HAL
open science

A Distributed Model Predictive Control Framework for Road-Following Formation Control of Car-like Vehicles (Extended Version)

Xiangjun Qian, Florent Altché, Arnaud de La Fortelle, Fabien Moutarde

▶ **To cite this version:**

Xiangjun Qian, Florent Altché, Arnaud de La Fortelle, Fabien Moutarde. A Distributed Model Predictive Control Framework for Road-Following Formation Control of Car-like Vehicles (Extended Version). 2016. hal-01309659

HAL Id: hal-01309659

https://hal.science/hal-01309659v1

Preprint submitted on 29 Apr 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Distributed Model Predictive Control Framework for Road-Following Formation Control of Car-like Vehicles (Extended Version)

Xiangjun Qian[1], Florent Altché[2,1], Arnaud de La Fortelle[1], Fabien Moutarde[1]
[1]PSL - Research University, Centre for Robotics, 60 Bd St Michel
75006 Paris, France
[2]École des Ponts ParisTech, Cité Descartes, 6-8 Av Blaise Pascal
77455 Champs-sur-Marne, France
Email: xiangjun.qian@mines-paristech.fr

***Abstract*—This work presents a novel framework for the formation control of multiple autonomous ground vehicles in an on-road environment. Unique challenges of this problem lie in 1) the design of collision avoidance strategies with obstacles and with other vehicles in a highly structured environment, 2) dynamic reconfiguration of the formation to handle different task specifications. In this paper, we design a local MPC-based tracking controller for each individual vehicle to follow a reference trajectory while satisfying various constraints (kinematics and dynamics, collision avoidance, *etc.*). The reference trajectory of a vehicle is computed from its leader's trajectory, based on a pre-defined formation tree. We use logic rules to organize the collision avoidance behaviors of member vehicles. Moreover, we propose a methodology to safely reconfigure the formation on-the-fly. The proposed framework has been validated using high-fidelity simulations.**

I. Introduction

Over the past decade, we have witnessed some major developments of autonomous driving technologies. With the vision of mass deployment of autonomous vehicles on the roads, the problem of coordinating groups of autonomous vehicles has gained a lot of attention. Platoons, as a type of on-road coordination, have been extensively studied [1]–[3]. Vehicles in a platoon move along a road in a linear formation with small longitudinal gaps. The advantage of platoons lies mainly in a reduced fuel consumption due to lower air drag and an increased road throughput thanks to smaller inter-vehicular distance.

We consider an extended version of platoons in which multiple vehicles can enter a formation with both longitudinal and lateral separations. We expect that this extension can find its applications in cooperative tasks, for instance protecting a VIP vehicle, snowplowing (see example in [4]), cooperative lane change, *etc.*.

There has been a significant volume of work on generic robot formation control problems [5]–[9]. References [5], [6] propose a leader-following approach in which a leader follows a reference trajectory, while other robots maintain a relative position offset with the leader. Virtual structure based techniques [7], [8] consider the formation as a moving rigid structure, where robots locally track their reference trajectories based on the trajectory of the structure. Distributed Model Predictive Control (MPC) is a promising technique for the formation control problem thanks to its generality and reconfigurability. The theoretical foundation of applying distributed MPC on the multi-vehicle coordination problem has been laid in [10], [11]. It is shown in [10] that we can stabilize a formation by only formulating local MPC problems and exchanging previously calculated optimal trajectories, provided that the previously calculated trajectories do not deviate too much from the ones being actually followed. In [11], a hybrid system approach combining logical rules with MPC is proposed. This approach allows to regulate the interactions between vehicles in an organized and predictable way, thus avoiding unexpected formation behaviors and therefore improving stability and feasibility.

The literature in generic robot formation control can be a source of inspiration for the on-road formation control problem. However, unique challenges set the on-road formation problem apart from its generic counterpart. First, vehicles must intelligently balance the goal of formation keeping with obstacle/collision avoidance in a highly constrained on-road environment. Second, vehicles must be able to deliberatively reconfigure their formation to handle different task specifications.

Previous work exists on the road-following formation control problem. Reference [12] represents the formation as a virtual structure and uses a hierarchical Model Predictive Control framework to maintain the formation. However, the intra-formation collision avoidance is not considered and the virtual structure is static. In [13], a hybrid two-layer framework is proposed, in which an outer layer working in discrete modes safely coordinates the vehicles to find collision-free trajectories, while an inner layer is tasked with trajectory tracking. Moreover, the formation is determined in advance and cannot be reconfigured. Other approaches include Laplacian-based control [14], graph searching methods [4], *etc.*.

This paper proposes and validates a novel on-road formation control framework established on the theoretical results of distributed MPC, upon which we propose multiple adaptations

to handle the challenges raised by the on-road driving setting. We adopt a fully distributed paradigm in which each vehicle is equipped with a MPC tracking controller. The module computes a sequence of control inputs to track a reference trajectory while satisfying various constraints (kinematics and dynamics, collision avoidance, *etc.*). The reference trajectory of a vehicle is computed from its leader's trajectory, based on a pre-defined formation tree. The major contribution of this paper is the use of hybrid system techniques to organize the collision avoidance behaviors of member vehicles. To the best of our knowledge, it is the first time that such design is proposed in this problem setting.

The hybrid approach allows to combine the optimization part of the MPC technique with logic rules that relieve the optimization process from lengthy collision constraints computations. Moreover, logic rules can be made semantically meaningful, which could lead to easier interactions between automated vehicles and people (be it other drivers or pedestrians, cyclists, *etc.*). Finally, our framework allows to safely modify the formation on-the-fly. In this article, we present preliminary results validated by simulation, demonstrating this framework is suitable for actual implementation.

The rest of the paper is organized as follows. In Section II, we present the MPC-based trajectory generation for vehicles. Section III presents the modeling of the on-road formation control problem. Section IV shows how this coordination framework can be used for dynamic formation changes. Section V validates our approach through simulations, and Section VI concludes the the paper.

II. MPC-based Trajectory Generation for Individual Vehicles

A. Road-following Coordinate System

We first define a road-following coordinate system. We assume that the centerline of the road can be described as a curve $\Gamma \in \mathbb{R}^2$ with C^3 continuity, ensuring that the curvature c of the curve and its derivative κ exist. We define a Frenet coordinate system (s, r), where s is the curvilinear abscissa along Γ, and r the lateral deviation. The left and right boundaries of the road are defined as continuous functions $\bar{r}(s)$ and $\underline{r}(s)$. To ensure the bijection from the (s, r) frame to a Cartesian frame, we require that for all (s, r) such that $\underline{r}(s) \leq r \leq \bar{r}(s)$:

$$1 - rc(s) < 0, \tag{1}$$

where $c(s)$ is the curvature profile of the road's centerline at point s.

B. Kinematic Bicycle Model in the Frenet Frame

We adopt the common practice in vehicle control to set up a two-layer control structure, in which an inner-loop controller stabilizes the system and a outer-loop planning module plans the trajectory according to the goal configuration. If the inner-loop controller is fast enough to track the given reference trajectories, the outer-loop dynamics can be approximately described as a standard kinematic bicycle model [15].

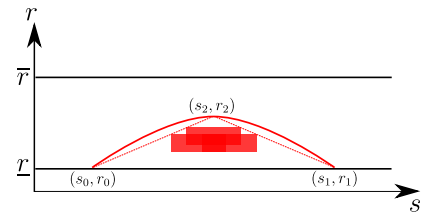


Fig. 1: Approximation of an obstacle region by a parabola.

As we consider on-road autonomous driving, we formulate the kinematic bicycle model in the road-following coordinate system as follows:

$$\dot{s} = v \cos\theta \left(\frac{1}{1 - rc(s)} \right), \tag{2a}$$

$$\dot{r} = v \sin\theta, \tag{2b}$$

$$\dot{\theta} = vk - v\cos\theta \left(\frac{c(s)}{1 - rc(s)} \right), \tag{2c}$$

$$\dot{v} = a, \tag{2d}$$

$$\dot{k} = \kappa. \tag{2e}$$

$c(s)$ is assumed to be known, for instance from maps. Other state variables r, v, θ and k are respectively the lateral offset, the vehicle speed, the heading alignment error of the vehicle with regard to the centerline and the curvature of the vehicle's trajectory. The chosen control inputs are the acceleration a and the first derivative of curvature, κ. We note $\boldsymbol{x} = [s, r, v, \theta, k] \in \boldsymbol{X}$ the state of the vehicle, and $\boldsymbol{u} = [a, \kappa] \in \boldsymbol{U}$ its control. $\boldsymbol{X}$ and $\boldsymbol{U}$ are compactly written form of the state constraint and the control constraint. We note $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$ the system (2a)-(2e). When there is a need to differentiate between different vehicles, subscripts like i, j can be added to those variables.

More details on the transformation of the kinematic bicycle model from a Cartesian frame to the Frenet frame can be found in [16].

C. Obstacle Modeling

Vehicle trajectories are generated using a receding horizon scheme based on optimization techniques. However, the convergence to a global optimum cannot be guaranteed. With one obstacle in the middle of the road, there are two homotopy classes of trajectories, respectively corresponding to avoiding the obstacle from the left and from the right. This situation leads to multiple stationary points for the optimization problem. In this paper, we heuristically address the problem by affecting obstacles to their closest side of the road, thus eliminating one homotopy class. A more detailed discussion on this issue can be found in [17].

Another issue in obstacle modeling is that the optimization-based scheme requires obstacle boundaries to be differentiable so that traditional gradient-based methods can be deployed; in general, road obstacles do not display such properties. To overcome this issue, we approximate obstacles by bounding triangles, and we use the unique parabola going through the three points of the triangle to get smooth obstacle boundaries, as represented in Fig. 1. Note that the parameters of this

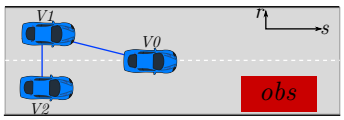


Fig. 2: A three vehicle formation with an obstacle on the road.

parabola can be found quickly by solving a system of three linear equations. In what follows, we let $\mathcal{O}$ be the set of all obstacles and, for an obstacle $o \in \mathcal{O}$ we define $h_o(\boldsymbol{x})$ such that $h_o(\boldsymbol{x}) \leq 0$ if, and only if, the vehicle is outside the concavity of the parabola corresponding to o at time t.

D. MPC formulation

The control objective for each vehicle is to track its reference trajectory. Therefore, we consider the following cost function that penalizes the aggregated deviation from the reference trajectory over the planning horizon. More specifically, the cost function is given as:

$$\begin{aligned}\mathcal{J}(\boldsymbol{x},\boldsymbol{u}) &= ||\boldsymbol{x}(T)-\boldsymbol{x}_{ref}(T)||_P^2 \\ &+\int_{t_0}^{t_0+T}\left(||\boldsymbol{x}(t)-\boldsymbol{x}_{ref}(t)||_Q^2+||\boldsymbol{u}(t)||_R^2\right)\end{aligned} \tag{3}$$

where t_0 is the current time instant and T is the prediction horizon. P, Q and R are positive diagonal weighting matrices of respective dimensions 5, 5 and 2. The running cost term $\int_{t_0}^{t_0+T}(||\cdot||_Q^2+||\cdot||_R^2)$ penalizes the deviation from the desired trajectory, as well as the control effort. The terminal cost term $||\cdot||_P^2$ is set to 0 in this paper. We will further discuss the construction of the reference trajectory $\boldsymbol{x}_{ref}$ in the following section, as it is the major enabler of our formation control framework.

At each sampling time $t = t_0$, we solve the following constrained optimization problem:

$$\min_{\boldsymbol{u}} \mathcal{J}(\boldsymbol{x},\boldsymbol{u}), \tag{4}$$

subj. to $\forall t \in [t_0, t_0+T]$,

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t),\boldsymbol{u}(t)), \tag{5a}$$

$$\boldsymbol{x}(t) \in \boldsymbol{X} \quad \text{and} \quad \boldsymbol{u}(t) \in \boldsymbol{U}, \tag{5b}$$

$$-\bar{a}_{i,lat} \leq v(t)^2 k(t) \leq \bar{a}_{i,lat}, \tag{5c}$$

$$h_o(\boldsymbol{x}(t)) \leq 0, \quad \forall o \in \mathcal{O}. \tag{5d}$$

Constraint (5a) is the compactly-written form of the vehicle model. Eq. (5b) captures various bound constraints on speed, lateral deviation, acceleration, *etc*. Eq. (5c) limits the lateral acceleration of the vehicle and Eq. (5d) ensures the obstacle avoidance constraints for all nearby obstacles, using the method described in Section II-C.

The resulting optimal trajectory is then (periodically) fed to a low level controller for tracking.

III. Cooperative Formation control

This section discusses the formation-control scheme that allows vehicles to maintain a convoy formation while avoiding obstacles and other vehicles.

A. Formation Model

Consider a set $\mathcal{N} = \{0, .., N\}$ of $N+1$ vehicles, in which we arbitrarily use 0 as leader; the leader can either be a physical vehicle, or a virtual reference point. We define the target shape of the formation through a $(N+1)\times 2$ matrix $\mathcal{S}$ in which each row vector (s_i^d, r_i^d) encodes the target position of vehicle i relatively to the leader. To describe the interdependence relations between vehicles in the formation, we define a formation tree as a directed tree $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, whose nodes are the vehicles of $\mathcal{N}$ and with a set of edges $\mathcal{E}$ such that the root node is the leader 0. Such a tree can be represented as an adjacency matrix (g_{ij}) of size $(N+1)\times(N+1)$, in which element g_{ij} equals 1 if the edge $i \to j$ is in $\mathcal{E}$, and 0 otherwise. Remark that we expect $g_{ij} = 1 \Rightarrow s_i^d \geq s_j^d$ so that a given vehicle needs only take information from the preceding vehicles, although it is not mandatory in our formulation.

Example 1. Consider a triangular formation of three vehicles as shown in Fig. 2. The shape $\mathcal{S}$ of the formation is defined as

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{cc} s_i^d & r_i^d \end{array} \\ \begin{pmatrix} 0 & 0 \\ -10 & 3 \\ -10 & -3 \end{pmatrix} \end{array}$$

The formation tree $\mathcal{G}$ defined by the adjacency matrix g is given as:

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$$

such that vehicle 1 computes its formation control trajectory relative to vehicle 0, and vehicle 2 computes its trajectory relative to vehicle 1.

B. Formation Control Scheme

Vehicles in the formation are assumed to use the MPC-based trajectory generation scheme described in Section II. We propose to adjust the reference trajectories of vehicles to drive them into the desired formation.

Consider an arbitrary pair of vehicles (i, j) such that $g_{ij} = 1$. A communication link can be established between i, j such that vehicle j periodically receives information on the planned trajectories of vehicle i. Assume that at time t the most up-to-date trajectory for vehicle i received by vehicle j is $\boldsymbol{x}_i([t_1, t_1+T_p])$, with $t_1 < t$: the trajectory $\boldsymbol{x}_i([t, t+T_p])$ can be simply estimated by simulating the trajectory from t_1+T_p to $t+T_p$ using the last value of the control. Under the assumption that the communication delay is small, we expect the estimated trajectory $\boldsymbol{x}_i([t, t+T_p])$ to remain close to the actually planned trajectory of vehicle i.

The desired relative position of j from i can then be calculated as

$$s_{ji}^d = s_j^d - s_i^d, \tag{6a}$$

$$r_{ji}^d = r_j^d - r_i^d. \tag{6b}$$

Finally, the reference trajectory $\boldsymbol{x}_{j,ref}$ can be obtained by offsetting the position components of $\boldsymbol{x}_i([t, t+T_p])$ with s_{ij}^d and r_{ij}^d and setting other components to zero.

The proposed approach allows a distributed implementation of the formation control scheme. Only a small amount of information related to the formation definition is synchronized among vehicles, while each vehicle uses local information to plan its own trajectory.

C. Intra-formation Collision Avoidance

Our aim is to design an intra-formation collision avoidance strategy such that vehicles will not collide in normal on-road driving situations, nor in situations where the formation is perturbed by obstacles. Remark that the exact rectangle-shaped safety region of a vehicle is non-differentiable, and multiple choices exist for collision avoidance. Taking a 2-vehicle scenario as an example (vehicle 1 and 2 in Fig. 2), if vehicle 1 approaches vehicle 2, vehicle 2 has four strategies to avoid collision: acceleration to the front of vehicle 1, deceleration behind vehicle 1, moving further to the right or circumvent vehicle 1 from its left. The problem becomes exponentially complex as the number of vehicles increases due to its combinatorial nature. In [11], a similar problem is handled through the formulation of a mixed integer problem, in which the avoidance decisions are modeled as binary variables. However, the real-time requirement of our algorithm stops us from adopting the same method.

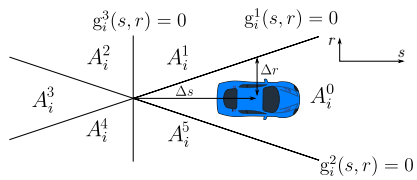


Fig. 3: Road space partitioning with respect to vehicle i.

Here, we adopt a hybrid modeling approach. Let us define an ordered priority list $\mathcal{L}$ as a permutation of $\{0, 1, ..., N\}$, representing the relative priorities between vehicles. With a slight abuse of notation, we let $\mathcal{L}(i)$ be the rank of i in $\mathcal{L}$, *i.e.* $\mathcal{L}(i) = j$ if, and only if, i is the j-th element of $\mathcal{L}$. We enforce the following constraint for any pair of vehicles $i, j \in \mathcal{N}$:

Design constraint: $\mathcal{L}(i) < \mathcal{L}(j) \iff s_i^d \geq s_j^d$

Under this constraint, the list $\mathcal{L}$ encodes a traffic rule for the formation of vehicles: each vehicle is only responsible for avoiding collisions with the higher-priority vehicles on the road, *i.e* with vehicles preceding it in $\mathcal{L}$.

To account for the existence of different homotopy classes, we divide the space outside the collision region in six areas: for a vehicle i, we partition the space using three affine functions $g_i^1(s,r) = 0$, $g_i^2(s,r) = 0$ and $g_i^3(s,r) = 0$ as shown in Fig. 3:

$$g_i^1(s,r) = -\frac{r - r_i}{\Delta r} + \frac{s - s_i}{\Delta s} + 1, \tag{7a}$$

$$g_i^2(s,r) = \frac{r - r_i}{\Delta r} + \frac{s - s_i}{\Delta s} - 1, \tag{7b}$$

$$g_i^3(s,r) = \frac{s - s_i}{\Delta s} + 1. \tag{7c}$$

where Δs and Δr are geometric parameters described in Fig. 3. For each vehicle i, these affine functions define a partition of the (s, r) plane in six sub-spaces labeled as $A_i^m, m \in \{0..5\}$ as shown in Fig. 3. By definition, vehicle i is always inside region A_i^0, and we enforce safety by preventing vehicles with lower priority than i from entering A_i^0. In what follows, we call A_i^0 the *protected region* for i, and the union of subsets $A_i^m, m \in \{1 \dots 5\}$ forms the *safe region* with respect to i.

For an arbitrary vehicle j, the following logic rule is introduced for all $i \in \mathcal{N}$:

$$\mathcal{L}(i) < \mathcal{L}(j) \Rightarrow (s_j, r_j) \in \bigcup_{m=\{1..5\}} A_i^m. \tag{8}$$

Rule (8) effectively forces each vehicle to remain in the safe region with respect to the vehicles having higher priority in $\mathcal{L}$. In a classic MPC setting, a decision variable would be required to select in which of the subspaces $A_i^m, m \in \{1 \dots 5\}$ vehicle j should be. In this article we adopt a different, simpler approach consisting of using the shape matrix $\mathcal{S}$ to implicitly constrain the choice of the homotopy class as follows: $\forall i \in \mathcal{N}$,

$$\mathcal{L}(i) < \mathcal{L}(j) \wedge s_{ji}^d \geq -\Delta s \wedge r_{ji}^d > 0 \Rightarrow g_i^1(s_j, r_j) \leq 0, \tag{9a}$$

$$\mathcal{L}(i) < \mathcal{L}(j) \wedge s_{ji}^d \geq -\Delta s \wedge r_{ji}^d < 0 \Rightarrow g_i^2(s_j, r_j) \leq 0, \tag{9b}$$

$$\mathcal{L}(i) < \mathcal{L}(j) \wedge s_{ji}^d < -\Delta s \Rightarrow g_i^3(s_j, r_j) \leq 0. \tag{9c}$$

Note that, although the space represented by $\bigcup_{m=\{1...5\}} A_i^m$ is non-differentiable, the above constraints are linear (and therefore differentiable). As a trade-off, the equations in (9) are more restrictive than rule (8). Constraints (9) are then incorporated into the MPC formulation (4)-(5) for enforcement. Also note that, in the above constraints, the shape matrix and value of Δs are parameters and not variables; therefore, only one of the three above constraint is actually added to the problem at each step.

Example 2. Let $\mathcal{L} = \{0, 1, 2\}$ be the priority list for the formation illustrated in Fig. 2. The constraint for vehicle 1 can be computed from (9) as $g_0^3(s_1, r_1) \leq 0$ such that the safe region (relative to vehicle 0) is $A_0^2 \cup A_0^3 \cup A_0^4$, forcing vehicle 1 to stay behind vehicle 0. The constraint for vehicle 2 is $g_0^3(s_2, r_2) \leq 0 \wedge g_1^2(s_2, r_2) \leq 0$, such that vehicle 2's safe region is $(A_0^2 \cup A_0^3 \cup A_0^4) \cap (A_1^3 \cup A_1^4 \cup A_1^5)$: vehicle 2 must stay behind vehicle 0, and stay on the right-hand or behind vehicle 1.

D. Discussions on stability and feasibility

Here we briefly discuss the stability and feasibility issues of the proposed framework.

This framework is an application and implementation of the more theoretical result of [10], where a general distributed model predictive control of a dynamically decoupled multi-agent system is considered. It is shown that stability of the formation can be ensured if, for each agent, the difference between trajectories resulting from two consecutive MPC optimizations is smaller than $\delta^2 q$ (which is called the *compatibility constraint*), where δ is the update interval and q is a properly chosen constant. The philosophy is that, provided each agent remains sufficiently close to the trajectory other agents base their own trajectory on, the global system remains stable.

It is also shown in their simulation that even without the constraint, the formation is stable thanks to the inherently consistent nature of MPC (although the theoretical guarantee is lost). On the other hand, adding compatibility constraints may introduce sluggish behaviors as agents cannot react quickly to unexpected changes in the environment. One condition that requires the compatibility constraint is when multiple maneuver choices (local optima) exist in the MPC formulation. In this case, the trajectories obtained from successive optimizations can be largely different as the solver may switch between these different local optima.

In our framework, we opt to not use the compatibility constraint. The hybrid modeling of intra-formation collision avoidance (9) suppresses the possibility of multiple local optima because the rules define a unique homotopy class of trajectories. The resulting formation is experimentally shown to be stable.

Another issue is the feasibility of local MPC during iterations. Consider the case of vehicle 1 in Fig. 2 suddenly turning right towards vehicle 2; the collision constraint can be violated due to the maneuver limits of vehicle 2 and the delay in communication. A possible solution is to use the invariant set theory [18] so that vehicle 2 is required to guard against all possible maneuvers of vehicle 1 during the MPC iteration. However, this method makes the MPC more conservative and requires more computational power. Here, we choose to design the vehicle formation to be sufficiently loose and transform the safety constraint (using g_i^1 as an example) to be a soft constraint as

$$g_i^1(s_j, r_j) \leq \varepsilon, \qquad \varepsilon \geq 0. \tag{10}$$

The occasional constraint violation is therefore permitted, but will be penalized by a quadratic term $K\varepsilon^2$ added to (4).

IV. Dynamic Formation Reconfiguration

Let $\mathcal{G}$ be a formation tree, $\mathcal{S}$ a shape and $\mathcal{L}$ a priority list, all compatible: we denote by $\mathcal{H} = (\mathcal{G}, \mathcal{S}, \mathcal{L})$ the corresponding formation. We give the following definition:

Definition 1 (Isormorphic formation)**.** Two formations $\mathcal{H}_1$, $\mathcal{H}_2$ are said to be isomorphic if $\mathcal{L}_1 = \mathcal{L}_2$.

In this paper, we only consider isomorphic formation changes: the reconfiguration of $\mathcal{S}$ and/or $\mathcal{G}$. Non-isomorphic formation changes that involve the modification of $\mathcal{L}$ are a topic for future research.

The reconfiguration of $\mathcal{G}$ only affects the behaviors of vehicles when the formation is perturbed. Thus we are free to modify $\mathcal{G}$ as long as the tree structure covers all nodes. On the other hand, we cannot arbitrarily modify the shape matrix $\mathcal{S}$ due to the design of intra-formation collision avoidance constraint (9). Consider the case of two vehicles i, j with $\mathcal{L}(i) < \mathcal{L}(j)$, where vehicle j is in the partition A_i^5 of vehicle i. Assuming that we reconfigure the shape matrix, if the new goal configuration of vehicle j resides in A_i^4, then vehicle j can plan a trajectory to its goal configuration because both A_i^4 and A_i^5 belong to the constraint $g_i^2(s, r) \leq 0$. However, if the goal configuration is in A_i^2, vehicle i cannot plan a feasible trajectory under the current formulation as A_i^2 is not a feasible region under the current constraint g_1^2. It is necessary to set an intermediate goal configuration at A_i^3. Once vehicle j reaches A_i^3, the collision avoidance constraint must be switched from g_i^2 to g_i^3 and then vehicle j can continue to move towards its goal configuration. This issue is a side effect of the space partitioning technique.

We propose the following definitions:

Definition 2 (1-step Reachable Element)**.** Consider two elements of the partition A_i^m and A_i^n; we say A_i^n is 1-step reachable from A_i^m if there exists $l \in \{1, 2, 3\}$ such that $g_i^l \leq 0$ over $A_i^n \cup A_i^m$.

This means we need at least one of the rules (9) to be kept during reconfiguration. We can see, A_i^1 is reachable from A_i^2 and A_i^3; A_i^2 is reachable from A_i^1, A_i^3 and A_i^4; A_i^3 is reachable from all elements except A_i^0; and symmetrically for the other areas.

Definition 3 (1-step Reachable Formation)**.** Consider two formations $\mathcal{H}_1$ and $\mathcal{H}_2$. We say that $\mathcal{H}_2$ is 1-step reachable from $\mathcal{H}_1$ if $\forall i, j \in \mathcal{N}$ s.t. $\mathcal{L}(i) < \mathcal{L}(j)$, the configuration of vehicle j with respect to vehicle i in $\mathcal{H}_2$ is 1-step reachable from $\mathcal{H}_1$.

Theorem 1. Consider an arbitrary pair of isomorphic formations $\mathcal{H}_1$, $\mathcal{H}_2$, we can transform from $\mathcal{H}_1$ to $\mathcal{H}_2$ through a finite sequence of 1-step reachable formations.

The proof is intuitive, since A_i^3 is reachable from all elements except A_i^0. Therefore, any formation can be transformed to a linear formation in a finite number of steps.

With the above theoretical result, in order to reconfigure the formation to the desired one, we only need to design an intermediate sequence of 1-step reachable formations. Then we can set up a discrete supervisor to control the transformation.

Example 3. We consider a formation of four vehicles. Fig. 6 illustrates a sequence of 1-step reachable isomorphic transformations. The corresponding shape matrices are given as follows.

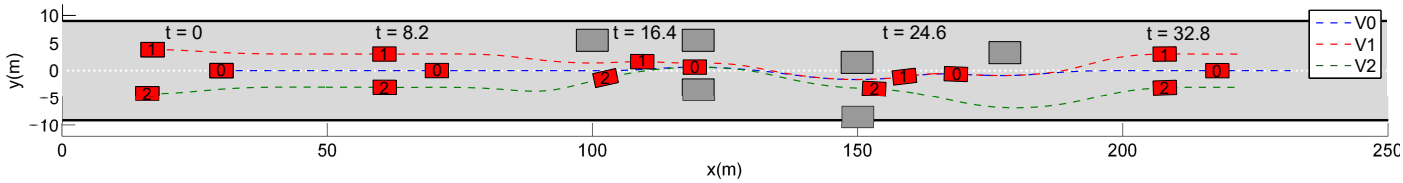
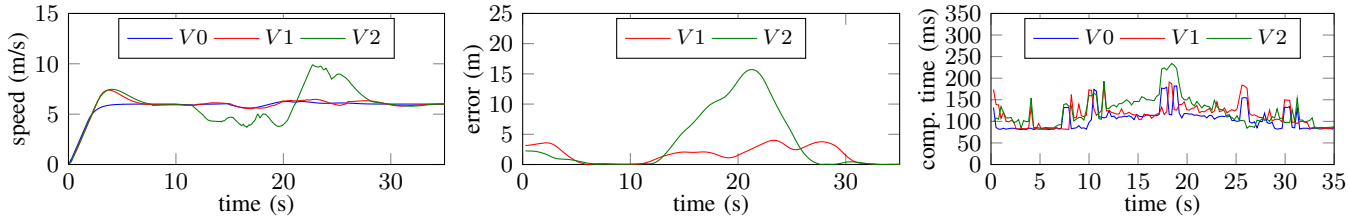


Fig. 4: First scenario: trajectories of three vehicles.



(a) (b) (c)

Fig. 5: First scenario: (a) vehicle speeds, (b) formation error, (c) computation time.

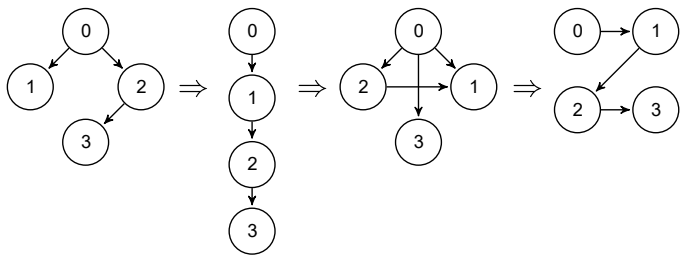


Fig. 6: A sequence of 1-step reachable isomorphic transformations.

$$\mathcal{S}_1 \begin{pmatrix} 0 & 0 \\ -10 & 3 \\ -10 & -3 \\ -20 & 0 \end{pmatrix} \quad \mathcal{S}_2 \begin{pmatrix} 0 & 0 \\ -10 & 0 \\ -20 & 0 \\ -30 & 0 \end{pmatrix} \quad \mathcal{S}_3 \begin{pmatrix} 0 & 0 \\ -10 & -3 \\ -10 & 3 \\ -20 & 0 \end{pmatrix} \quad \mathcal{S}_4 \begin{pmatrix} 0 & 3 \\ 0 & -3 \\ -10 & 3 \\ -10 & -3 \end{pmatrix}$$

V. Experiments and Results

We have implemented our framework in the high-fidelity robotic simulator Webots [19]. Webots implements a realistic vehicle model with steering dynamic response and non-linear tire models. The proposed algorithm is implemented in C++ and we use the ACADO toolkit [20] to solve the MPC problem.

We have designed two test scenarios. In both scenarios, vehicles are equipped with noise-free localization systems and delay-free communication devices. The desired speed of the formation for both scenarios is given as $v_{\mathcal{H}} = 6\,\mathrm{m/s}$. Vehicle parameters are given as: $0 \leq v_i \leq 10\,\mathrm{m/s}$, $|a_i| \leq 2.5\,\mathrm{m/s^2}$, $|k_i| \leq 0.2\,\mathrm{m^{-1}}$, $|\kappa_i| \leq 0.1\,\mathrm{m^{-1}s^{-1}}$ and $\bar{a}_{i,lat} = 2.5\,\mathrm{m/s^2}$. The parameters used for the leader are $Q_0 = diag(0, 4, 2, 20, 20)$, $R_0 = diag(1, 200)$. The parameters used for the followers are $Q_i = diag(1, 2, 0, 20, 20)$, $R_i = diag(1, 200)$. The constraint violation penalty parameter described in Section III-D is selected as $K = 10000$. The prediction horizon for all vehicles is $T_i = 5\,\mathrm{s}$. The trajectory re-planning interval is $\Delta T_i = 0.256\,\mathrm{s}$. Thus at time t_0, a follower has access to the planned trajectory of its leader at time $t_0 - 0.256$. The parameters for space partition are given as $\Delta s = 10\,\mathrm{m}$ and $\Delta r = 3\,\mathrm{m}$. To quantify the formation error, we introduce e_i as the combination of the longitudinal formation error and the lateral formation error: $e_i = \sqrt{(e_i^s)^2 + (e_i^r)^2}$.

The first scenario involves three vehicles in a triangle-shaped formation (Fig. 2) which can be used, for instance, for a snowplowing application [4]. The desired formation remains static during the entire simulation, while vehicles must avoid on-road obstacles and cross narrow corridors. Fig. 4 shows the computed trajectories of three vehicles using our framework. We remark that vehicles are able to quickly form the desired formation, and maintain it in the absence of obstacles. In the vicinity of obstacles (at the 100-meters mark), vehicles are able to swerve around them and even temporarily deform the formation to pass. Fig. 5a presents the speed of each vehicle during the simulation. We observe that vehicle 2 decelerates at $t = 13\,\mathrm{s}$ to avoid vehicle 1 when crossing a narrow corridor, showing the effectiveness of the proposed intra-formation collision avoidance strategy. Fig. 5b shows the formation errors of vehicle 1 and vehicle 2; we confirm that error converges quickly to 0 when the road is clear. The computation time profiles in Fig. 5c demonstrates the real-time ability of the proposed algorithm.

The second scenario considers isomorphic formation changes for four vehicles on a curvy road. The formation sequence is illustrated in Example 3. The time instants when we switch the formation are respectively $t = 15.4\,\mathrm{s}$, $t = 30.8\,\mathrm{s}$ and $t = 46.5\,\mathrm{s}$. Fig. 7 presents the trajectories of the four vehicles during the experiment. This experiment demonstrates that isomorphic formation changes can be performed smoothly while avoiding collisions. Moreover, the curvy nature of the road is handled nicely by our framework.

Videos of the two experiments are available on-line[1].

[1] https://youtu.be/QIGIgCmBr0A

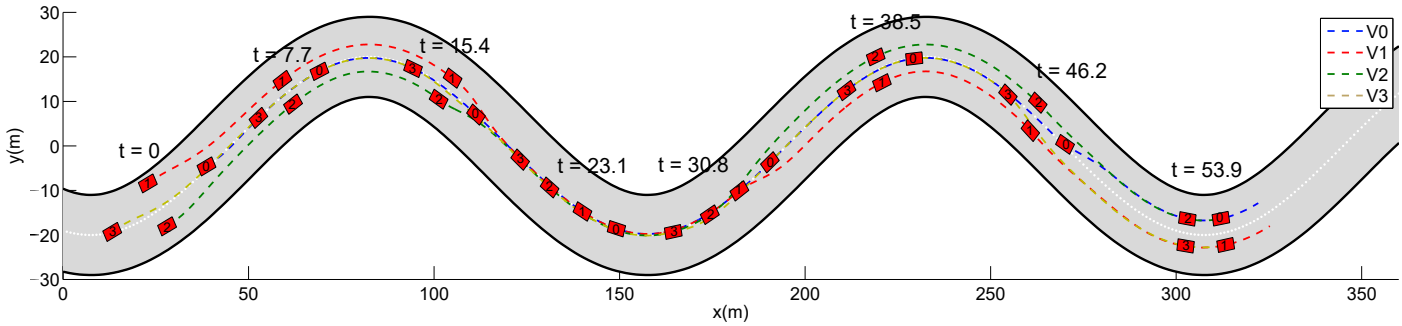


Fig. 7: Second scenario: trajectories of four vehicles.

VI. Conclusion

We have presented a distributed MPC-based formation control framework for car-like autonomous vehicles on the road. The proposed approach is distributed so that each vehicle locally computes a trajectory compatible with the formation control, using information exchanged through communication. We make use of logical rules to handle intra-formation collision avoidance. Moreover, we design a strategy for reconfiguration between isomorphic formations in a safe and timely manner. High-fidelity computer simulations have demonstrated the effectiveness of the approach in handling challenging on-road conditions with obstacles, road diets and curvy segments.

This preliminary work clearly opens several research perspectives. A first avenue is to study and enhance the set of collision avoidance rules; here the rules (9) have been chosen as they are symmetrical and intuitive. However, more complex rules can be considered. A second avenue of research is related to reconfiguration: the first simulation shows that we can avoid obstacles without needing an explicit reconfiguration. However, it seems rational to engage in a formal reconfiguration, which has been shown to be easily carried out in the second simulation. We will study the high-level decision-making process for reconfigurations. Finally, future work also includes the evaluation of this approach under realistic perception and communication conditions. This work is scheduled to be implemented in real vehicles under the framework of the European project AutoNET2030 [21].

References

[1] S. E. Shladover, C. A. Desoer, J. K. Hedrick, M. Tomizuka, J. Walrand, W.-B. Zhang, D. H. McMahon, H. Peng, S. Sheikholeslam, and N. McKeown, "Automated vehicle control developments in the PATH program," *Vehicular Technology, IEEE Transactions on*, vol. 40, pp. 114–130, Feb. 1991.

[2] B. J. Harker, "PROMOTE-CHAUFFEUR II & 5.8 GHz vehicle to vehicle communications system," in *Advanced Driver Assistance Systems, 2001. ADAS. International Conference on (IEE Conf. Publ. No. 483)*, pp. 81–85, 2001.

[3] E. Chan, P. Gilhead, P. Jelinek, P. Krejci, and T. Robinson, "Cooperative control of sartre automated platoon vehicles," in *19th ITS World Congress, Vienna, Austria*, pp. 22–26, 2012.

[4] M. Ono, G. Droge, H. Grip, O. Toupet, C. Scrapper, and A. Rahmani, "Road-Following Formation Control of Autonomous Ground Vehicles," no. Cdc, pp. 4714–4721, 2015.

[5] J. Shao, G. Xie, and L. Wang, "Leader-following formation control of multiple mobile vehicles," *Control Theory Applications, IET*, vol. 1, pp. 545–552, Mar. 2007.

[6] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "Leader–follower formation control of nonholonomic mobile robots with input constraints," *Automatica*, vol. 44, no. 5, pp. 1343–1349, 2008.

[7] W. Ren and R. W. Beard, "Virtual structure based spacecraft formation control with formation feedback," *AIAA Paper*, vol. 4963, 2002.

[8] W. Ren and R. Beard, "Decentralized scheme for spacecraft formation flying via the virtual structure approach," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 1, pp. 73–82, 2004.

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

[10] W. B. Dunbar, *Distributed receding horizon control of multiagent systems*. PhD thesis, California Institute of Technology, 2004.

[11] F. Borrelli, T. Keviczky, G. J. Balas, G. Stewart, K. Fregene, and D. Godbole, "Hybrid decentralized control of large scale systems," in *Hybrid systems: Computation and control*, pp. 168–183, Springer, 2005.

[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 *2016 IEEE Intelligent Vehicle Symposium*, 2016.

[13] I. Arad, S. Arogeti, and R. Ronen, "Coordinated path following control for a group of car-like vehicles with an application to intelligent transportation system," in *Control Automation Robotics Vision (ICARCV), 2014 13th International Conference on*, pp. 59–64, Dec 2014.

[14] A. Marjovi, M. Vasic, J. Lemaitre, and A. Martinoli, "Distributed graph-based convoy control for networked intelligent vehicles," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*, pp. 138–143, June 2015.

[15] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and Dynamic Vehicle Models for Autonomous Driving Control Design," pp. 2–7.

[16] T. Weiskircher and B. Ayalew, "Predictive ADAS : A Predictive Trajectory Guidance Scheme for Advanced Driver Assistance in Public Traffic *,"

[17] P. Bender, O. S. Tas, J. Ziegler, and C. Stiller, "The combinatorial aspect of motion planning: Maneuver variants in structured environments," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*, pp. 1386–1392, IEEE, 2015.

[18] T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G. J. Balas, "Decentralized receding horizon control and coordination of autonomous vehicle formations," *Control Systems Technology, IEEE Transactions on*, vol. 16, no. 1, pp. 19–33, 2008.

[19] O. Michel, "Webots: Professional Mobile Robot Simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. BIOROB-ARTICLE-2004-001, pp. 39–42, 2004.

[20] B. Houska, H. J. Ferreau, and M. Diehl, "ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.

[21] A. de La Fortelle, X. Qian, S. Diemer, F. Moutarde, and S. Bonnabel, "Network of Automated Vehicles: the AutoNet2030 vision," in *ITS World Congress*, 2014.