

Distributed Graph-Based Control of Convoys of Heterogeneous Vehicles using Curvilinear Road Coordinates

Iñaki Navarro

Florian Zimmermann

Milos Vasic

Alcherio Martinoli

Abstract—This paper investigates the problem of controlling a heterogeneous group of vehicles with the aim of forming multi-lane convoys. We use a distributed, graph-based control law, implemented in a longitudinal coordinate system parallel to the road. Each vehicle maintains a local graph with information from only nearby vehicles, in which the desired distances between vehicles are calculated dynamically. This allows for fast adaptation to the changes in the number of vehicles and their positions. We have also implemented a distributed mechanism that allows vehicles to change lane in a cooperative way within the convoy. Systematic experiments have been carried out in a high-fidelity simulator in order to show the performance of the proposed control law.

I. INTRODUCTION

Improving road traffic flow and safety, reducing fuel consumption, and enhancing driving comfort could be achieved by means of cooperative strategies for groups of autonomous vehicles. Different works have shown that when multiple vehicles drive within a spatially coordinated scheme such as convoys or platoons, safety can be increased [1], and fuel consumption reduced [2], [3].

Previous contributions in group control of vehicles have been mostly focused on platoon (i.e. single-lane convoys), using reactive spacing control methods for consecutive vehicles. Point-follower and vehicle-follower [4], [5], adaptive cruise control (ACC) [6], cooperative ACC (C-ACC) [7] and local controllers [8] are the main approaches for platoon control. In these strategies, the desired inter-vehicular spacing is kept by means of basic (usually first order) control rules such that every vehicle matches its distance and speed with the vehicle ahead. It is mathematically proven and experimentally shown that these reactive control methods can lead to string instability issues [9], [6], especially if the number of vehicles is large. Furthermore, it is very challenging to adapt these methods to multi-lane convoy control where there is no leading vehicle in front and the number of involved vehicles can change dynamically.

Kato et al. [10] introduced the concept of multi-lane convoy with the aim of increasing safety levels of autonomous vehicles in highways. Multi-lane convoys allow for cooperation among vehicles in different lanes, potentially increasing safety. In terms of control strategy they used a linear feedback (proportional) control system to adjust

the inter-vehicle distance between each vehicle and the leader of the convoy while using a very precise differential Global Navigation Satellite System (GNSS) in each car for localization. The approach proposed by this contribution showed two major limitations: first, the number of vehicles in the convoy was pre-established and static; and second, the convoy needed a leader.

In this article, we present a distributed controller for multi-lane convoys of heterogeneous vehicles. We consider an unknown number of vehicles on a road which are able to communicate locally with each other and can localize themselves. As the previous work by Goyal et al. [11] and Marjovi et al. [12] our approach is based on graph-based Laplacian control. The convoys implemented by Goyal et al. are static in terms of number of vehicles involved and the positions they occupy. The solution by Marjovi et al. goes one step further: the convoy adapts to the path given by the road, and vehicles can join and leave the convoy by means of a complex messaging system. The main difference of our approach is that the graph-based controller is applied in the longitudinal axis parallel to the road using curvilinear coordinates. This allows the convoy to adapt its shape to the road, and perform sharp turns. It also avoids the competition existing in Marjovi et al.'s solution between the Laplacian controller and the lane keeping controller.

In our work, the desired distance bias between vehicles is created and maintained calculating a reference offset from the frontmost vehicle by means of simple messages exchanged between neighbors. It allows the convoy to cope with vehicles of different lengths, and to implement a cooperative lane change mechanism. There are several previous works on cooperative lane change, e.g. [13], [14], but to our knowledge this is the first time it is addressed as an integrated operation within a multi-lane convoy.

The remainder of this article is organized as follows. In Section II, we describe the distributed graph-based controller, and the way its graphs are created and managed. The experimental setup, implementation details, and experimental results of the proposed controller are described and discussed in Section III. Finally, Section IV draws the conclusions of this work.

II. DISTRIBUTED GRAPH-BASED CONTROL OF CONVOYS

The convoy implementation presented in this paper grants a number of heterogeneous vehicles the capability to form and maintain a convoy on several lanes. It also allows vehicles to perform lane changes within the convoy. The

The authors are with the Distributed Intelligent Systems and Algorithms Laboratory, School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland. {inaki.navarro, florian.zimmermann, milos.vasic, alcherio.martinoli}@epfl.ch. Iñaki Navarro is supported by the European FP7 project AutoNet2030 (Grant Agreement NO. 610542). Milos Vasic is supported by PSA Groupe.

distributed controller which allows the convoy formation is based on Laplacian control.

A. Background

In graph-based theory, a directed simple graph with N elements is defined as a pair $\mathcal{G} = (V, E)$, where $V = \{v_i, i = 1 \dots N\}$ is the *vertex set*, and $E \subseteq V \times V$ is the *edge set*. The elements of E are ordered pairs of elements $e_k = (v_i, v_j)$ with $k = \{1 \dots |E|\}$. The i -th node *neighbors subset* is defined as $\mathcal{N}_i = \{\forall v_j \in V : (v_i, v_j) \in E\}$. Given such a graph, we can define the *incidence matrix* $\mathcal{I} \in \mathbb{R}^{N \times |E|}$ as:

$$\mathcal{I}_{i,k} = \begin{cases} -1 & \text{if } e_k = (v_i, v_j) \\ 1 & \text{if } e_k = (v_j, v_i) \\ 0 & \text{otherwise} \end{cases}$$

where e_k is the k -th edge of \mathcal{G} . The definition of the *incidence matrix* allows us to define the *Laplacian matrix* as

$$\mathcal{L} = \mathcal{I} \cdot \mathcal{W} \cdot \mathcal{I}^T \quad (1)$$

where the *weight matrix* $\mathcal{W} \in \mathbb{R}^{|E| \times |E|}$ is a diagonal matrix whose element $\mathcal{W}_{k,k}$ relates to the importance of each edge e_k .

In our problem, the vertexes (V) correspond to controlled agents or vehicles and edges (E) correspond to inter-vehicle communication links for position sharing. Using graph-based formation control, a stable solution to the formation problem in one dimension is given by [15]:

$$\dot{\vec{x}} = -\mathcal{L}(\vec{x} - \vec{b}) \quad (2)$$

where \vec{x} is the absolute position vector for all agents, and the desired offsets of agents to the formation are given by the bias vector \vec{b} . This basic control principle is applicable under the assumptions of holonomicity of vehicles, absolute positioning, and connectivity maintenance for ensuring the existence of edges in \mathcal{G} .

B. Controllers

In order to create the convoy two separate controllers are used. The lateral controller ensures that each vehicle is centered in the appropriate lane, while indirectly maintains a certain lateral distance between vehicles given by the lane width. The longitudinal controller implements the desired inter-vehicle distance by means of a distributed implementation of a graph-based control law.

The kinematics of our vehicles can be described by a bicycle model, which is realistic if we consider Ackermann-steering vehicles moving at low speeds and small steering angles. The inputs of the bicycle model are the speed v and the steering angle ϕ :

$$\begin{cases} \dot{x} &= \cos(\theta) \cdot v \\ \dot{y} &= \sin(\theta) \cdot v \\ \dot{\theta} &= \frac{\tan(\phi)}{L} \cdot v \end{cases} \quad (3)$$

where $[x, y]^T$ is the position of the middle point of the rear axle of the vehicle in Euclidean reference frame, θ represents the orientation of the car relative to the x -axis, and L is the wheelbase.

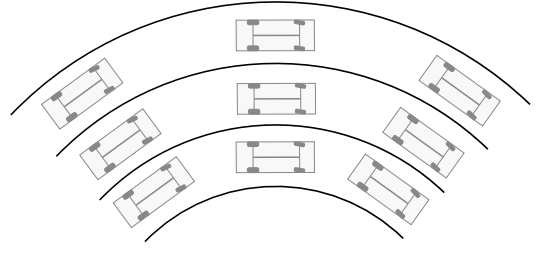


Fig. 1. Sketch of a convoy of nine vehicles in three lanes that adapts to the curvature of the road.

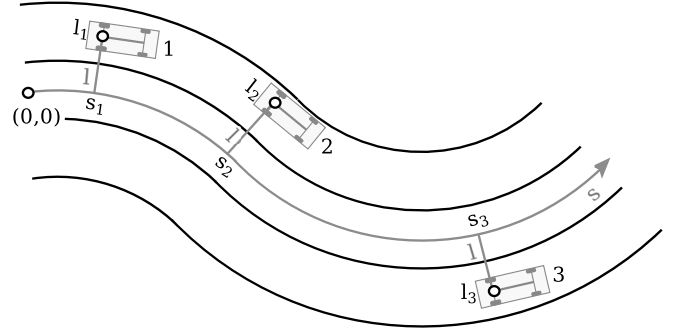


Fig. 2. Curvilinear coordinates (s, l) . The positions of three vehicles are presented, together with the reference s axis centered on the reference lane.

1) *Longitudinal Controller*: The longitudinal controller keeps vehicles at a desired inter-vehicle distance while moving them at a certain group speed. Figure 1 shows a sketch of a convoy which adapts to the curvature of the road. The vehicles in the top lane need to follow a longer path than those on the other lanes to avoid getting delayed given the larger radius curvature. For the same reason, and in order to keep vehicles on different lanes in parallel, the inter-vehicle distance in Cartesian coordinates differs between the lanes.

In order to make convoys adaptive to the road shape, we use curvilinear coordinates in two dimensions (s, l) . The longitudinal coordinate (s) of each vehicle is the position along the road. It corresponds to the length of the path followed on the road from an arbitrary origin $(0, 0)$ (see Fig. 2). Since there might be more than one lane in the road we choose the path followed by the center of a reference lane. The lateral coordinate (l) is the distance from the vehicle to the closest point on the reference lane (see Fig. 2).

The longitudinal controller is then implemented as a distributed Laplacian graph-based formation on the s coordinate. Each vehicle implements a modification of Eq. 2 which adds a desired forward group speed (V_{gdes}) to the convoy:

$$\dot{\vec{s}} = -\mathcal{L}(\vec{s} - \vec{b}) + V_{gdes} \quad (4)$$

where \vec{s} represents the positions of the vehicles in the s coordinate.

The graph used in Eq. 4 is calculated by each vehicle taking into account only the vehicles in its local neighborhood (see Section II-C). Although each individual vehicle performs formation control with its neighbors only, at the global level, the graph is collectively connected through

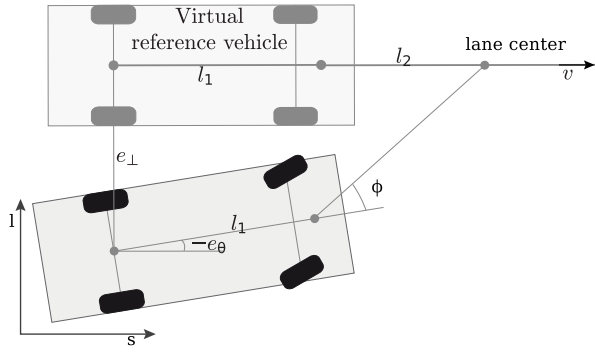


Fig. 3. Lateral and orientation error shown for a vehicle for a goal position centered and aligned with the lane. Picture is adapted from [11].

a sequence of smaller local graphs. Graph connectivity is a necessary condition for the consensus convergence of the formation [16]. Hence, vehicles can maintain convoy formation equally stable as in the case of a global graph.

This approach allows for a fully distributed implementation, i.e., each vehicle executes its own control loop locally. Furthermore, vehicles do not need to know the total number of vehicles nor their positions; only the positions of vehicles in their local neighborhood are necessary.

The desired speed of each vehicle in the s axis (\dot{s}_i) resulting from Eq. 4 needs to be translated into speed commands of each vehicle (v). We assume that all the vehicles follow the same direction of the road ($-\pi/2 < \theta < \pi/2$), and that most of the time they are parallel or almost parallel to the road ($\theta \simeq 0$), thanks of the action of the lateral controller (see Section II-B.2). Under this last condition, and assuming that the vehicle is in the reference lane of the coordinate system, we can express the first part of Eq. 3 in (s, l) coordinates as:

$$\dot{s}_i \simeq v \quad (5)$$

Since some vehicles might travel in other lanes than the reference lane of (s, l) , we need to cope with the different radii of each lane, so that the vehicles adapt to the curvature of the road. Our controller can be expressed as:

$$v = \dot{s}_i \cdot \frac{\delta s_{il}}{\delta s_{rl}} \quad (6)$$

where δs_{rl} is the arc-length infinitesimal of the reference lane at the closest point to the vehicle, and δs_{il} is the arc-length infinitesimal of the vehicle lane at the closest point to the vehicle.

2) *Lateral Controller*: The objective of the lateral controller is to reach the center of the lane with the correct orientation of the road. Without loss of generality, we can use the (s, l) coordinate system parallel to the road, with the s positive axis pointing in the direction of movement. Thus, the orientation θ of a vehicle is equal to the heading error $-e_\theta$ with respect to the desired final orientation (see Fig. 3), while the lateral error is $e^\perp = l_{lane} - l$, where l_{lane} is the l coordinate of the lane to which the vehicle belongs.

We use the lateral controller defined by Linderoth [17] which is able to bring e^\perp and e_θ to 0 under the assumption

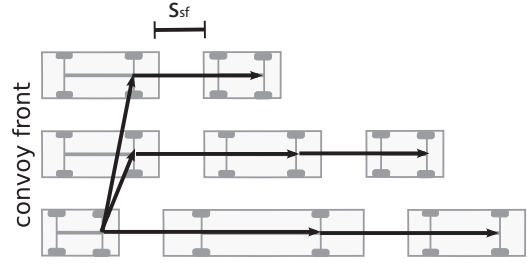


Fig. 4. A sketch of a convoy with relevant distances indicated. The solid lines with arrows show the tree graph formed to calculate the offset.

that the vehicle moves forward ($v > 0$):

$$\tan(\phi(t)) = \frac{-\cos(e_\theta(t))e^\perp(t) - (l_1 + l_2)\sin(e_\theta(t))}{l_1 - (l_1 + l_2)\cos(e_\theta(t)) + \sin(e_\theta(t))e^\perp(t)} \quad (7)$$

where l_1 and l_2 are two positive control constants.

C. Distributed Graph Construction

In this section we describe the way the local graphs are built by each vehicle, as well as the bias vectors (\vec{b}).

A vehicle is considered as a neighbor by any other vehicle if they are closer than a certain range (R). In our concrete implementation the positions of nearby vehicles are obtained via local communication, but to this end local perception would also work. At every execution cycle each vehicle sends its current position, and its ID , together with other variables that help to build the bias vector. When a vehicle receives a message from another vehicle it adds it to its local graph if it is under range R . Using the graph, it builds the Laplacian matrix (\mathcal{L}) by means of Eq. 1. Note that vehicles on different lanes are also considered neighbors if the distance to them is shorter than R .

In order for the convoy controller to work, the biases attributed by the vehicles to each other need to be consistent and symmetric (e.g., two vehicles need to independently attribute an equal but opposite bias to each other in their own coordinate frames). In addition, biases among all the vehicles need to be consistent when looking at the whole global graph.

By design, the vehicles in the front of the convoy align their front parts, and the vehicles following need to keep a safety distance s_{sf} to the precedent vehicle (see Fig. 4). This safety distance is measured in the s coordinate, so the Cartesian value will change depending on the curvature of the road (see Fig. 1).

For the bias calculation, each vehicle i computes an offset value (off_i) in the s coordinate referred to the offset of a common reference vehicle. There is no explicit declaration of the common reference vehicle, and vehicles continuously recalculate their off_i based on neighboring vehicles' information. Each vehicle i sends to its neighbors, together with its ID and position, the lane it is occupying ($lane_i$), its current offset (off_i), and its length ($length_i$).

Initially every vehicle sets its offset to an arbitrary common initial value, for instance $off_i = 0$. Once a vehicle i

receives a first message, it becomes part of a convoy and updates its graph and offset value. The offset is calculated every execution cycle in three possible ways depending on the positions its neighbors occupy. If there are neighboring vehicles on the same lane in front of the vehicle, the closest of them is chosen as local reference (vehicle r) and the offset is calculated as:

$$off_i = off_r + s_{sf} + length_i \quad (8)$$

If there are no neighboring vehicles on the same lane in front of it, but there are neighboring vehicles in front of it on other lanes, the farthest of them is chosen as local reference (r) and the offset is calculated as:

$$off_i = off_r - length_i + length_r \quad (9)$$

If there are no neighboring vehicles in front of it on any of the lanes then the offset is left unchanged and the vehicle itself is the common reference.

The bias to each neighbor n is calculated as the difference between the neighbor offset and the own offset:

$$b_n = off_n - off_i. \quad (10)$$

This algorithm makes the front of the convoy to align, and vehicles to move up to the front if there is no front neighbor on its own lane hindering it from doing so (Eq. 9). In addition each vehicle maintains a safety distance s_{sf} to his precedent vehicle (Eq. 8). The correct offset for each vehicle is automatically propagated from the head vehicle of the convoy to the tail, following a tree graph, from the roots to the leaves (see Fig. 4). Using only local communication and information, the graph resulting of the union of all the local graphs is connected, and the computed biases are consistent between them.

D. Lane Change

We have added a single-vehicle lane-change functionality within the convoy. It is implemented by temporal modifications of the biases of the vehicle changing lane and few of its neighbors. Lane changes are done from a lane to a contiguous one, but several concatenated lane changes can be performed.

For safety reasons, before a vehicle changes its lane, there needs to be space available on the target lane. Also, while the vehicle is moving from the initial lane to the target lane, the vehicles behind it on the initial lane must wait until the lane change is completed before moving into the available space in the original lane.

The lane change is initiated by the vehicle (C) performing it. It is implemented in four steps with the help of two other vehicles (H_1 and H_2) which act as helpers in the two central steps. The safety space is created by this three vehicles using the offset propagation mechanism and the whole maneuver is graphically represented in Fig. 5.

Step 0: The lane change is triggered by vehicle C and chooses a neighbor B on the target lane behind which it will insert itself. B is chosen as the frontmost neighbor in the target lane whose rear is behind vehicle C front. If there

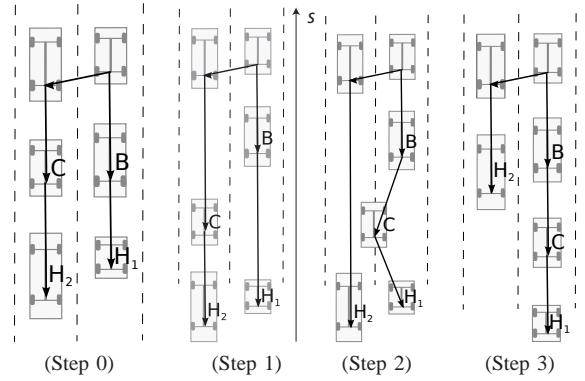


Fig. 5. Sketch of a lane change maneuver. The convoy direction coincides with the direction of the s coordinate. The solid lines with arrows show the graph formed at each step for the offset calculation. Step 0: vehicle C initiates a lane-change to the right lane, choosing as goal position a spot behind vehicle B . Step 1: vehicle C aligns itself with its new offset on the target lane, while vehicle H_1 creates space for C on the target lane. Step 2: vehicle C moves to the target lane. Vehicle H_2 waits until C has moved to its new lane. Step 3: vehicle H_2 fills the space left by vehicle C on its previous lane after C has finished its transition.

are no neighboring vehicles in the target lane or they are all in front of C then Step 1 is skipped.

Step 1: In this step, C stays on the original lane but positions longitudinally on the goal position it will occupy in the target lane. In addition, it chooses the vehicle behind vehicle B as its helper of type 1 (H_1), so that it adjusts its longitudinal positions to leave space for C . This two longitudinal adjustments are done by changing their offsets. Consequently, the offsets of all neighbors behind them on their lanes will be adjusted next time they rebuild their local graphs.

In order to set the new offset, C fakes its length using a virtual larger one, and sends it to its neighbors for their offset and bias calculations. It is calculated as:

$$length_C = 2 \cdot realLength_C - off_C + off_B + s_{sf} \quad (11)$$

where $length_C$ is the new fake length communicated to other vehicles and used to calculate its new offset and bias to neighbors, $realLength_C$ is the actual real length of the vehicle, off_C is the previously calculated offset of C , off_B is the offset of B , and s_{sf} is the safety distance.

By sending this fake length and calculating its new own offset with it, the appropriate biases are calculated and the right longitudinal adjustment takes place. The resulting longitudinal distance from B to C is $realLength_C + s_{sf}$.

If there is a neighbor H_1 behind B on the target lane, C designates it as a helper by setting a variable with the ID of H_1 in its messages. While H_1 receives these messages it needs to modify its position also by setting a fake length:

$$length_{H_1} = realLength_{H_1} + realLength_C + s_{sf} \quad (12)$$

This creates a distance of $2 \cdot s_{sf} + realLength_C + realLength_{H_1}$ between H_1 and B .

Step 2: Vehicle C periodically checks if there is enough free space on the target lane before proceeding to Step 2. In this step, C changes its lane number to the target

lane, sets back its $length_C = realLength_C$, and removes H_1 as a helper in the messages. When H_1 receives the first indicating its help is no longer needed, it sets back $length_{H_1} = realLength_{H_1}$. As a result, C calculates its offset directly from B , and H_1 from C (using Eq. 8). In addition C starts its movement towards the target lane.

Since C now belongs to the target lane, the vehicle behind it (H_2) will try to occupy the space C is going to leave. In order to avoid a collision between C and H_2 , C appoints H_2 as helper of type 2 in its local messages. Then H_2 sets a fake length as follows:

$$length_{H_2} = realLength_{H_2} + length_C + s_{sf} \quad (13)$$

This allows H_2 to remain at the same position relative to the vehicle that was previously in front of C . Note that the $length_C$ added here is the fake longer one computed and transmitted by C in Step 1.

Step 3: Once C is close enough to the center of the target lane, the lane change is over. C removes H_2 from the helper type 2 in its messages, allowing H_2 to close the gap with its front neighbor.

This lane change mechanism is implemented without the need for specific messages, using the graph-based formalism, IDs for each vehicle, and local communication. The actual lane changing maneuver is performed directly by means of the lateral controller (Eq. 7), and there is no smooth trajectory planning. Such trajectory planning is out of the scope of this paper.

III. EXPERIMENTS

In order to test our distributed algorithm, we have performed a set of systematic experiments using Webots [18], a high-fidelity submicroscopic simulator originally developed for mobile robotics, which has recently been upgraded to support automotive platforms [19]. In this context, *submicroscopic* means that it provides a higher level of detail than usual microscopic models, faithfully reproducing intra-vehicle modules (e.g., individual sensors and actuators).

Our simulated vehicle models incorporate basic rigid dynamics and properties of real cars such as typical steering dynamics response and friction of the tires.

We show results for eight sets of experiments involving different vehicles and configurations of roads in order to test the functionalities of our proposed algorithm.

A. Implementation Details

The graph-based convoy is implemented in a distributed way, each vehicle running separately a controller with a frequency of 15.625 Hz (64 ms period). There is no shared memory between the controllers, and so vehicles interact with each other only through message passing.

The different controller parameters used in the experiments are summarized in Table I. The values are the same for every type of vehicle used, so they are not tuned to cope with individual differences.

The speed output of each vehicle controller is translated by the *Car* library integrated in Webots into motor acceleration

TABLE I
CONTROLLER PARAMETERS USED IN THE IMPLEMENTATION OF THE
EXPERIMENTS.

Parameter	Value
Matrix \mathcal{W} weights $\mathcal{W}_{k,k}$	0.08
Desired group speed V_{gdes}	11.11 m/s (40 km/h)
Lateral controller parameter l_1	3 m
Lateral controller parameter l_2	6 m
Safety distance s_{sf}	15 m
Local graph range R	50 m

commands by means of a PID controller. The steering angle output is transferred to the wheels using also a built-in PID controller of the same library.

Each vehicle is able to measure its position by means of a realistic GNSS module, its heading using a simulated compass, and can share information with nearby vehicles using radio communication. The GNSS is simulated with and without noise. The added noise is Gaussian with zero mean and standard deviation of 0.25 m in order to obtain realistic measurements of the self-localization of the vehicles. In the real-world, accuracy of 0.25 m is achievable using GNSS receivers together with real-time ground based corrections [20]. The compass measurements can be also perfect or have added Gaussian noise with zero mean and standard deviation of 0.02 rad .

We have added an arbitrary package loss probability to make the communication more realistic. In this communication mode every package can be discarded with a probability of 0.3.

To better deal with package loss, a simple filter mechanism was implemented. Every message sent also includes a time stamp and the current velocity of the vehicle. In reception, if no update has been received at a certain execution cycle from a particular vehicle already inside the neighborhood, its position is linearly estimated from its last transmitted position and velocity. Additionally, if no new message is received from a neighbor for a certain duration, the neighbor is removed from the neighborhood. This timeout was set to 1 s .

In order to know positions within the road and to be able to compute distances in the (s, l) coordinate system, each vehicle leverages a common map of the road. This map consists of a sorted list per lane containing spaced points of the center of the lanes.

B. Experimental Setup

We have tested eight different experimental setups, by combining two roads, two types of convoys and different combinations of localization noise and package loss.

The first road ($R0$) has an oval (or zero) shape (see Fig. 6a). The second road ($R8$) has an almost eight shape (see Fig. 6b). Both roads have four lanes. The second most inner lane is used as reference lane for the s coordinate calculation. The length of one lap of each road, and the radius of curvature measured in the middle of the road (between lanes 2 and 3) are presented in Table II.

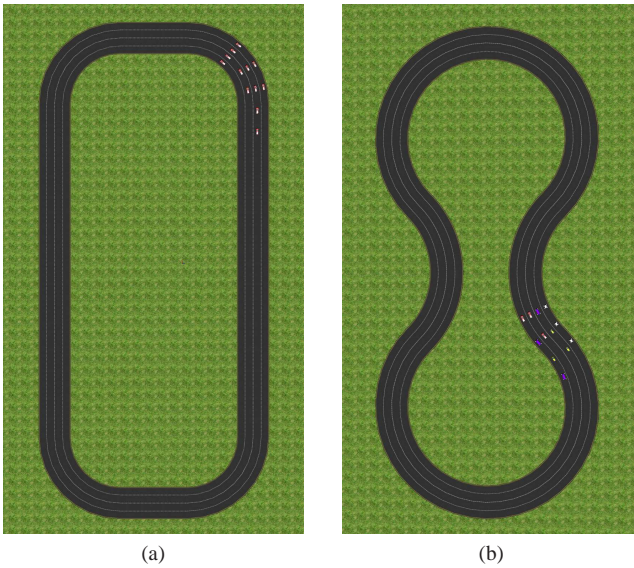


Fig. 6. (a) Road *R0*. (b) Road *R8*.

TABLE II
ROAD CHARACTERISTICS.

Road	Lap length (m)	Radius (m)
<i>R0</i>	979	53
<i>R8</i>	917	73

TABLE III
VEHICLE CHARACTERISTICS.

Vehicle	Wheelbase (m)	Weight (kg)
<i>BMW X5</i>	2.995	2000
<i>Citroën C-Zero</i>	2.55	1200
Custom 1	4	1000
Custom 2	2	1000

Four different type of vehicles are present in the experiments. We use realistic and calibrated models of a *Citroën C-ZERO* and a *BMW X5*. The other two vehicles are custom models of non-existent real cars. Their main characteristics are described in Table III.

Two types of convoys are used in the experiments. The first (*Hom*) is homogeneous, consisting of 12 *BMW X5*. The second (*Het*), heterogeneous, is formed of three vehicles of each type (12 in total).

We explore four combinations of localization noise and package loss: perfect (*Perf*), with no package loss or localization noise; just package loss (*PackL*); only adding localization noise (*PoseN*); and realistic (*Real*), which includes package loss and localization noise.

Lane change is only tested in the fourth experimental set. Automatic lane changes are triggered by a concrete vehicle at times 60 s and 120 s. If lanes exist both at the left and right of the vehicle, the target lane is chosen randomly with equal probabilities.

Every set consists of 20 experiments of 150 s, each with different random initial poses. The initial random orientation is given by a uniform distribution centered in the direction of the road and with a range of $\pm 0.5 \text{ rad}$. The initial positions

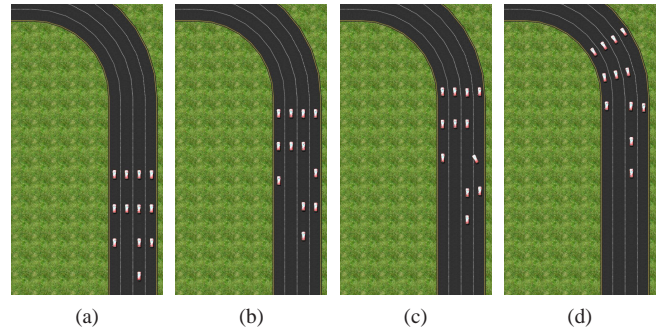


Fig. 7. Lane change in experimental set *ROHomRealLaneCh*. The vehicle on the rightmost lane, second row, intends to change lane to the left. (a) Step 0. (b) Step 1 has almost been completed and there is enough space to proceed with the lane change. (c) Step 2, the vehicle performs the lateral movement to reach the target lane. (d) Step 3, the vehicles in the rightmost lane fill the gap left by the vehicle that just changed lane.

are also given by a uniform distribution, covering the width of the road and a road length of 60 m. Potential overlapping poses are discarded.

Each experimental set is named given a concatenation of the road type, convoy type, and position and communication combination. For instance, *ROHomPerf* refers to the experimental set in road *R0*, with 12 *BMW X5* vehicles and perfect communication and self-localization. If the experiments include lane change the suffix *LaneCh* is added.

C. Results

We look at four metrics to measure the performance of our controller. The first two are the angular error (e_θ) and lateral error (e^\perp), which show the individual error of each vehicle within the lane (see Fig. 3).

The third is the longitudinal error measured in the s coordinate (e^\parallel). It is computed as the average of the longitudinal distance errors to everyone of the neighbors in its local graph. The error to each neighbor is the difference in absolute value between the desired bias and the actual distance to that neighbor. It is measured individually by each vehicle, but reflects the performance of the convoy, particularly of the vehicle and its neighbors. Note that if there is a short temporal miscalculation in the local graph and biases, then the computed longitudinal error will be high even if the vehicles are properly positioned.

The last metric, group speed or convoy speed (V_g) is the speed in the s coordinate averaged for every vehicle.

Four captures of the lane change mechanism of a experiment of set *ROHomRealLaneCh* are shown in Fig. 7.

Figure 8 shows the evolution over time of the four metrics also for set *ROHomRealLaneCh*. Initially the three errors (e_θ , e^\perp , and e^\parallel) are large and the convoy speed is low, but after $t = 15 \text{ s}$ the errors decrease and the group speed reaches values close to the desired group speed (V_{gdes}). When the convoy drives in zones showing curvature, the angular and lateral error increase as it can be seen on the corresponding plots. The lane changes triggered at times 60 s and 120 s can be observed when looking at the longitudinal error and group speed plots. It can be seen how the longitudinal error

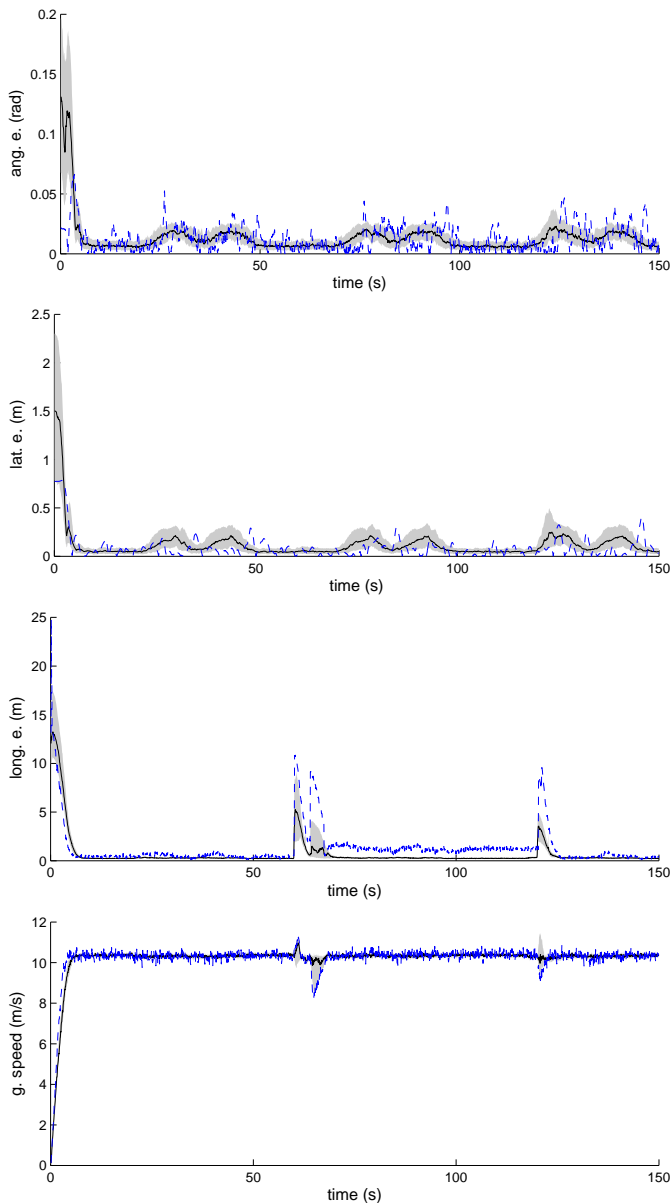


Fig. 8. Plots over time of metrics e_θ , e^\perp , e^\parallel , and V_g for set *ROHomRealLaneCh*. They aggregate the values for each of the 20 experiments and 12 vehicles in the case of e_θ , e^\perp and e^\parallel , and for the 20 experiments for V_g since it is a measure of the whole convoy. The black solid lines represent the median, while the gray bands above and below show the first and third quartile. The blue dashed line shows the value of a single vehicle during a concrete experiment in the case of e_θ , e^\perp and e^\parallel , and of a single experiment for the whole convoy for V_g .

increases during the lane change process due to the changes in the graph and continuous adaptation to it. Note that for the periods of time when the lane change is not taking place these plots are also representative of *ROHomReal*.

The boxplots in Fig. 9 allow us to understand how the localization error, package loss, convoy type and road affect the performance of the convoy. The angular error (Fig. 9a) and lateral error (Fig. 9b) follow a similar pattern when comparing them for the different sets of experiments. There is no significant difference between perfect communica-

tion and losing packages when we look at *ROHomPerf* and *ROHomPackL*, and these two sets have the best performance. Both angular and lateral error slightly increase when self-localization noise is added (see *ROHomPoseN* and *ROHomReal*), as it can be expected. If we compare the heterogeneous convoy with the homogeneous, the error in the heterogeneous case is slightly larger. This may be due to not completely tuned parameters. Finally, we can see that the errors in road *R8* are the largest, since all the segments in the road have curvature.

The longitudinal error (Fig. 9c) is considerably low except for the outliers. In the absence of positioning noise (*ROHomPerf* and *ROHomPackL*) the longitudinal error is lower than for the rest of the cases. There seem not to be effect on longitudinal error when comparing the convoy type or the road type. We suspect that outliers larger than 2.5 m might be due to temporal glitches in the graph that lead to computed large errors, and do not imply large errors in the actual distances between vehicles.

When looking at the group speed (Fig. 9d), it can be seen that speeds are slightly lower than the desired one (11.11 m/s). This lower speed can be expected since the group speed is controlled in open loop. The largest speeds are found in sets *ROHomPerf* and *ROHomPoseN*, those in which there is no package loss. When package loss is introduced the group speed is reduced. There is no effect of the localization noise, convoy type or road on the speed performance.

IV. CONCLUSIONS

We have designed and successfully tested a distributed algorithm for multi-lane convoys of heterogeneous vehicles. We base our solution on graph-based control and use curvilinear coordinates, which allows the convoy to adapt to the shape of the road.

One of the novelties of the proposed solution is the distributed mechanism which creates and maintains the neighboring graphs and biases. It allows for coping with vehicles of different lengths and adapting to changes in the number of vehicles and their positions. This mechanism potentially facilitates changing the safety distance depending on the vehicles involved. Although not implemented here, heavy vehicles such as trucks, might need to keep a larger safety distance.

The convoy speed is currently fixed to a constant value. A future improvement could be to modify the group speed according to the local properties of the road. It will require a consensus mechanism on the group speed between the vehicles.

The reference trajectories on each lane are currently fixed, causing the lane changes to be not completely smooth. An on-line path planner could improve the performance.

Experimental validation was carried out in a high fidelity simulator. The next step will be to test the convoy controllers using real vehicles in the framework of the FP7 European project *AutoNet2030* [21].

REFERENCES

- [1] L. Xu, L. Y. Wang, G. Yin, and H. Zhang, "Communication information structures and contents for enhanced safety of highway vehicle platoons," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4206–4220, 2014.
- [2] F. Farokhi and K. H. Johansson, "A game-theoretic framework for studying truck platooning incentives," in *IEEE 16th International Conference on Intelligent Transportation Systems*, 2013, pp. 1253–1260.
- [3] K.-Y. Liang, J. Mårtensson, and K. H. Johansson, "When is it fuel efficient for a heavy duty vehicle to catch up with a platoon?" in *7th IFAC Symposium on Advances in Automotive Control*, 2013, pp. 738–743.
- [4] 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," *IEEE Trans. on Vehicular Technology*, vol. 40, no. 1, pp. 114–130, 1991.
- [5] Y.-J. Pan, "Decentralized control of vehicles in platoons with robust nonlinear state estimation," in *Proc. IEEE Int. Conf. on Automation Science and Engineering*, 2008, pp. 145–150.
- [6] L. Xiao and F. Gao, "Practical string stability of platoon of adaptive cruise control vehicles," *IEEE Trans. on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1184–1194, 2011.
- [7] C. Wang and H. Nijmeijer, "String stable heterogeneous vehicle platoon using cooperative adaptive cruise control," in *IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 1977–1982.
- [8] K.-Y. Liang, "Coordination and routing for fuel-efficient heavy-duty vehicle platoon formation," Licentiate Thesis, KTH, Royal Institute of Technology, 2014.
- [9] P. Seiler, A. Pant, and K. Hedrick, "Disturbance propagation in vehicle strings," *IEEE Trans. on Automatic Control*, vol. 49, no. 10, pp. 1835–1842, 2004.
- [10] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, and H. Fujii, "Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications," *IEEE Trans. on Intelligent Transportation Systems*, vol. 3, no. 3, pp. 155–161, 2002.
- [11] S. Goyal, R. Falconi, and A. Martinoli, "Local graph-based distributed control for safe highway platooning," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010, pp. 6070–6076.
- [12] A. Marjovi, M. Vasic, J. C. B. Lemaître, and A. Martinoli, "Distributed Graph-based Convoy Control for Networked Intelligent Vehicles," in *Proc. IEEE Intelligent Vehicles Symp.*, 2015, pp. 138–143.
- [13] D. Wang, M. Hu, Y. Wang, J. Wang, H. Qin, and Y. Bian, "Model predictive control based cooperative lane change strategy for improving traffic flow," *Advances in Mechanical Engineering*, vol. 8, no. 2, pp. 1–17, 2016.
- [14] U. Khan, P. Basaras, L. Schmidt-Thieme, A. Nanopoulos, and D. Katsaros, "Analyzing cooperative lane change models for connected vehicles," in *2014 International Conference on Connected Vehicles and Expo*, 2014, pp. 565–570.
- [15] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.
- [16] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Trans. on Robotics*, vol. 23, no. 4, pp. 693–703, 2007.
- [17] M. Linderoth, K. Soltesz, and R. Murray, "Nonlinear lateral control strategy for nonholonomic vehicles," in *American Control Conference*, 2008, pp. 3219–3224.
- [18] O. Michel, "Webots: Professional mobile robot simulation," *Int. Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004.
- [19] S. Goyal, Y. Zhang, and A. Martinoli, "A realistic simulator for the design and evaluation of intelligent vehicles," in *13th Int. IEEE Conf. on Intelligent Transportation Systems*, 2010, pp. 1039–1044.
- [20] W. Hedgecock, M. Maroti, J. Sallai, P. Volgyesi, and A. Ledeczi, "High-accuracy differential tracking of low-cost gps receivers," in *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, 2013, pp. 221–234.
- [21] A. de La Fortelle, X. Qian, S. Diemer, J. Grégoire, F. Moutarde, S. Bonnabel, A. Marjovi, A. Martinoli, I. Llatser, A. Festag, and K. Sjöberg, "Network of automated vehicles: The AutoNet2030 vision," in *21st World Congress on Intelligent Transport Systems*, 2014. doi: 10.13140/2.1.2390.4649

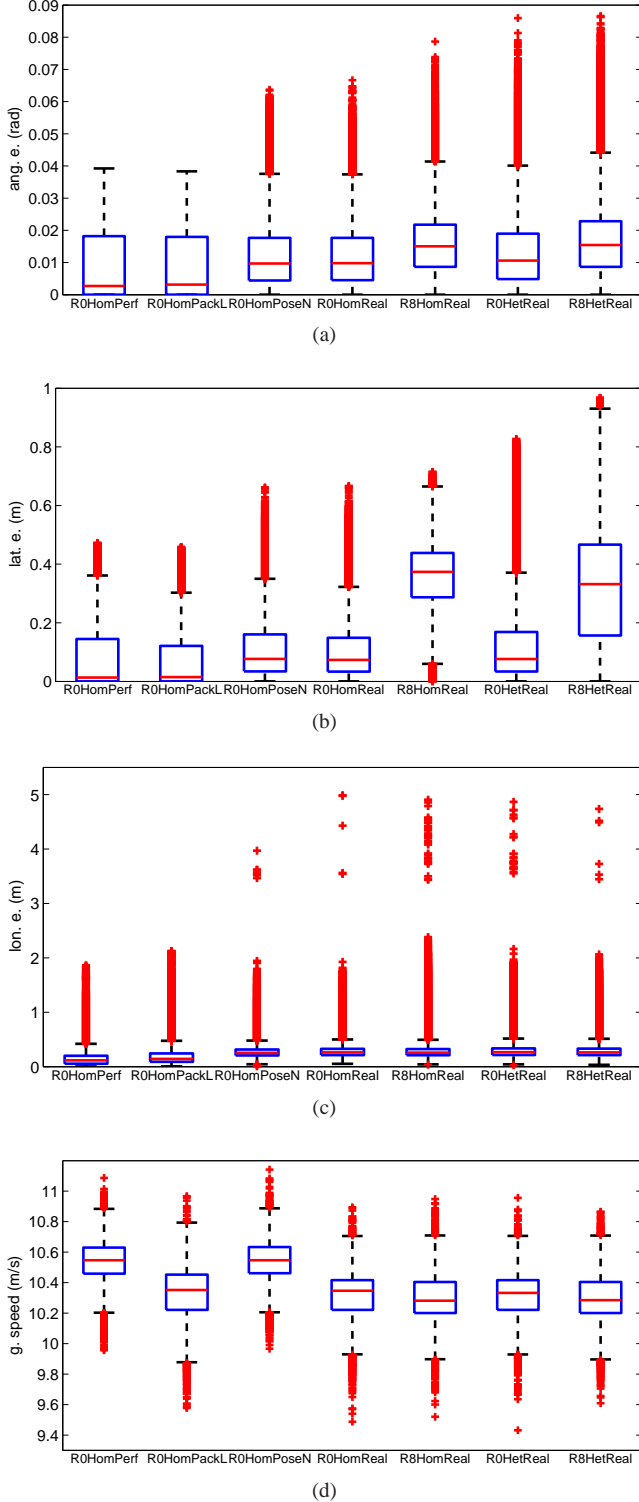


Fig. 9. Boxplots for each of the metrics (e_θ , e^\perp , e^\parallel , and V_g) for seven sets of experiments. Each box aggregates all the measures from $t = 45$ s to $t = 150$ s (the end of the experiment) for the 20 experiments of each set. The box represents the upper and lower quartiles, the line across the middle marks the median, the bars extend to the most extreme data points not considered outliers, and the red crosses show outliers. (a) Angular error (e_θ). (b) Lateral error (e^\perp). (c) Longitudinal error (e^\parallel). (d) Group speed (V_g).