) and \(s\) are dual variables. The optimal value of the dual problem is the distance between the two polytopes \(\mathcal{P}_1\) and \(\mathcal{P}_2\) and is constrained to be larger than minimum distance. Hence the constraint on dual problem optimal value is equivalent to the following feasibility problem \(\exists \lambda \geq 0, \mu \geq 0, s: -b_1^\top \lambda - b_2^\top \mu \geq d_{\text{min}}, A_1^\top \lambda + s = 0, A_2^\top \mu - s = 0, \|s\| \leq 1\). This reformulation can be substituted instead of collision avoidance constraint (11g) in the motion planning optimization problem (11). The same reformulation can be used for (11). Therefore, problem (11) can be rewritten as

\[
\min_{u^i(\cdot|t), \lambda_{ij}(\cdot|t), \mu_{ij}(\cdot|t), s_{ij}(\cdot|t)}
\]
subject to

\[
- b_i(z'(k|t))^\top \lambda_{ij}(k|t) ≥ d_{\text{min}},
- b_j(z'(k|t))^\top \mu_{ij}(k|t) ≥ d_{\text{min}},
\]
\[
\lambda_i(z'(k|t))^\top \lambda_{ij}(k|t) + s_{ij}(k|t) = 0,
\lambda_j(z'(k|t))^\top \mu_{ij}(k|t) + s_{ij}(k|t) = 0,
\]
\[
\|s_{ij}(k|t)\| \leq 1, -\lambda_{ij}(k|t) ≤ 0,
- \mu_{ij}(k|t) ≤ 0, \text{for all } i \in \mathcal{V}, j \in \mathcal{N}_i,
\]
where \(\lambda_i\) and \(\mu_i\) are functions of \(z'(k|t)\) and represent the polytopic set of \(i\)th vehicle at step \(k\) predicted at time \(t\). Similarly \(\lambda_j\) and \(\mu_j\) denote the polytopic set of \(j\)th vehicle.
which belongs to neighbor set $N_i$. The dual variables $\lambda_{ij}$, $\mu_{ij}$, and $s_{ij}$ are coupled through the collision avoidance constraint among vehicle $i$ and vehicle $j$. $\lambda_{ij}(\cdot | t)$, $\mu_{ij}(\cdot | t)$ and $s_{ij}(\cdot | t)$ represent the sequence of dual variables over the MPC horizon $N$. So $\lambda_{ij}(\cdot | t) = \{\lambda_{ij}(t), \ldots, \lambda_{ij}(t+N|t)\}$, $\mu_{ij}(\cdot | t) = \{\mu_{ij}(t), \ldots, \mu_{ij}(t+N|t)\}$ and $s_{ij}(\cdot | t) = \{s_{ij}(t), \ldots, s_{ij}(t+N|t)\}$. For simplicity the static obstacle avoidance constraint (II) is removed in the above formulation and only the collision avoidance among vehicles are formulated. However, it can be added using dual reformulation.

One main advantage of the proposed planning method is that the required minimum distance between the vehicles $d_{\text{min}}$, which can be chosen as a design parameter, is always enforced during the lane change maneuvers. In theory, the trajectories can be obtained for zero $d_{\text{min}}$, which means the polytopic sets (cars) can move on each other boundaries. In practice, $d_{\text{min}}$ should be determined based on the quantification of uncertainty of physical models and stochastic measurement errors, which is one future extension of this work.

V. FORMATION AS SEQUENCE OF MOTION PRIMITIVES

An alternative approach for the proposed optimization-based motion planning is behavior-based planning. In this section, we review a behavior-based planning using sequence of motion primitives [14] which we will use to benchmark our proposed optimization-based planning against. As described in Section [I] among all the existing methods, the behavior-based approach which uses a sequence of motion primitives is more suitable for formation of multi-lane platoons. In robotics applications, a complex dynamical task is achieved by synthesizing a sequence of motion primitives. In a similar way, achieving the desired platoon formation requires that a sequence of motion primitives be performed by each single vehicle in the platoon. We consider this method as a baseline and will compare our proposed optimization-based motion planning with this behavior-based method using a simple example scenario in Section [VI] and will discuss the advantages of our approach compared to this baseline.

For each motion primitive a number of parameters have to chosen. The examples of parameterized motion primitives for a single car in multi-lane formation are

- slow down: parameterized by desired speed and desired deceleration,
- cruise control (CC): parameterized by desired speed
- lane change: parameterized by lane index, desired acceleration or deceleration,
- adaptive cruise control (ACC): parameterized by the front’s car velocity and the desired inter-vehicle distance.

Planning sequence of motion primitives for each vehicle in the platoon to achieve a certain formation is hard to formulate and analyze mathematically. In this method, the system of vehicles is modeled as a hybrid system with various motion primitives as discrete modes and the transition maneuvers between them as continuous dynamics. To plan a sequence of motion primitives we have to solve a mixed-integer program (MIP), where different types of motion primitives are integer decision variables and the vehicles’ states are the continuous decision variables. However, MIPs are in general difficult to solve. An alternative common approach is to obtain the sequence of motion primitives according to a rule-based approach and then execute each motion primitives using the individual controllers for each primitives. Since the study of behavior-based approach is not the focus of this paper, we simplify the problem and assume the sequence of motion primitives for each vehicle is already determined based on some rules. Given the sequence of primitives, we design controllers to execute them. We design all the controllers, using MPC scheme such that the reference tracking cost is minimized while respecting vehicle dynamics and input and state limits. To keep the brevity of the paper, the controllers’ mathematical formulations are not discussed here, but detailed description can be found in our previous works. For example, we designed an MPC cruise controller (CC) discussed in [19] to execute following a desired velocity. Also, we designed an adaptive cruise control (ACC) to maintain a proper distance from the front car and follow the front car’s velocity, using the MPC formulation described in [20]. "Lane change" is achieved by changing the center of lanes as reference. In Section [VI] we will use these controllers to execute the given motion primitives for a simple example scenario for multi-vehicle formation.

VI. NUMERICAL RESULTS

We conducted three simulation scenarios to verify the effectiveness of the proposed motion planning algorithm. The simulations are conducted in MATLAB, the optimization problem is modeled using YALMIP and the nonlinear optimization is solved using IPOPT. The results are reported for three cases: a) platoon formation and re-configuration, b) obstacle avoidance, and c) comparison with behavior-based approach. The vehicle dimensions are chosen as 4.5m length and 1.8m width. The road width is chosen as 3.7m, which is the highway lane width standard at the United States. The control input limits are chosen as realistic physical limits of actual passenger vehicle. The acceleration input lower and upper bounds are chosen as $-4m/s^2$ and $4m/s^2$, respectively and its change is limited to $-1m/s^2$ and $1m/s^2$. The steering input lower and upper bounds are chosen as $-0.3rad$ and $0.3rad$ and its change is limited to $0.2rad/s$. At each iteration the optimization problem (I3) is solved and the first control input is applied to the vehicle kinematic model (I) for all the vehicles. Then the horizon is shifted and same procedure is repeated for the next step. For all the three scenarios the simulation results are presented as top view snapshots, as well as a series of state and action plots. The vehicles colors of the snapshots and plots are matched. The video for formation reconfiguration and obstacle avoidance scenarios is available online at this link [https://github.com/RoyaFiroozi/Centralized-Planning](https://github.com/RoyaFiroozi/Centralized-Planning).
A. Platoon Re-Configuration

In this scenario the platoon formation is alternating between two different configurations, as seen in Fig. 6. The platoon of four vehicles is moving in a two-dimensional configuration. The vehicles are moving in three different lanes and the platoon reshape into one-dimensional configuration and all the vehicles merge into one lane. The initial configuration is $C_I(2,[1,1,1],p_I)$, with $p_I = [0,5.5;6.0;-4.5,0]$ (matrix rows are separated by semicolons). The final configuration is $C_F([0,1,0],p_F)$, with $p_F = [0,0,0;0,0,0,0,0,0,0,0]$. The initial longitudinal coordinates for all the four vehicles are $[x^1(0),x^2(0),x^3(0),x^4(0)] = [10.5,4.5,0.5,15]$ and the initial lateral coordinates are $[y^1(0),y^2(0),y^3(0),y^4(0)] = [1.85,5.55,1.85,9.25]$. $d_{\text{min}}$ is chosen as 0.3m, the horizon $N$ is 5, sampling time $\Delta t$ is 0.2s, simulation time $T$ is 120, $\rho$ is 0.25 and $v_{\text{max}}$ is 20m/s. Fig. 7 represents the vehicles’ states and actions. The plots show the transient behavior between the two modes or configurations. The longitudinal and lateral coordinates $x$ and $y$, as well as heading angle $\psi$ and velocity $v$ for all the vehicles are shown in different colors which are matched with the colors in Fig. 6. The control actions $a$ and $\delta$ are also illustrated for all the vehicles. As seen the platoon reaches its steady state at final configuration after about 25 seconds.

B. Obstacle Avoidance

In obstacle avoidance scenario multiple vehicles are traveling together in a multi-lane platoon formation and once an obstacle is detected in the left lane, it triggers the transition to another configuration and the decision-maker selects single-lane configuration in the right lane to avoid the obstacle. The vehicles in the other lane make enough gap to facilitate safe and smooth lane changing and merging for the vehicles in the lane with obstacle. Fig. 7a shows the top view snapshots for obstacle avoidance simulation. The red vehicle has to change lane because a static obstacle (black object) has been detected on its lane. The yellow and blue vehicles make gap for the red vehicle to merge into their lane. The obstacle is modeled as a polytopic set and the obstacle avoidance constraints are introduced. The initial longitudinal coordinates for all the three vehicles are $[x^1(0),x^2(0),x^3(0)] = [10.5,4.5,0.5]$ and the initial lateral coordinates are $[y^1(0),y^2(0),y^3(0)] = [1.85,5.55,1.85]$. $d_{\text{min}}$ is chosen as 0.2m, the horizon $N$ is 8, sampling time $\Delta t$ is 0.1s, simulation time $T$ is 100, $\rho$ is 0.25 and $v_{\text{max}}$ is 10.5m/s. The vehicles’ states and actions are shown for in Fig. 7b. As seen, the steady state is achieved and 1D platoon is formed after about 10 seconds.

C. Comparison with Behavior-Based Approach

To compare our proposed approach with the behavior-based approach discussed in Section V. We consider a simple example scenario that two vehicles, which are moving together in the same lane, make enough gap for the third vehicle to allow it to merge into their lane. We chose this simple scenario to be able to determine the sequence of motion primitives for each vehicle intuitively without any mathematical analysis. However, sequence of motion primitives should be obtained using mathematical analysis such as MIP for more complicated scenarios. The simulation results for behavior-based approach is shown in Fig. 8a. The sequence of motion primitives for this simulation are:

1) Red car follows a constant desired velocity (CC).
2) Yellow car slows down.
3) Blue car performs lane-change.
4) Blue car follows the red car (ACC).
5) Yellow car follows the blue car (ACC).

As seen in Fig. 8a, at step (1), the cars are moving to the right in two-dimensional platoon and the yellow car slows down to make a proper gap to allow the blue car to merge into the lane, while the red car is moving with constant speed. Step (2) shows the lane change of the blue car. Step (3) illustrates the reconfiguration of one-dimensional platoon. In this simulation, the collision avoidance constraints among the cars are not imposed, so the blue car changes its lane only after a large enough gap is created between the red and yellow cars. Even for this simple scenario obtaining maneuvers with larger velocity and closer inter-vehicle distance was impossible after running extensive simulations. We replicated the same scenario with optimization-based planning. The same initial conditions and parameters are used for both methods. The initial longitudinal coordinates for the three vehicles are $[x^1(0),x^2(0),x^3(0)] = [6,12,0.5]$ and the initial lateral coordinates are $[y^1(0),y^2(0),y^3(0)] = [1.85,5.55,5.55]$. $d_{\text{min}}$ is chosen as 0.2m, the horizon $N$ is 8, sampling time $\Delta t$ is 0.1s, simulation time $T$ is 150, $\rho$ is 0.25 and $v_{\text{max}}$ is 10.5m/s. The resulting maneuvers obtained by motion primitive and optimization-based approaches are presented in Fig. 8b and Fig. 8c respectively. As seen in Fig. 8b the $x$ plot, the yellow car longitudinal position is far behind the other two. Also in $v$ plot, the yellow car reduces its speed dramatically and the blue car is changing its speed. However in Fig. 8c that shows the obtained trajectories using optimization-based approach the cars maintain a tight inter vehicle distance as seen in the $x$ plot and the velocities and accelerations are changing smoothly.

In addition, for this example, despite our extensive tuning efforts, it was not possible to obtain trajectories at highway speed and tight inter-vehicle distance, using motion primitive approach. The reason is that this approach requires proper tuning of many parameters and switches as discussed earlier. However, the optimization-based approach yields trajectories with highway speed 30m/s and tight inter-vehicle distance 0.2m. The results are shown in Fig. 2. In summary, the motion primitive approach does not provably enforce the collision avoidance constraints. Furthermore, to design tight mini platoons at highway speed, the proposed optimization-based approach is simplified compared to motion primitives approach in which extensive tuning is required for all the switches and all the possible parameters.
Fig. 6: **Top:** Platoon reshapes from multi-lane configuration into single-lane configuration. Four vehicles moving in three different lanes merge in one lane. Step (1) shows four vehicles moving to the right in three different lanes. This 2D configuration represents a steady state of the platoon. Steps (2) to (5) show the merging maneuver and finally step (6) demonstrates 1D platoon configuration as another steady state of the platoon. **Bottom:** The vehicles’ states and actions in a merging maneuver are presented. The plots demonstrate the transient between two configuration (steady state). The colors of all the plots are matched with the color of the vehicles in top view snapshots.

VII. CONCLUSION

We proposed an architecture for autonomous navigation of multi-lane platoons on public roads. The architecture is composed of real-time decision-making, offline motion planning and real-time path following modules. The simulation results demonstrate that a platoon of vehicles can form geometrically flexible and reconfigurable shapes in tight environment while moving at highway speed. We showed in the case of sudden change in the environment, like appearing an obstacle or slow traffic in one lane, the multi-lane platoon of vehicles can perform collaborative maneuvers and change their configuration to merge into faster lanes. We compared our approach with behavior-based planning, in which the formation and reconfiguration is achieved by a sequence of motion primitives. We showed that to design tight maneuvers for mini-platoons at highway speed our proposed optimization-based method is simplified compared to the motion primitive approach, which requires extensive tuning for the switches and parameters. The future work will be robustification of the planning scheme by handling the uncertainty caused by model mismatch, sensor measurements and communication delays and using closed-loop policies instead of open-loop ones.

VIII. ACKNOWLEDGEMENT

The information, data, or work presented herein was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0000791. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

REFERENCES

[1] J. Guanetti, Y. Kim, and F. Borrelli, “Control of connected and automated vehicles: State of the art and future challenges,” *Annual Reviews in Control*, vol. 45, pp. 18 – 40, 2018.
[2] A. A. Alam, A. Gattami, and K. H. Johansson, “An experimental study on the fuel reduction potential of heavy duty vehicle platooning,” *13th International IEEE Conference on Intelligent Transportation Systems*, pp. 306–311, 2010.
[3] A. Alam, B. Besselink, V. Turri, J. Martensson, and K. H. Johansson, “Heavy-duty vehicle platooning for sustainable freight transportation: A cooperative method to enhance safety and efficiency,” *IEEE Control Systems Magazine*, vol. 35, no. 6, pp. 34–56, 12 2015.
Fig. 7: (a) A static obstacle (black object) is detected in the red vehicle lane, the yellow and blue vehicles make gap for the red to merge into their lane. Vehicles are travelling in 2D configuration are flexible and are able to reshape in case of presence of obstacle in one lane. (b) The plots correspond to the snapshots and represents the vehicles’ states and actions during the simulation. As seen, the steady state is achieved and 1D configuration is formed.

Fig. 8: (a) Formation using sequence of motion primitives is demonstrated. At step (1), the cars are moving to the right in two-dimensional formation and the yellow car starts slowing down to make enough gap for the blue car to merge, while the red car doesn’t change its speed. At step (2), the blue car changes lane to merge the platoon. At step (3) blue follows the red car and yellow follows the blue car and 1D platoon is formed. (b) Planning using sequence of motion primitives (c) Optimization-based planning.

Fig. 9: The trajectories obtained by optimization-based approach with highway speed and tight inter-vehicle distance are shown.

[4] X. Sun and Y. Yin, “Behaviorally stable vehicle platooning for energy savings,” Transportation Research Part C: Emerging Technologies, vol. 99, pp. 37 – 52, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X18312245

[5] J. K. Hedrick, D. McMahon, V. Narendra, and D. Swaroop, “Longitudinal vehicle controller design for ivhs systems,” in 1991 American Control Conference, June 1991, pp. 3107–3112.

[6] S. E. Shladover, C. A. Desoer, J. K. Hedrick, M. Tomizuka, J. Walrand, W. Zhang, D. H. McMahon, H. Peng, S. Sheikholeslam, and N. McKeown, “Automated vehicle control developments in the path program,” IEEE Transactions on Vehicular Technology, vol. 40, no. 1, pp. 114–130, Feb 1991.

[7] R. Rajamani, S. B. Choi, B. Law, J. K. Hedrick, R. Prohaska, and P. Kretz, “Design and experimental implementation of longitudinal control for a platoon of automated vehicles,” in Rajmanian, 2000.

[8] Han-Shue Tan, R. Rajamani, and Wei-Bin Zhang, “Demonstration of an automated highway platoon system,” in Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No.98CH36207), vol. 3, June 1998, pp. 1823–1827 vol.3.

[9] M. Guerita, R. Billot, N.-E. El Faouzi, J. Monteil, F. Armetta, and S. Hassas, “How to assess the benefits of connected vehicles? a simulation framework for the design of cooperative traffic management strategies,” Transportation Research Part C Emerging Technologies, vol. 67, 04 2016.

[10] A. Loria, J. Dasdemir, and N. A. Jarquin, “Leader-follower formation and tracking control of mobile robots along straight paths,” IEEE Transactions on Control Systems Technology, vol. 24, no. 2, pp. 727–732, 03 2016.

[11] J. Chu, Z. Qu, E. Pollak, and M. Falash, A New Multi-objective Control Design for Autonomous Vehicles. inbook, 10 2008, vol. 381, pp. 81–102.

[12] X. Qian, A. de La Fortelle, and F. Moutarde, “A hierarchical model predictive control framework for on-road formation control of autonomous vehicles,” in 10.1109/IVS.2016.7535413, 06 2016.

[13] C. W. Reynolds, “Flocks, herds and schools: A distributed behavioral model,” SIGGRAPH Comput. Graph., vol. 21, no. 4, pp. 25–34, Aug. 1987.

[14] T. Balch and R. C. Arkin, “Behavior-based formation control for multirobot teams,” IEEE Transactions on Robotics and Automation, vol. 14, no. 6, pp. 926–939, 12 1998.

[15] R. Olfati-Saber, “Flocking for multi-agent dynamic systems: algorithms and theory,” IEEE Transactions on Automatic Control, vol. 51, no. 3, pp. 401–420, 03 2006.

[16] N. E. Leonard and E. Fiorelli, “Virtual leaders, artificial potentials and coordinated control of groups,” in Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228), vol. 3, 12 2001, pp. 2968–2973 vol.3.