Distributed Graph-based Convoy Control for Networked Intelligent Vehicles

Ali Marjovi, Milos Vasic, Joseph Lemaitre and Alcherio Martinoli

Abstract— This paper presents an approach for formation control of multi-lane vehicular convoys in highways. We extend a Laplacian graph-based, distributed control law such that networked intelligent vehicles can join or leave the formation dynamically without jeopardizing the ensemble's stability. Additionally, we integrate two essential control behaviors for lane-keeping and obstacle avoidance into the controller. To increase the performance of the convoy controller in terms of formation maintenance and fuel economy, the parameters of the controller are optimized in realistic scenarios using Particle Swarm Optimization (PSO), a powerful metaheuristic optimization method well-suited for large parameter spaces. The performances of the optimized controllers are evaluated in high-fidelity multi-vehicle simulations outlining the efficiency and robustness of the proposed strategy.

I. INTRODUCTION

Developing cooperative strategies for groups of future intelligent vehicles is one of the main objectives in the automotive industry aiming at improving road traffic flow and safety, reducing CO_2 emissions and enhancing driving comfort. Grouping neighboring vehicles (in form of platoons or convoys) is one possible way to achieve these objectives. Many previous research projects in this field (e.g., PATH [1], CHAUFFEUR I and II [2], and SARTRE [3]) have proven that fuel consumption could be decreased by 15% to 30% while the road throughput on motorways can potentially be multiplied by a factor of 3 to 5 through group control of vehicles. This current study is integrated in an ongoing European project named AutoNet2030[1] and proposes a methodology for dynamic convoying of intelligent vehicles in highway scenarios.

Differently from other contributions, in our work we consider not only single-lane pre-established fixed formations, but also multi-lane loose formations able to dynamically accommodate joining and leaving vehicles. In this context, *a group of three or more cooperative vehicles maintaining a loose formation (typically their distance and speed)* is defined as *convoy* [4]. In contrast to *platoons*, in convoys neither a leading vehicle nor a centralized controller/supervisor exists, the formation is not constrained to a single lane, and its structure is loose and dynamic.

Obtaining such features pose difficult challenges to design control strategies for convoys.

The cooperation, control, and management of the vehicles in a cooperative area can be done either in a centralized or decentralized fashion. Although centralized techniques can usually result in optimal solutions, they impose the requirement of high level of connectivity in the system and generate a substantial computational burden in the central unit. In systems with a large number of inputs and outputs, distributed architectures have several inherent advantages over their centralized counterparts, including fault tolerance, reliability, and scalability [5]. This study tackles the convoy problem without using any point of centralization, leveraging a fully distributed control law.

To state the problem, consider an unknown number of intelligent vehicles in a road which are able to communicate with each other locally and can localize themselves. The problem is how these vehicles can establish a dynamic multi-lane convoy which remains stable while allowing new cars to join, as well as the current cars to leave. The convoy control strategy should integrate essential driving behaviors, namely lane-keeping and obstacle avoidance, and should maximize the overall fuel economy of the vehicles and minimize formation errors.

Most of the works in group control of vehicles have been focused on single-lane platooning problems using reactive spacing control methods for consecutive vehicles. Point-follower and vehicle-follower [1], [6], adaptive cruise control (ACC) [7], [8], cooperative ACC (C-ACC) [9] and local controllers [10] are the main approaches for single lane platooning control. In these strategies, the desired inter-vehicular spacing is maintained through basic (usually first or second order) control laws such that every controlled vehicle matches its distance and speed with the vehicle ahead. It is mathematically proven and experimentally shown that these reactive control methods lead to string stability issues [7], especially if the group size is large. Moreover, 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 is dynamically changing.

Kato et al. [11] introduced the concept of multi-lane platooning for increasing the safety levels of cooperative intelligent vehicles in highways. They showed that in a multi-lane group control, the safety level can be increased since vehicles cooperate and coordinate not only within the same lane but also with the vehicles in the neighboring lanes. The downside of this work was that the number of vehicles in this study was fixed (to 5) and the first vehicle

acted as the leader. A notable exception is represented by the work of Gowal et al. [12], which proposed leaderless graph-based control [13] for multi-lane platooning. Their proposed graph-based controller is capable of driving the cars to any arbitrary formation. Using distributed graph-based control, each vehicle can independently compute its next goal point. The result is that the whole group will eventually converge to the desired formation. One drawback of this work is that since the formation graph is apriori defined, the topological location of each vehicle in the formation structure as well as the number of vehicles are pre-established, which implies that no vehicle can leave or join the group. Another downside of the three contributions above is that their control algorithms do not take into account neither road and lane geometries nor dynamic obstacles, meaning that the platoon may get established in any random position which can be outside of the roads and will not have the ability of avoid possible obstacles.

In this paper, we improve the scalability of the graph-based formation control by defining two new concepts, *local identifiers and neighborhood*, such that vehicles can generate dynamic graphs with their neighbors. This way, vehicles can join or leave the group dynamically without jeopardizing the group's stability. Moreover, we integrate lane-keeping and obstacle avoidance behaviors for the intelligent vehicles into the proposed convoy control strategy. Finally, we also optimize the parameters of the formation topology using Particle Swarm Optimization (PSO), a powerful metaheuristic method well adapted for optimization in spaces characterized by large dimensions and real numbers. To the best of our knowledge, no previous work has proposed a graph-based control strategy which leads to dynamic multi-lane convoy formations.

II. DYNAMIC GRAPH-BASED CONVOY CONTROL

In this section we briefly describe the static graph-based formation and then upgrade it to dynamic multi-lane convoy control.

A. Static graph-based formation control

Considering a fixed number of vehicles initially placed in random locations in the space, the question is how to design a distributed control algorithm which guides the individual vehicles to converge into a desired spatial formation (e.g., the ones shown in Fig. 1 and Fig. 2). As already mentioned, graph-based formation control [12], [13] is a solution for solving this problem. In graph-based formation, an undirected graph $\mathcal{G} = (V, E)$ is defined in which vertexes V correspond to controlled agents (vehicles in this case) and edges E correspond to inter-vehicle communication and relative positioning links. Built upon basic linear algebra, a stable solution to the formation control problem in two dimensions [14] is given by:

$$\dot{\mathbf{x}} = -\left(\mathcal{L} \otimes I_2\right)\left(\mathbf{x} - \mathbf{b}\right) + v_G \qquad (1)$$

with $\mathcal{L} = \mathcal{I} \cdot \mathcal{W} \cdot \mathcal{I}^T$, where $\mathcal{L}$ (called Laplacian matrix) is obtained from the incidence matrix $\mathcal{I}$ that defines the
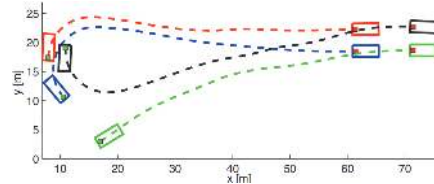

Fig. 1. Trajectories of four cars converging to a rectangular formation using static graph-based control. The picture is adapted from [12].

edges of $\mathcal{G}$ and the *weight matrix* $\mathcal{W}$ which is a diagonal matrix used to tune the weights assigned to the edges. I_2 is simply a 2×2 identity matrix (since we work only in a two-dimensional plane). The (x, y) absolute position vector for all vehicles is given by $\mathbf{x}$, and the desired offsets of the vehicles to the formation centroid are given by the bias matrix $\mathbf{b}$. The parameter v_G represents the desired (goal) velocity of the vehicles.

Since usual vehicles are non-holonomic systems, an appropriate translation between the holonomic state variables $\dot{x}(t)$, $\dot{y}(t)$ and the non-holonomic control variables of a vehicle (longitudinal speed and steering) is needed. Similar to [12], we model the non-holonomic vehicles using a kinematic bicycle model which is readily studied in the literature [15]. As Fig. 1 demonstrates, Eq. (1) guides the vehicles on trajectories which will lead to the desired formation defined by the biases. Additional details about non-holonomic graph-based control are provided in [12], [13].

B. Dynamic graph-based convoy control

In the real world, vehicles are obliged to respect lanes; we therefore take this into account in the definition of our formation topologies (i.e., single-lane convoy, two-lanes convoy, three-lanes convoy, etc.). On the other hand, the vehicles occasionally need to avoid obstacles on their lane due to any possible reason. We propose an *interleaved multi-lane topology* shown in Fig. 2 and Fig. 4 to allow vehicles to merge into their side lanes in case of avoiding obstacles without colliding with the other vehicles. This topology provides enough space on the left and right of the vehicles for temporary usage and thus increases their safety. The illustration in Fig. 2 shows the most general case, when there are many vehicles in the convoy (both in terms of length and number of lanes); however, in practice the same principles can be followed to establish convoys with the same topology in fewer lanes.

To make the static graph-based formation control scalable and dynamic (i.e., to enable vehicles to join or leave the convoy or to change lanes within the convoy), we propose an approach in which graphs containing connections between vehicles are dynamically created, locally maintained, and automatically modified on the fly. We first explain the main ingredients of the dynamic graphs in our system, namely *local neighborhood* and *local identifiers*.

Local neighborhood: We define a neighborhood $\mathcal{N}_D$ of a vehicle using topological distance, that is, distance measured in number of vehicles (as opposed to a metric distance measured in meters). See Fig. 2 for an illustration of
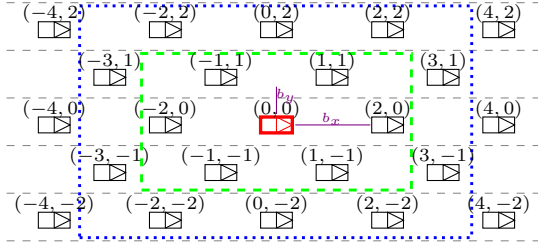
Fig. 2. Topological neighborhood $\mathcal{N}_D$ of the vehicle shown in red. The green dashed line depicts $\mathcal{N}_D = 1$, while the blue dotted line depicts $\mathcal{N}_D = 2$. Vehicles are enumerated in the local coordinate system of the vehicle shown in red. b_x and b_y represent the longitudinal and lateral bias offsets between the immediate neighbors.
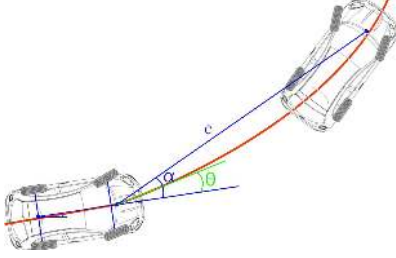

Fig. 3. The distance and azimuth of one car relative to another. The red line shows the road path.

neighborhoods for $\mathcal{N}_D = 1$ and $\mathcal{N}_D = 2$. The same principle can be applied in the cases where the number of lanes is limited or when there are no vehicles in front or behind the vehicle for which the neighborhood is determined.

Local identifiers: In this method, each vehicle enumerates the other vehicles in its vicinity using its local coordinate frame, by assigning them local identifiers which are ordered pairs (2-tuples) containing topological distances in x- and y-axes (in its local right-handed coordinate system). For instance, a vehicle driving directly in front and in the immediate left lane will be assigned an identifier $(1, 1)$, while a vehicle driving behind in one lane to the right will be considered to have an identifier $(-1, -1)$. This concept is depicted in Fig. 2. Finally, each vehicle will contain a list of local identifiers for vehicles in its neighborhood. Each local identifier is paired with a physical (global) ID of the corresponding vehicle, which is unique.

Graph-based formation control for multi-lane interleaved convoys: The Laplacian control (Eq. 1) can be upgraded to a decentralized approach, assuming a connected (but not necessarily complete) graph, using only relative positioning information [12], [13]. Making the assumption that any vehicle $\mathcal{V}^m$ is able to measure its distance $e_{m,n}$ and azimuth $\alpha_{m,n}$ to other neighboring vehicles $\mathcal{V}^n$, the controller running in each vehicle (relative to its own coordinate frame) for the desired multi-lane interleaved formation is modified to:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \sum_{(m,n) \in \mathcal{N}_D} w_{m,n} \cdot \left(e_{m,n} \cos\left(\alpha_{m,n} - \theta \right) - m \cdot \left(\frac{b_x}{2} \right) \right) \\ \sum_{(m,n) \in \mathcal{N}_D} w_{m,n} \cdot \left(e_{m,n} \sin\left(\alpha_{m,n} - \theta \right) - n \cdot b_y \right) \end{bmatrix} + \begin{bmatrix} v_{Gx} \\ v_{Gy} \end{bmatrix}$$

(2)

where $m, n \leq \mathcal{N}_D$ represent the topological distance in x-

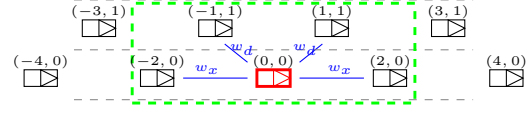and y-axis (local identifier) of the car to other vehicles in its local neighborhood and θ is the angle of the road at the current position of the car (see Fig. 2 and Fig. 3). It should be noted that in Eq. 2, the bias distance to the neighboring vehicles with $id = (m, n)$ is carefully designed for the interleaved formations presented in this paper.

For example, consider the two-lanes convoy depicted in Fig. 4 for $\mathcal{N}_D = 1$. Vehicle $(0, 0)$ is aware of its four closest neighbors, namely vehicles denoted with identifiers $(-2, 0)$, $(-1, 1)$, $(1, 1)$, and $(2, 0)$. Based on Eq. (2) the vehicle $(0, 0)$ performs graph-based control using its bias offsets and distances relative to the only four direct neighbors in its graph. In this formula, the x-bias considered for vehicle $(2, 0)$ will be b_x where as for vehicle $(1, 1)$ it will be $\left(\frac{b_x}{2} \right)$. While the y-bias for vehicle $(2, 0)$ will be zero where as for the vehicle $(1, 1)$ it will be b_y.

In order to have a homogeneous formation, we categorize the edges' weights into two groups; the longitudinal edge weights (w_x), and the diagonal edge weights (w_d), so that all the longitudinal and diagonal edges' weights are identical. Fig. 4 depicts this concept. So $\forall (m, n) \in \mathcal{N}_D$ we have:

$$w_{m,n} = \begin{cases} w_x & \text{if } n=0 \\ w_d & \text{otherwise} \end{cases}$$

(3)

In the proposed method, although an individual vehicle is performing formation control with its neighbors only, at the global level, the graph is collectively connected through the sequence of smaller local graphs. Hence, vehicles can maintain convoy formation equally stable as in the case of the static graph-based formation control. Furthermore, this approach is fully dicentralized, that is, each vehicle executes its control loop locally. Vehicles do not need to know the total number of vehicles nor their absolute position in the formation; only the relative poses within the local neighborhood are necessary.

C. Convoy functions

Once vehicles know how to build and maintain their dynamic graphs, various simple message passing functions allow vehicles to change lanes, leave the convoy, or new vehicles to join. For example, in case of joining a convoy, the new vehicle broadcasts a join request, which is received by all the vehicles within its communication range. Upon receiving a join request, each vehicle verifies if it is the closest to the new vehicle by checking its local graph and the position of the joining vehicle. The closest vehicle determines the position at which the new vehicle should join (e.g., in front or behind). Then, it communicates the relevant convoy parameters to the new vehicle (such as the local graph containing ID of neighboring vehicles, bias parameters, weights of the graph-based control, convoy speed, etc.) and

Fig. 4. Example of a two-lanes convoy for $\mathcal{N}_D = 1$ and the longitudinal and diagonal weights.

also informs all the vehicles in its local neighborhood about the new vehicle. Further details about these message passing protocols are out of scope of this paper.

Regarding the communication protocol, all vehicles periodically update their position through update position broadcasts. These broadcasts can be implemented in reality through Cooperative Awareness Messages (CAM) using DSRC or WAVE technologies (IEEE 802.11p standard). With respect to the current CAM specifications, a few parameters (e.g., the Convoy-ID) should be added to the protocols.

D. Lane keeping

Alongside the distributed formation control, we have implemented a lane-keeping behavior as a separate control law. Its goal is to detect lane markings on the road and keep the vehicle positioned centrally within the lane in which it is driving. Our lane-keeping algorithm is based on the output of one forward-looking vision sensor, mounted frontally on each vehicle. Similar to [16] which contains image smoothing using Gaussian filter, edge detection using Canny detector, and line detection with the help of Hough transform, our algorithm outputs the angle to the centerline of the lane. The vehicle steering angle ϕ_l is set to match the tangent of the centerline using a PI controller.

E. Obstacle avoidance

An obstacle avoidance behavior is also integrated in this work. Using a LIDAR mounted in front of each vehicle, any object in front is detected and localized relative to the local coordinate system of the vehicle. Similar to [17], we analyze the LIDAR's output and extract the features of the perceived obstacles and compute the angle of the nearest silhouette edge of the occlusions relative to the heading of the car. Based on this angle, the necessary deviation angle of the planned trajectory due to the obstacles is computed and named as ϕ_o. This angle is integrated in the final behavior of the vehicle.

F. Fusing behaviors

To make the vehicles perform convoy control and also lane-keeping and obstacle avoidance simultaneously, we fuse the three control laws together. The steering angle of the vehicle can be computed as:

$$\dot{\phi} = \alpha\dot{\phi}_l + \beta\dot{\phi}_o + (1 - \alpha - \beta)\dot{\phi}_g \tag{4}$$

where subscript l represents the input from the lane keeping, o from the obstacle avoidance, and g from the graph-based control law. The parameters α and β are two balancing coefficients which allow for adjusting the balance between convoy, lane keeping, and obstacle avoidance behaviors.

G. Optimizing the parameters of the proposed method

A main challenge is how to set the parameters of the composed convoy control law described above such that the performance of the vehicles in the convoy is maximized. These parameters include:

- weights of the graph-based Laplacian control law (w_x, w_d) in Eq. (3),

- balancing coefficients of convoy/lane-keeping/obstacle-avoidance behaviors (α, β) in Eq. (4).

In particular, we are interested in maximizing the performance of the system such that:

- **the overall fuel consumption** is minimized,
- **the formation errors** of the vehicles in the convoy are minimized and no collision happens.

To minimize the fuel consumption, we propose using the following performance metric for measuring the overall acceleration applied to a vehicle i:

$$g_i = \frac{1}{N}\sum_{k=1}^{N}\frac{|v_i(k) - v_i(k-1)|}{T \cdot a_M} \tag{5}$$

where $v_i(k)$ is a time-discrete speed of vehicle i, $\frac{|v_i(k) - v_i(k-1)|}{T} = \frac{|\Delta v_i(k)|}{T}$ is the acceleration term for vehicle i during a short period of T (which is the time-step of iteration k in the graph-based Laplacian control), N represents the total number of iterations in an experiment, and a_M is a constant which represents the maximum achievable acceleration of vehicle i.

To minimize the spatial formation errors of vehicle i, the following metric is proposed:

$$h_i = \frac{1}{N}\sum_{k=1}^{N}\frac{\sqrt{\left(e_i^{\parallel}(k)\right)^2 + \left(e_i^{\perp}(k)\right)^2}}{d_M} \tag{6}$$

where $e_i^{\parallel}(k)$ and $e_i^{\perp}(k)$ are respectively the longitudinal and lateral formation errors of vehicle i during discrete-time iteration k, and d_M represents the maximum possible displacement distance of a vehicle in a convoy. It should be mentioned that if one vehicle collides with an obstacle, these errors dramatically increase in all the neighboring cars naturally.

Fitness functions g_i and h_i are normalized to the interval $[0, 1]$, and they need to be minimized. We turn this minimization into a maximization problem by defining $G_i = 1 - g_i$ and $H_i = 1 - h_i$. The total fitness function in one experiment is the sum of aggregate fitness functions for individual vehicles normalized by the total number of vehicles in the convoy n:

$$F = \frac{1}{n}\sum_{i=1}^{n}(G_i + H_i). \tag{7}$$

H. Optimization method

As can be seen from Eq. (7), there is no explicit relationship between the optimization parameters (weights and balance coefficients) and the value of the fitness function. We therefore choose to exploit a robust metaheuristic optimization technique to optimize our design parameters, Particle Swarm Optimization (PSO). PSO algorithm is a population-based metaheuristic approach which is robust to the presence of noise on the input and output variables and is well suited to distributed systems. Due to its robustness and simplicity, it has been used in a wide range of applications, including multi-robot spatial control. In this work we use a noise-resistant version of PSO developed in our laboratory (see [18] for more details).

Fig. 5. The vehicle used in the simulations which is equipped with camera, LIDAR, GNSS and radio modules (left). One of the testbeds designed for simulations with more than 1 Km length (right).
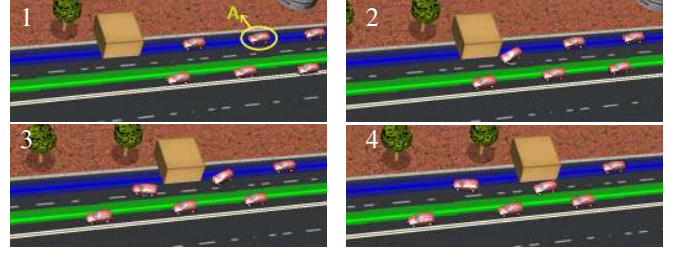

Fig. 6. Six cars maintaining convoy formation while dynamically avoiding a big obstacle (the box in yellow). The videos of this experiment are provided at the mentioned web link.

TABLE I

THE PARAMETERS USED IN THE PSO ALGORITHM, EXPLAINED IN [18]

N_p	N_i	t_e	N_{re}	w_p	w_n	V_{max}	Inertia w
8	50	12.5	1	0.1	0.1	0.2	0.8

III. EXPERIMENTS AND RESULTS

Experiments and performance evaluation of candidate solutions are carried out using Webots [19], a powerful, submicroscopic, high-fidelity simulator originally developed for mobile robotics, which has recently been upgraded to support automotive platforms[2]. Webots provides a useful framework for faithfully reproducing multiple intelligent vehicles based on the Open Dynamic Engine (ODE) [20] library. The proposed dynamic graph-based convoy controller is implemented in Webots using C++ language.

Our simulated vehicle model incorporates basic rigid dynamics properties of a real car such as typical steering dynamic response and friction of the tires. This vehicle is developed by Cyberbotics[3] Ltd. and its physical appearance is based on the BMW X5. We have added to this vehicle a radio communication device, a laser scanner, a camera and a realistic GNSS module. Fig. 5 (left) represents one of these vehicles in the Webots environment. Up to 12 vehicles are used in the various experiments of this study. We added a Gaussian noise with $\sigma = 1$ m to the GNSS output in order to obtain realistic measurements for self-localization of the vehicles. The video camera is mounted on top of the vehicle, pointing ahead, and is used for the lane-keeping behavior. It provides 15 frames per second at a resolution of 640×480 pixels. Vehicles can send and receive messages to each other using the radio communication module. We have set the maximum packet rate to 15 Hz for each vehicle. The LIDAR is a simulated Sick-LMS291 with a maximum range of 80 m and an accuracy of 10 mm (1 sigma) that is mounted on the front bumper of the vehicle, for obstacle avoidance.

We have designed several environments realistically reproducing potential highway layouts containing rings with multiple lanes, exit and merge points. Fig. 5 (right) shows one of these environments. To simplify the lane-keeping behavior, we have colored the center line of different lanes with specific colors.

The objective is to find the best values of the discussed parameters in various situations. Here is a list of scenarios which we consider in the optimizations process:
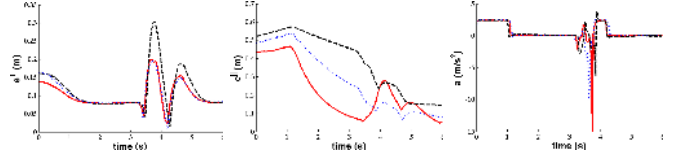
Fig. 7. Acceleration and formation errors for vehicle A in the obstacle avoidance scenario shown in Fig. 6. The obstacle avoidance starts happening at 3.3 s. Blue (dotted line) shows the result of the controller with optimal values $(w_x, w_d, \alpha, \beta) = (0.30, 0.16, 0.45, 0.42)$. Black (dashed line) and red (solid line) show the results of the controller while the parameters are set to $(0.8, 0.8, 0.1, 0.8)$ and $(0.1, 0.1, 0.1, 0.1)$ respectively.

- Convoy control while driving in a straight forward road,
- Convoy control while driving in a curved road,
- Convoy control while there is an obstacle blocking some parts of the road and vehicles need to avoid it,
- Convoy control while a vehicle joins,
- Convoy control while a vehicle changes its lane,
- Convoy control while a vehicle leaves.

Considering only one of the above scenarios will provide a set of optimized values for the parameters which may not lead to efficient functionality for the other scenarios. So we consider all of these scenarios in the optimization process and we look for a set of values which provides the best overall performance. Although the proposed method does not assume any limit for scenario parameters, to provide comparably consistent results, hereafter we focus only on two-lane convoy scenarios with $b_x = 20$ m and $b_y = 6$ m and the desired cruise speed of the convoy $v_{Gx} = 60$ km/h in all the experiments. The parameters of the PSO algorithm used in this paper are presented in Table I. The PSO performance converges after 50 iterations and its best particle is: $w_x = 0.30$, $w_d = 0.16$, $\alpha = 0.45$, and $\beta = 0.42$.

The proposed method is fully functional if proper weights (w_x, w_d, α, and β) are chosen for the vehicles' controller[4]. The optimized parameters are used in the scenarios mentioned above and the trajectory of the vehicles are observed. Vehicles' movements are notably smoother and the convoy formations are more precise while using optimized values for the parameters. Fig. 6 shows snapshots of an example scenario in which six cars are convoying in two-lanes on a highway. In this scenario, there is a large yellow obstacle in one of the lanes which cars avoid
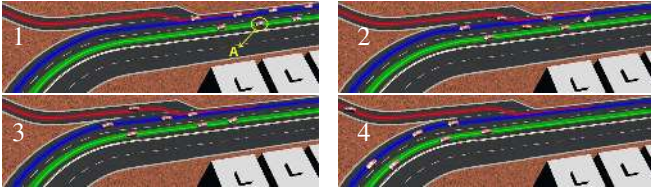
Fig. 8. Seven cars maintaining convoy formation. One car leaves the convoy and the other cars automatically adapt their position and converge to a new configuration converting to a six-vehicle convoy.
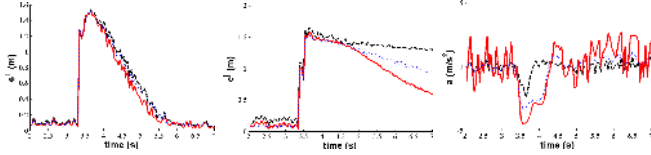

Fig. 9. Acceleration and formation errors for vehicle A in the convoy scenario shown in Fig. 8. The parameters are the ones explained in Fig. 7.

while still maintaining the convoy formation. Fig. 8 shows a few snapshots of another scenario in which one car exits from a convoy. Many similar interesting behaviors are seen in various experiments (see the video mentioned in the footnote).

To demonstrate the effect of the parameters on the convoy control, we repeated the above scenarios multiple times with different sets of values (including the optimized values). Fig. 7 and Fig. 9 show the acceleration, $e^\perp$ and $e^\parallel$ of vehicle A in the two mentioned scenarios in three experiments. In each experiment the parameters $[w_x, w_d, \alpha, \beta]$ were set to a different sets of values; once to one set of optimized values, once to $[0.8, 0.8, 0.1, 0.8]$ (called *aggressive* weight set here), and another time all to 0.1 (called *conservative* weight set). In the first scenario, the vehicles start from zero velocity, so their acceleration is positive during the first second. The obstacle avoidance starts happening at time 3.3 s. As can be seen from Fig. 7 and Fig. 9, compared to the *aggressive* weight set, the optimal values present smoother acceleration profile, comparable convergence in longitudinal error, but slower convergence in lateral error. On the other hand, although *conservative* weight set results in generally lower acceleration for the vehicles, it demonstrates bigger formation errors for longer duration. One can conclude that the optimized solution achieves a good trade-off for both minimizing acceleration changes and formation errors.

IV. CONCLUSION

We presented a distributed graph-based formation control approach for multi-lane convoying on highways, which is capable of handling dynamic changes in the formation topology while keeping lanes and avoiding obstacles. While an individual vehicle performs formation control with its neighbors only, at the convoy level, the graph is collectively connected through the sequence of the local neighborhood graphs. Vehicles can join, leave, or change lanes without impacting the functionality of the convoy. The designed formation topology is an interleaved multi-lane structure which gives enough space to the cars to use their side-lanes

while avoiding obstacles. We also investigated the impact of optimizing the parameters of the overall resulting control law in realistic scenarios using PSO in order to minimize both fuel consumption of the vehicles and formation control errors. Using optimized values for the parameters, vehicles' movements are smoother and the convoy formations are more faithful to the desired topology.

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," *IEEE Trans. on Vehicular Technology*, vol. 40, no. 1, pp. 114–130, 1991.

[2] B. Harker, "PROMOTE-CHAUFFEUR II & 5.8 GHz vehicle to vehicle communications system," in *IET Int. Conf. on Advanced Driver Assistance Systems*, 2001, pp. 81–85.

[3] E. Chan, P. Gilhead, P. Jelínek, P. Krejci, and T. Robinson, "Cooperative control of SARTRE automated platoon vehicles," in *Proc. of the Intelligent Transportation System World Congress*, 2012.

[4] 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 Cong. on Intelligent Transport Systems*, Detroit, U.S.A, DOI: 10.13140/2.1.2390.4649, 2014.

[5] R. D'Andrea and G. E. Dullerud, "Distributed control design for spatially interconnected systems," *IEEE Trans. on Automatic Control*, vol. 48, no. 9, pp. 1478–1495, 2003.

[6] 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.

[7] 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.

[8] D. Swaroop, J. K. Hedrick, and S. Choi, "Direct adaptive longitudinal control of vehicle platoons," *IEEE Trans. on Vehicular Technology*, vol. 50, no. 1, pp. 150–161, 2001.

[9] S.-Y. Han, Y.-H. Chen, L. Wang, and A. Abraham, "Decentralized longitudinal tracking control for cooperative adaptive cruise control systems in a platoon," in *IEEE Int. Conf. on Systems, Man, and Cybernetics*, 2013, pp. 2013–2018.

[10] K.-Y. Liang, "Coordination and routing for fuel-efficient heavy-duty vehicle platoon formation," KTH, Royal Institute of Technology, 2014.

[11] 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.

[12] S. Gowal, 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.

[13] R. Falconi, S. Gowal, and A. Martinoli, "Graph based distributed control of non-holonomic vehicles endowed with local positioning information engaged in escorting missions," in *Proc. IEEE Int. Conf. on Robotics and Automation*, AL, USA, 2010, pp. 3207–3214.

[14] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[15] S. Patwardhan, H.-S. Tan, and J. Guldner, "A general framework for automatic steering control: system analysis," in *Proc. of the American Control Conference*, vol. 3, 1997, pp. 1598–1602.

[16] F. Ren, J. Huang, M. Terauchi, R. Jiang, and R. Klette, "Lane detection on the iPhone," in *Arts and Technology*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Jan. 2010, no. 30, pp. 198–205.

[17] A. Sayar, O. F. Arar, S. Akpinar, S. Kosunalp *et al.*, "Approaches of road boundary and obstacle detection using lidar," in *Control and Automation Theory for Transportation Applications*, vol. 1, no. 1, 2013, pp. 211–215.

[18] E. Di Mario and A. Martinoli, "Distributed particle swarm optimization for limited-time adaptation with real robots," *Robotica*, vol. 32, no. 02, pp. 193–208, 2014.

[19] O. Michel, "Webots: Professional mobile robot simulation," *Int. Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004.

[20] R. Smith, "Open dynamics engine (ODE)," *http://www. ode. org*, 2009.