# A Modular Approach to Platooning Maneuvers

Michele Segata*†, Falko Dressler*, Renato Lo Cigno†
*Computer and Communication Systems, Institute of Computer Science, University of Innsbruck, Austria
†Department of Information Engineering and Computer Science, University of Trento, Italy
{segata,dressler}@ccs-labs.org, locigno@disi.unitn.it

***Abstract*—Driving vehicles in platoons has the potential to improve traffic efficiency, increase safety, reduce fuel consumption, and make driving experience more enjoyable. A lot of effort is being spent in the development of technologies, like radars, enabling automated cruise control following and ensuring emergency braking if the driver does not react in time; but these technologies alone do not empower real platooning. As platoons will initially share the road with human-driven vehicles, interesting new questions regarding the interactions between the two categories of vehicles arise. In this paper we briefly describe the focus of our research, i.e., the analysis of interferences caused by non-automated vehicles during a maneuver. As an example, we consider the JOIN maneuver. We define the application layer protocol to support the maneuver, together with situations that can prevent successful termination, and describe how they can be detected. We then show the validity of the idea by simulating some sample scenarios, showing either that the maneuver can successfully be performed, or safely be aborted. As final contribution, we describe our idea toward a modular approach, i.e., the development of complex maneuvers by combining smaller sub-maneuvers, aiming to ease development and safety analysis.**

## I. INTRODUCTION

Better road usage and increased safety will pass through the capability of vehicles to implement cooperative driving, *platooning* for short. Albeit recently there has been a strong focus on autonomous or semi-autonomous driving [1], where Inter-Vehicular Communication (IVC) is not needed, only platooning, which requires fully developed IVC, can guarantee improved road safety, while increasing the infrastructure usage and reducing fuel consumption [2].

Platooning is much more than simple car following. Platoons must be built and split, vehicles must be able to join and leave, the platoon leader must be changed, e.g., because the driver is tired or has reached his destination. All the possible maneuvers must be supported by a proper application level protocol, providing the communication primitives or Application Programming Interface (API) needed to implement them. Indeed, this is only the starting point, as the API must provide also the means to cope with impairments, unexpected situations, partially failing communications, interfering vehicles, and finally also the emergency maneuver to relinquish the vehicles' control to all the drivers safely in case there are no more the conditions to operate the platoon.

In this paper, we briefly analyze the application layer protocol for a join maneuver, which is able to handle interferences by human driven vehicles. We implement the protocol into our platooning extension for Veins [3], [4], and show how it performs in two sample scenarios. Finally we describe how we intend to further tackle this problem in our future research, i.e., moving toward a modular approach.

## II. RELATED WORK

The scientific community investigated different ways to perform maneuvers in an Automated Highway System (AHS), both with and without the infrastructure cooperation. One approach assuming infrastructure cooperation tackles the problem from a control theoretic point of view, defining the laws to control the vehicles during the maneuvers, together with higher layer mechanisms to the cars involved [5], [6].

Another high level approach is presented in [7]. The authors describe a set of different communication patterns that can be used in order to exchange data while performing a maneuver. Moreover, they define a set of controllers each of those responsible for a different situation. For instance, there is a controller dedicated to obstacle avoidance. However, not much details on how a particular fault should be detected and communicated to other parties are given.

Other works focus on mechanical and network fault handling, investigating how to detect and to react to them in order to minimize risks [8] or performance loss [9].

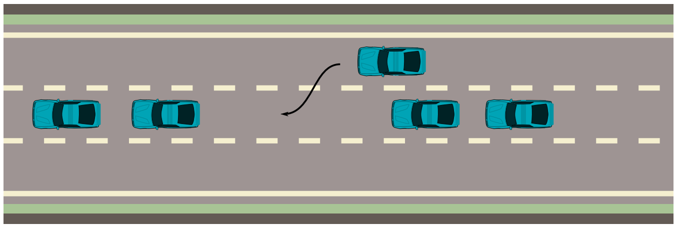
Recently, mixed highway scenarios have gained more attention [10]. The aim is to make platoons able to travel on public roads, avoiding the deployment of dedicated infrastructure. This poses new challenges that need to be addressed due to the presence of human drivers which might interfere with platooning operations. In [11], for example, the authors study mechanisms to perform cooperative maneuvers (e.g., a lane change by an entire platoon) to avoid dangerous situation.

In this context, the challenge of defining an application level protocol that support the different maneuvers, seen as different applications, has not been tackled to the best of our knowledge. The identification of external events due to the presence of other road users, or to other impairments as communication faults, and the algorithms that the applications deploy to react to the situation are extremely important to make platooning safe and acceptable by the broad public.

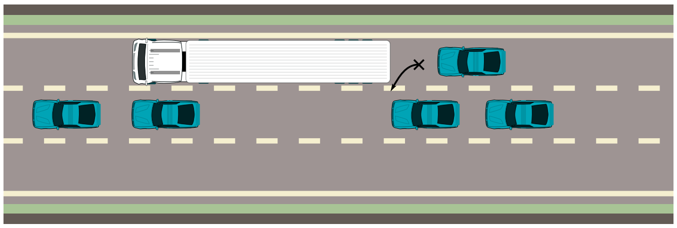
## III. MANEUVERS AND SCENARIOS

To properly support platooning, a set of required maneuvers needs to be implemented. The first and most studied one is the FOLLOW maneuver, i.e., standard cruising, where interesting issues on multi-body control have to be solved and that represent the steady-state of a platoon. From a communications perspective FOLLOW can be realized with standard DSRC/WAVE beacons; a working version implementing the controller defined in [6] is available in an extension of Veins simulator we use for evaluation [3], [4].

From a protocol point of view the maneuvers to form and to manage a platoon are more challenging, e.g., JOIN, LEAVE,



(a) Normal procedure



(b) Interference by slower vehicle in front

Figure 1. Graphical sketch of different situations for the JoinMiddle maneuver. Automated cars shown in dark color.

Merge, and Split require a more sophisticated coordination among cars than simply receiving beacons from the other cars in the platoon. Moreover, they have additional parameters, e.g., the position in the platoon where a car wants to join.

Here, we are instead interested in shedding some insight on platoons management and the communication challenges they pose, specially in face of "external threats", such as human driven vehicles interfering with the maneuvers.

As a representative of management maneuvers in this paper we focus on the Join procedure, assuming that one car joins the platoon in the middle, which is clearly more challenging than joining at the head or at the tail of the platoon. Besides considering the plain procedure, we also include in the protocol "escape" procedures, to handle cases when there are interferences by human-driven vehicles. For the sake of simplicity the escape is just aborting the maneuver and returning to normal platooning.

An example of a Join maneuver is shown in Figure 1. In the standard setup (Figure 1a), a vehicle creates a gap to let another one in. A slower human-driven vehicles may however be encountered while approaching the platoon which prevents the joiner to conclude the maneuver (Figure 1b). This situation must be detected and reported to the high layer logic which should decide what is the best action to undertake.

In this paper we consider two specific scenarios. In an extended version of this work we considered more situations, but for the sake of brevity we have chosen a subset and tried to focus on the idea:

- Scenario 1 (far truck interference): the joining vehicle encounters a truck on the lane where it is trying to join, but the truck does not prevent the conclusion of the maneuver as it is far enough.
- Scenario 2 (close truck interference): as for Scenario 1, a slow truck obstructs the joining vehicle, but this time it is forced to abort to avoid a collision.

We assume that vehicles are controlled and travel as envisioned in the SARTRE project [10]: drivers instruct the vehicle, which are otherwise entirely autonomous, through a Human Machine Interface (HMI). Actions like steering or touching the brakes disengage the Cooperative Adaptive Cruise Control (CACC) and lead to the platoon split. How this happens, however, is out of the scope of this paper.

All platooning-capable vehicles are equipped with an IEEE 802.11p compliant device, a GPS receiver, and a radar. The CACC, in order to safely perform automated close-following, needs acceleration and speed values of a subset of vehicles in the platoon. Such subset depends on the design of the controller itself. We adopt the controller designed during the PATH project [6], where each vehicle requires acceleration and speed of the platoon's leader and the vehicle in front. Other designs, with different characteristics, are possible [9], but do not influence the maneuvers we focus on.

All protocols are implemented on top of standard broadcast beacons transmitted at 10 Hz as commonly required [12]. "Unicast" messages are obtained by identifying the intended recipient at the application level, with a proper tagging in messages, which however can be read by any other vehicle, adding redundancy and reliability to the system.

The number of events that can interfere with platoon maneuvering are humongous, but here we only consider the one envisaged in the scenarios already described: the goal of this work is verifying the feasibility of automatic maneuvering controlled via a standard DSRC/WAVE vehicular network environment in a mixed scenario, and we do not pretend to make an exhaustive study.

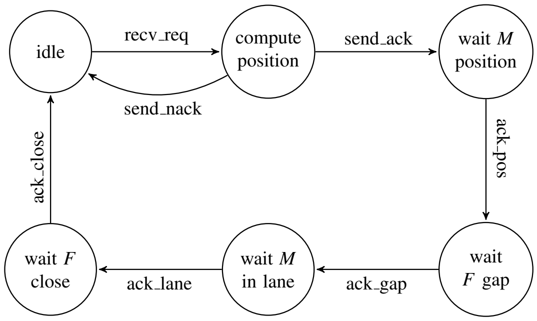
We think that the scenarios we consider (Figure 1b) can be very common in case of platooning cars, which travel faster than trucks. Note that whether the truck is equipped with communication devices or not is irrelevant: it will in any case interfere with the maneuver. We want to explore if implementing proper reactions to this situation, i.e., completing the maneuver if the truck is far enough or abort it if the truck is too close, is feasible and if the situations are distinguishable with the on-board sensing (the radar).

There are other situations which can prevent the successful termination of the maneuver, e.g., when an unauthorized human-driven car enters the platoon, or when a network fault occur. We addressed these problems but we intentionally omit them in here, as the aim is mainly to show future research directions.

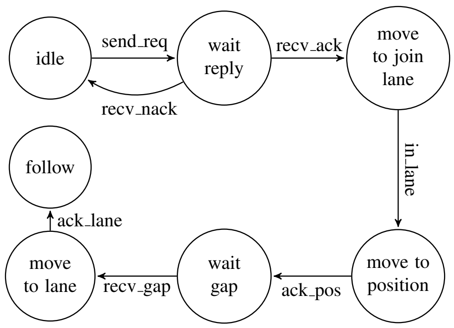
## IV. Join Application Protocol

Consider the Join maneuver, in particular with a car entering in the middle of the platoon. Three vehicles are "actively" involved in the procedure. The joiner $M$ sends a join request to the leader $L$, which replies back with the position at which $M$ is supposed to join. $M$ then moves into position on one of the two lanes adjacent to the platoon. When $M$ is in the position indicated by $L$, car $F$ opens a gap to let $M$ in. $M$ and $F$ close their gaps and the procedure terminates.

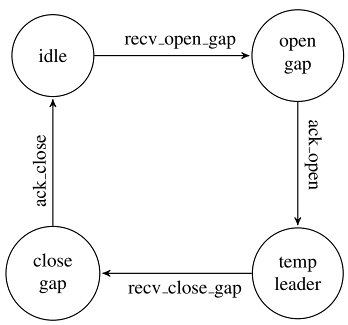
All packets sent for notification, i.e., to perform state changes, must be reliably transmitted at least to the vehicle that has the active role in the maneuver. This is obtained including in the broadcast beacons the identity of the intended recipient, which will return an application layer acknowledgement enabling the




(a) Leader $L$




(b) Joining vehicle $M$




(c) Follower $F$

Figure 2. State machines for the vehicles involved in a JOINMIDDLE maneuver. No fault/misbehavior detection included.

detection of lost packets and possibly triggering retransmissions. It is conceivable to achieve the same goal by using IEEE 802.11p unicast frames. This possibility, however, has the drawback that the other cars do not receive this message, so they miss part of the information about the maneuver status.

The state machines at the different vehicles that define this JOIN protocol are shown in Figure 2. We only represent the maneuver itself for the sake of clarity, without including all the details to detect faults and impairments and the actions taken to counter them: considering every possible fault or impairment is more a task for a standard specification than for a proof-of-concept prototype. In our implementation when the maneuvers cannot be completed as intended, it is simply aborted, i.e., $M$ does not join the platoon. The 'idle' state corresponds indeed to the steady state platooning for all the cars but $M$, which until has received the positive join acknowledgement from $L$ remains human driven. At the end of the procedures all car return to the steady state 'idle' platooning, thus for the joiner $M$ entering the 'follow' state of this procedure means becoming a normal follower car.

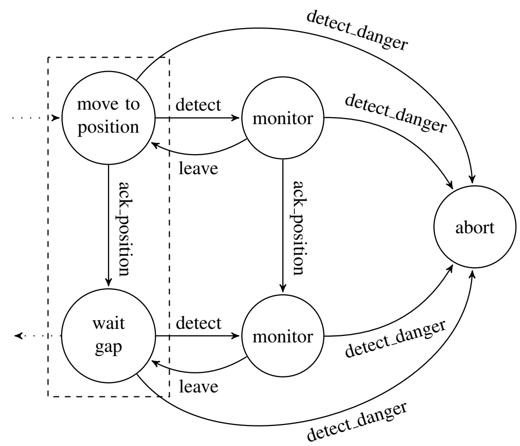



Figure 3. State machine for the detection of a slow vehicle in front.

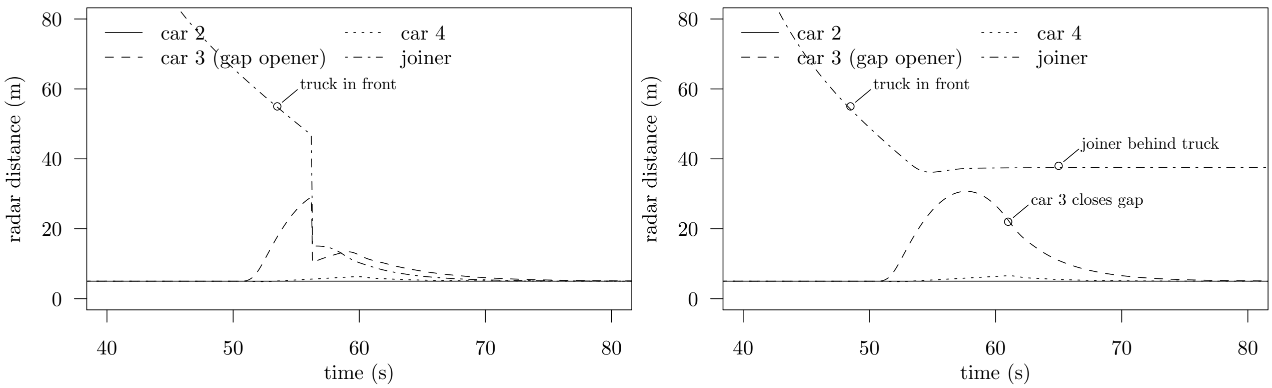
We now extend the state machine to cope with the situation in which $M$ detects a vehicle in front while trying to get in the correct position to join the platoon. This can happen during the "move to position" and the "wait gap" statuses of the state machine in Figure 2b. To handle this case we can extend the state machine of $M$ as shown in Figure 3, where the two states enclosed in the dotted line are the same states of Figure 2b. When $M$ detects a vehicle in front, it first switches to the "monitor" state. The radar can indeed detect objects which are up to 200 m to 300 m distant, which do not immediately interfere with the maneuver. Whenever a dangerous situation is detected, e.g., the Adaptive Cruise Control (ACC) is mandating to decelerate to avoid a collision, then the maneuver is aborted.

Notice that being in the "monitor" state does not prevent to continue the maneuver. If $M$ is able to move to the join position, and the vehicle in front does not endanger maneuver's safety, it can continue waiting for $F$ to open the gap and, in case, successfully complete the maneuver.

## V. IMPLEMENTATION AND EVALUATION

For the protocol evaluation, we implemented it into the platooning enabled extension of Veins [3] and test it the aforementioned scenarios. To show the validity of our approach, we implement anomaly detection mechanism connected to basic countermeasure procedures. In particular, to detect the presence of a slow vehicle in front, we exploit data obtained from the radar, and compute the acceleration that the ACC would apply. If the deceleration becomes greater than $3\,\mathrm{m/s^2}$, then the system issues a warning. The countermeasure connected to this warning is to make the joiner $M$ send an abort message to the leader, disabling CACC and switching back to ACC.

We analyze the maneuver in the different scenarios from a vehicle dynamics point of view. Plots in Figure 4 show the dynamics of the vehicles in the platoon, plus the dynamics of the joiner $M$. The figures plot the distance from the vehicle in front as perceived by the radar.




(a) Scenario 1

(b) Scenario 2

Figure 4. Vehicles dynamics for the different scenarios showing radar distance measured by car $x$.

We start with the analysis of Scenario 1 (Figure 4a). The plot shows how the maneuver is correctly performed. The joiner $M$ approaches the platoon from the side, and when in position, $F$ and car $4$ slow down to open a gap. The joiner $M$ detects a truck in front, as shown by the radar trace, but it is far enough to let $M$ in. The gap is then slowly closed, and the procedure terminates.

In Scenario 2 (Figure 4b) instead, $M$ reaches the join position, $F$ starts to open the gap, but $M$ has to abort to avoid a collision. At this time, $M$ switches to ACC and remains behind the truck, while $F$ closes the gap and the platoon continues to drive as before.

These results show how the protocol can be easily extended to detect and react to anomalies in the procedure. Using the same approach, we can develop the state machines for other interferences, or for other maneuvers, analyze possible weaknesses, and study how to tackle them.

## VI. Conclusion – Toward a Modular Approach

This work has proposed and analysed an application level protocol to support Join maneuver in two sample scenarios, showing that relatively simple logic can support complex maneuvers as letting a vehicle join a platoon in the middle of the same, while guaranteeing that in case of interference the maneuver can safely be aborted. The state machines of Figure 2, however, do not handle all possible situations that might occur, and as previously mentioned they need to be extended. When including all possible kind of interferences and network faults, state machines might become very large, and thus their verification become difficult.

Consider the Join maneuver we described beforehand. We can split it as follows: $M$ performs a LaneChange followed by a LargeDistanceFollow of the car in front of $F$, then $F$ should OpenGap, $M$ must JoinAtBack the first sub-platoon, and $F$ should finally CloseGap. Now imagine that $M$ wants to leave the platoon. $M$ and $F$ should OpenGap, $M$ leaves invoking LaneChange, and the platoon can continue after $F$ performs CloseGap. Moreover, some of these sub-maneuvers can further be split, e.g., JoinAtBack or OpenGap. The latter can be performed by combining LeaderChange with LargeDistanceFollow.

We think that the smaller is the procedure, the easier is its design and its verification. Our future work thus includes a clear definition of a basic set of maneuvers that can be composed in order to perform more complex ones.

## References

[1] T. Luettel, M. Himmelsbach, and H.-J. Wuensche, "Autonomous Ground Vehicles–Concepts and a Path to the Future," *Proceedings of the IEEE*, vol. 100, no. 13, pp. 1831–1839, May 2012.

[2] A. Davila, E. Aramburu, , and A. Freixas, "Making the best out of aerodynamics: Platoons," in *SAE Technical Paper 2013-01-0767*, 2013.

[3] M. Segata, F. Dressler, R. Lo Cigno, and M. Gerla, "A Simulation Tool for Automated Platooning in Mixed Highway Scenarios," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 16, no. 4, pp. 46–49, October 2012.

[4] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, 2011.

[5] R. Horowitz and P. Varaiya, "Control Design of an Automated Highway System," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 913–925, July 2000.

[6] R. Rajamani, H.-S. Tan, B. K. Law, and W.-B. Zhang, "Demonstration of Integrated Longitudinal and Lateral Control for the Operation of Automated Vehicles in Platoons," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 4, pp. 695–708, July 2000.

[7] F. Michaud, P. Lepage, P. Frenette, D. Letourneau, and N. Gaubert, "Coordinated Maneuvering of Automated Vehicles in Platoons," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 437–447, December 2006.

[8] R. Rajamani, A. Howell, C. Chen, J. Hedrick, and M. Tomizuka, "A Complete Fault Diagnostic System for Automated Vehicles Operating in a Platoon," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 4, pp. 553–564, July 2001.

[9] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer, "Graceful Degradation of CACC Performance Subject to Unreliable Wireless Communication," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. The Hague, The Netherlands: IEEE, October 2013.

[10] C. Bergenhem, Q. Huang, A. Benmimoun, and T. Robinson, "Challenges of Platooning on Public Motorways," in *World Congress on Intelligent Transport Systems*, Busan, Korea, October 2010.

[11] S. Lam and J. Katupitiya, "Cooperative Autonomous Platoon Maneuvers on Highways," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2013)*. Wollogong, Australia: IEEE, July 2013, pp. 1152–1157.

[12] J. Ploeg, B. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, "Design and Experimental Evaluation of Cooperative Adaptive Cruise Control," in *IEEE International Conference on Intelligent Transportation Systems (ITSC 2011)*. Washington, DC: IEEE, October 2011, pp. 260–265.