CISTER

Research Centre in
Real-Time & Embedded
Computing Systems

Conference Paper

Towards a Cooperative Robotic Platooning Testbed

Ênio Filho*

Nuno Guedes*

Miguel Mestre

Bruno Vieira

Ricardo Severino*

Anis Koubâa*

Eduardo Tovar*

*CISTER Research Centre

CISTER-TR-200311

2020/04/15

Towards a Cooperative Robotic Platooning Testbed

Ênio Filho*, Nuno Guedes*, Miguel Mestre, Bruno Vieira, Ricardo Severino*, Anis Koubâa*, Eduardo Tovar*

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP P.Porto)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: enpvf@isep.ipp.pt, 1140477@isep.ipp.pt, 1160930@isep.ipp.pt, bffbv@isep.ipp.pt, sev@isep.ipp.pt, aska@isep.ipp.pt, emt@isep.ipp.pt

https://www.cister-labs.pt

Abstract

The deployment of information and communication technologies in vehicles and into the transportation infrastructure in general, holds the promise of significant improvements to traffic safety and efficiency. The ETSI ITS-G5 standard presents itself as a viable and already available solution, to enable such intelligent social and mobility scenarios in the near future, including cooperative and autonomous vehicle platooning. However, the usage of wireless communications in safety-critical scenarios poses several challenges, and their reliability and safety must be adequately tested and validated. To do this, the safety concerns and cost of relying on real vehicles is prohibitive for early deployments. A solution lies in the use of robotic platforms, since these are relatively cheaper and allow to partially test real platforms and components, as well as different control mechanisms. This work presents the development of a 1/10 scale Cooperative Platooning Robotic Testbed with such aim. Real ITS-G5 On Board Units (OBU) were integrated in the vehicles for communications support and a cooperative control algorithm that solely relies on communications was successfully implemented.

Towards a Cooperative Robotic Platooning Testbed

Enio Vasconcelos Filho[§], Nuno Guedes[§], Bruno Vieira[§], Miguel Mestre[§],
Ricardo Severino[§], Bruno Gonçalves [†], Anis Koubaa [§ ‡], Eduardo Tovar[§]
[§]Cister Research Centre ISEP, Polytechnic Institute of Porto Porto, Portugal
[†] GMV Innovating Solutions, Portugal. [‡] Prince Sultan University, Saudi Arabia.
[§](enpvf, 1140477, bffbv, 1160930, rarss, emt)@isep.ipp.pt [†] bruno.goncalves@gmv.com, [‡]

***Abstract*—The deployment of information and communication technologies in vehicles and into the transportation infrastructure in general, holds the promise of significant improvements to traffic safety and efficiency. The ETSI ITS-G5 standard presents itself as a viable and already available solution, to enable such intelligent social and mobility scenarios in the near future, including cooperative and autonomous vehicle platooning. However, the usage of wireless communications in safety-critical scenarios poses several challenges, and their reliability and safety must be adequately tested and validated. To do this, the safety concerns and cost of relying on real vehicles is prohibitive for early deployments. A solution lies in the use of robotic platforms, since these are relatively cheaper and allow to partially test real platforms and components, as well as different control mechanisms. This work presents the development of a 1/10 scale Cooperative Platooning Robotic Testbed with such aim. Real ITS-G5 On Board Units (OBU) were integrated in the vehicles for communications support and a cooperative control algorithm that solely relies on communications was successfully implemented.**

***Index Terms*—Cooperative Platooning; Robotic Testbed; Intelligent Transportation Systems;**

I. Introduction

The Vehicular Platooning ITS (Intelligent Traffic Systems) scenario is becoming an increasingly hot topic considering its promise for improving traffic safety, efficiency and achieving reduced fuel emissions, by exploring the advantages of having groups of vehicles moving closer together. In these scenarios road capacity is increased, while in parallel, traffic congestion declines. For the same reason, emissions fall, due to the reduction of the air resistance, [1], even in high traffic scenarios [2]. This positively impacts several societal aspects, generating an overall improvement in quality of life [3]. However, the literature is clear in pointing out the absolute relevance of supporting vehicle to vehicle (V2V) as well as vehicle to infrastructure (V2I) communications, to optimize the control of these agents, supporting what is known as Cooperative Vehicular Platooning (CoVP) [4], where each following vehicle uses information from its own in-vehicle sensors, in addition to data received from the leader vehicle, to cooperatively measure and adjust its position, based on the speed, direction and acceleration of the preceding vehicle. However, just as V2X communication can improve platooning safety [5], via the introduction of an additional information sources, its usage also raises some challenges concerning the reliability and security of communications and its impact on traffic safety and efficiency [6], [7]. In fact, several negative impacts can be observed in platooning control given some communications problems, such as packet loss, and transmission delay [8]. Therefore, there is the need to implement monitoring tools or safety mechanisms in CoVP, as proposed in [9], to increase the safety of these systems. Importantly, extensive testing and validation must be carried out to understand the safety limits of these systems.

This work was partially supported by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology), within the CISTER Research Unit (UIDB/04234/2020); by FCT and the EU ECSEL JU under the H2020 Framework Programme, within project ECSEL/0002/2015, JU grant nr. 692529-2 (SAFECOP); also by the Operational Competitiveness Programme and Internationalization (COMPETE 2020) under the PT2020 Partnership Agreement, through the European Regional Development Fund (ERDF), and by national funds through the FCT, within project POCI-01-0145-FEDER-032218 (5GSDN).

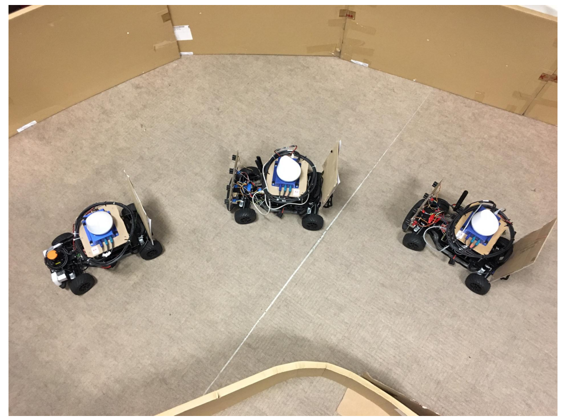
Fig. 1. Platooning Testbed running

With this objective, different approaches can be followed. Relying on simulation software [10], [11] is the most common and economic approach for analysing these issues. However, although the ability to easily scale the system can be considered as a significant advantage, the fact that these do not encompass the processing characteristics or constraints of real platforms reduces its effectiveness. To partially address this, several efforts have been carried out to integrate simulations with Hardware in the Loop (HiL) [12] and even to deploy in real cars [13]. However, such tests are expensive and difficult to escalate. In the middle-ground of such test and validation options, robotic testbeds appear as a good solution, considering that due to its flexibility, they can integrate with different platforms that are to be deployed in vehicles, can be deployed indoor in controlled environments, and can partially replicate a realistic scenario at a fraction of the cost of a real vehicle.

In this work, we present a 1/10 scale Cooperative Robotic Testbed Platform (RoboCoPlat), which integrates with ETSI ITS-G5 [14] embedded communications OBUs. The proposed testbed supports the deployment of different test scenarios in a indoor or outdoor environment, in a reduced area.

With such testbed it will be possible no only to test and evaluate different control algorithms and the communications performance, but also to test, validate and demonstrate new mechanisms that can improve the behaviour of the system, while implementing them in platforms that are much closer to a real vehicle. This will enable us to deduce safety measures from tests in a controlled environment, in multiple path configurations, and with the possibility to add new vehicles at a relatively low cost, compared with real cars.

The main contributions proposed in this work are: (1) The development of a flexible robotic testbed platform called RoboCoPlat, which can integrate with different technologies and sensors; (2) Integration of an ETSI ITS-G5 OBU to enable V2V communications; and (3) implementation and test of a cooperative platooning control model. To our best knowledge, there are no ITS-G5 enabled robotic platooning testbeds in the research community or otherwise, designed to develop, test and validate cooperative platooning and enabling technologies.

II. Related Work

There are several works on vehicle platooning, however, few instantiate their proposals over real hardware deployments. In this section we focus on practical implementation, particularly on robotic testbeds. The authors of [15] developed a low-cost testbed that can be implemented in different models of vehicles in order to test different control algorithms to follow trajectories autonomously. The proposed testbed does not support V2X communications, but instead a communication link to monitor the vehicle status. Thus, in this work, the only platooning enabler are the on-board sensors. The testbed developed in [16] presents the same limitations. The main focus is pointed at the evaluation of the specific components of each vehicle and not on the communications or interactions between the vehicles. Another robotic testbed platform is used in [17]. This testbed relies on the HoTDeC hovercraft, developed at the University of Illinois. It allows the implementation of different control models in order to simulate vehicles and their behaviors, sharing data between them. This platform is quite flexible and has been used in different projects, allowing some tests with cooperative driving. However, the vehicle dynamics are quite different from a traditional car, and the communications have no similarity with ETSI ITS-G5 standard or any other communication technology that can be considered a candidate to support these systems. This project uses a camera as a central controller to define the position of each vehicle and send information to the vehicles using WiFi with ZeroMQ messaging system. Choosing the right communications technology is of great importance considering it plays a decisive role on the performance of the control system. A similar testbed is developed in [18], using vehicles in a scale of 1:14 for trucks and 1:10 for passengers cars. The authors present a small-scale testbed for automated driving, that also allows the implementation of different control strategies, even for platooning. However, the implemented platooning in this testbed is not cooperative and only relies on local information.

At Arizona State University, a group of researchers [19] developed the vehicular cloud robots (VC-bots) testbed, which aimed at enabling an open platform for both research experiments and education services on VANET, vehicular cloud computing infrastructures and future smart vehicles applications. In this work, the vehicles are setup from different robotic platforms in order to simulate different models of cars. This platform is quite flexible, allowing the development of different cooperative platooning strategies [20]. However, the communication between the vehicles is based on WiFi networking, what is significantly different from the ITS-G5 standard for vehicles communications. This project features separate control systems for the longitudinal and lateral control. Longitudinal control is enabled by WiFi communication, while the lateral control is performed by means of camera vision. Instead we would like to have fully communications-assisted platooning, longitudinal and lateral.

In [21], the authors developed a system that uses 5G ultra reliable and low-latency communications (uRLLC) for deploying cooperative tasks. In this work, the objective was to design a V2X communication platform, that allowed flexible reconfiguration within a short frame structure, rapid real-time processing, and flexible synchronization. This system was integrated in an autonomous vehicle in order to test cooperative driving scenarios, such as semi-simultaneous emergency brake. However, this testbed is limited, as it only targets the communications platform and it is oblivious of the potential impacts upon a cooperative controller.

In contrast to the above mentioned works, our testbed provides clear advantages: (1) it relies on ROS for enabling new sensors and platforms integration, which increases its flexibility and reconfiguration options, and its integration with simulation software. This allows the initial development of a control model in a simulator over a ROS environment, and to bring it to life in the robotic testbed in a comprehensive and continuous integration effort; (2) it integrates a true communications OBU (ETSI ITS-G5) which will enable the field trial of different communication scenarios in parallel with different cooperative control algorithms, to better study its inter-dependencies in terms of safety; (3) it is cheaper than any other deployment with real-size autonomous vehicles, thus the number of vehicles can easily be increased; and (4) it is highly portable, and can be easily deployed in a new indoor or outdoor environment, in different track configurations. The current version of RoboCoPlat with three cars (a leader and two followers) is presented in Figure 1.

III. Robotic Testbed: Vehicle Structure

Each vehicle of our testbed is based on the F1tenth vehicle architecture [22], an open-source autonomous cyber-physical platform, with some additional sensors. This high-performance autonomous vehicle architecture was designed as a means to short-circuit the access to autonomous driving deployment and validation via an affordable vehicle solution with realistic dynamics i.e. Ackermann steering and ability of traveling at high speeds i.e. above 60 km/h. Besides this shared objective, this is a proven architecture that has been also supporting the

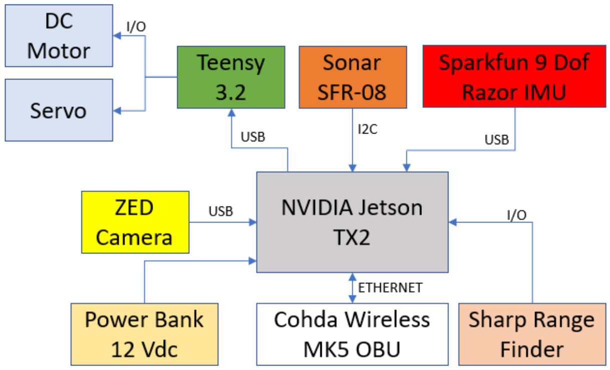

Fig. 2. Hardware Architecture

F1tenth autonomous racing competition of 1/10 scale racing vehicles, on which teams are invited to test their perception, navigation and control algorithms, in time trial or head to head competition formats.

A. System Architecture

We use the Traxxas RC Ford Fiesta ST as base vehicle for integrating all the platforms, sensors and actuators to enable an fully autonomouss vehicle. The RoboCoPlat architecture is presented in Figure 2, and is replicated among all the vehicles, with exception of the first vehicle which also features a Lidar for enabling improved SLAM capabilities. In this architecture, the central component is the Nvidia Jetson TX2 [23], that is a fast, power-efficient embedded AI computing device. This 7.5-watt computing platform, features a 256-core NVIDIA Pascal GPU, 8GB of DDR memory and 59.7GB/s of memory bandwidth. It has a eMMC 5.1 storage with 32 GB and a Dual-Core NVIDIA Denver 2 64-Bit CPU and also a Quad-Core ARM®Cortex®-A57 MPCore. This processing component is responsible for computing all the data input e.g. from sensors and OBU, and applying the developed algorithms. As this element doesn't provide a direct interface to the vehicle's motor and servo, we setup a Teensy 3.2 to convert the speed and steering angles of the vehicle into PWM's signals to actuate on the motor and servo i.e. for speed control and direction. A Cohda Wireless MK5 OBU is also integrated in the architecture via and ethernet connection to the Jetson TX2. More details about the integration will be provided next.

The operating system running on the Jetson TX2 is Linux Ubuntu 16.04.6 Xenial. The ROS-based system implements the processing pipeline to enable the execution of the platooning algorithms, by relying on additional ROS packages such as Zed Python API, Vision OpenCV and Razor IMU 9dof. The zed python mechanism is responsible for providing camera image processing which is latter used to enable visual odometry. This architecture is presented in figure 3, where the SRC container has the principal nodes designed to control the movement of the vehicles. The communication with the peripherals is provided by *Serial Talker* and *Range Finder* nodes, while the *Car State* node collects the data from the sensors and computes the position of the vehicle. There is an specific node that calculates the angular speed of the vehicle, using the information provided by the IMU, *Angular Speed* and a node responsible for the platooning control of the vehicle, *Platooning Controller*, on which we can implement different algorithms.

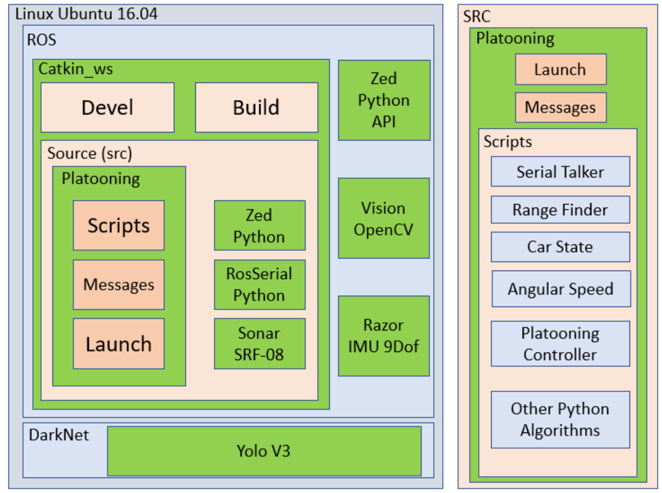

Fig. 3. Software Architecture

B. Control Algorithm

The platooning model implemented in this project follows the same approach as the one developed in [11]. Its simplicity, and the fact that it solely relies on communications to achieve platooning, seemed interesting to better explore in the future the communications' performance impact over the platooning service. The platoon is constituted by a Global Leader (GL), which is manually driven, and by the follower vehicles (F), so that for each vehicle besides the GL there is the possibility to be both a local leader (LL) and a follower. In other words, in a platoon composed of a Global Leader V_0, and by several identical autonomous vehicles V_i with $i \in [1, n]$, V_0 is responsible for establishing the trajectory and speed of the platoon. A vehicle V_i is a follower of the vehicle V_{i-1} and a local leader of the vehicle V_{i+1}, as presented in Figure 4.

As presented in [11], the controller is divided into longitudinal and lateral control. Both are based on PID controllers, where the longitudinal one is responsible to keep the inter distance between V_i and V_{i+1} using the the global coordinates, (latitude and longitude, x_i, y_i) and the speed (s_i) of the local leader, and the global coordinates and speed of V_{i+1}. The lateral controller is responsible for the adjust of the follower orientation through steering action. The lateral controller of V_{i+1} compares his heading, h_{i+1} with the heading of the local leader, h_i in time t. The PID controller was suitable in this application given the almost constant speed of the vehicles during most of the circuit. Also, the speed adjusts are not very abrupt, keeping the acceleration constraints of the system.

The controller is one of the components of the software architecture presented in figure 3. As a multi-use testbed, the controller of RoboCoPlat can be replaced by any other control model in future works.

C. Inter-vehicle Communication Architecture

To enable V2V communications we integrated Cohda Wireless MK5 [24] ETSI ITS-G5 compliant OBUs in the vehicles. Each MK5 OBU in vehicles transmit data to other OBUs

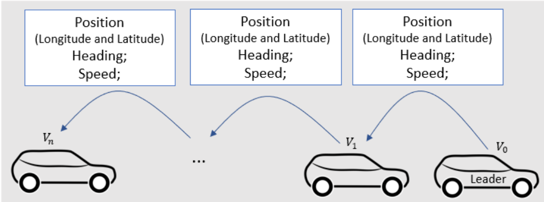

Fig. 4. Communication Model

in a defined range using through broadcast messages. The integration between the Jetson TX2 and the OBU system was carried out by developing a MK5/ROS Bridge software module, taking advantage of the ROS publish/subscribe architecture, using TCP sockets. This structure, presented in Figure 5, was called MK5/ROS bridge, and provides a bi-directional communication between ROS systems, whether they are simulated or not, and the MK5 OBUs. On the OBUs side, the received messages from the ROS topic containing the vehicles information, including its position, heading and speed, are processed by the bridge and passed to the Message Broker, which then handles it, and passes the relevant information to the ETSI modules, to fill in the necessary information at the standardized message containers. On the follower's side, the platooning system subscribes to the topic published by the ROS/MK5 bridge, and upon message reception, captures the position of the other elements of the platoon, including its local leader.

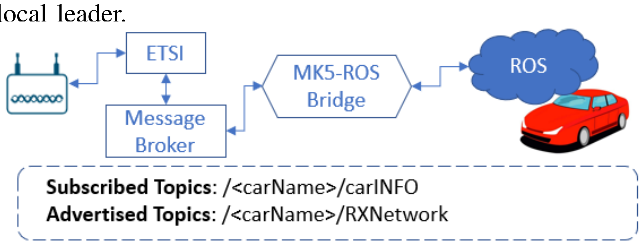

Fig. 5. MK5/ROS Bridge

Exchanged messages between the OBUs in RoboCoPlat are periodical and inform the current position and status of its source ITS host. In this setup, messages are sent in a $5Hz$ frequency, that can be adjusted in different scenarios. The Protocol Data Unity, PDU, of the messages can be observed in Figure 6. This figure presents the basic structure of the messages that are transmitted by the vehicles. The *basic container* indicates the sender of the message and is set with the OBU configuration. The *high frequency container* contains the vehicle information that will be used in the platooning controller.

Basic Container		High Frequency Container		
ITS HEADER	VEHICLE ID	POSITION: Latitude and Longitude (m)	HEADING: Roll, Pitch, Yaw (°)	LINEAR SPEED (m/s)

Fig. 6. Message Container

D. Localization

Although the integrated OBUs feature an embedded GPS module, the main objective was to deploy the testbed in a controlled indoor environments, without requiring external localization sources to the vehicle. Thus, in order to implement cooperative platooning, it was necessary to compute the spatial position and orientation of the vehicles, as such information is passed to the follower vehicle so that it is able to repeat the leader's trajectory. Although the leader is fitted with a Hokuyo10LX Lidar for improved performance, to keep the cost of remaining vehicles low, we decided not to implement Lidar on the follower vehicles. Instead, the current version of RoboCoPlat uses the ZED stereo camera [25] visual odometry and an IMU [26] for detecting vehicle position and speed. In order to increase the precision of the odometry, a fusion of the data provided by those sensors was done using an Extended Kalman Filter (EKF). Several laps were carried out on an oval track to assess the effectiveness of the solution. At the end of each lap, the actual position of the vehicle was compared with the position indicated by ego vehicle. As expected, this error increases with the number of laps if one relies only on visual odometry. The obtained results presented in figure 7 show a minor error of a few centimeters for this solution. This test was performed with a mean of the obtained values of 10 tests of 5 laps each.

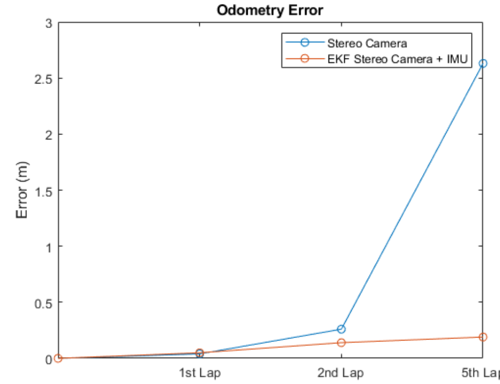

Fig. 7. Odometry Error

IV. Experimental Validation

In order to test the overall cooperative platooning system, an example scenario is outlined and demonstrated in figure 8. The local leader travels on a designed path at constant speed of $1.0m/s$, while continuously providing to the followers, via the OBUs, relevant information such as linear and angular position, speed and steering angle as presented in Figure 6. The follower receives this information and uses it to adjust its longitudinal and lateral motion, keeping a safe distance to the leader. The target distance between the vehicles is set as $3.0m$. The initial distance between the cars is $2.0m$. The global position of the vehicles is defined in terms of cartesian coordinates that represents latitude and longitude of them.

For the sake of analysing the system's response and precision, we setup the control system to carry out synchronized distance and orientation adjustment, meaning the follower vehicle follows its leader by maintain the target distance and applying the necessary lateral corrections to mimic the behavior of the preceding vehicle in terms of orientation and speed. Figure 9 depicts the path traveled by the leader and the follower (car_2). It is possible to observe that the follower keeps the distance for the leader, adjusting its position with an average error close to $0.5m$. The distance between the

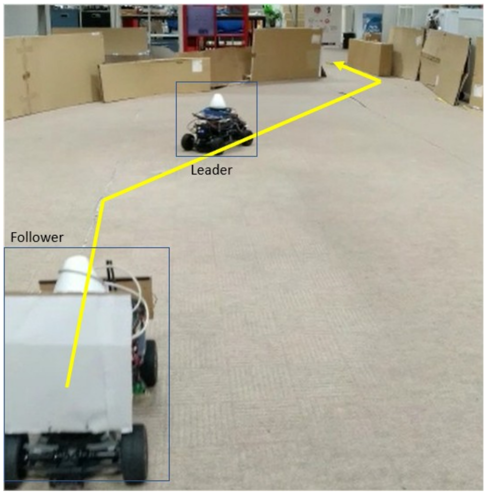

Fig. 8. *Platoon's path*

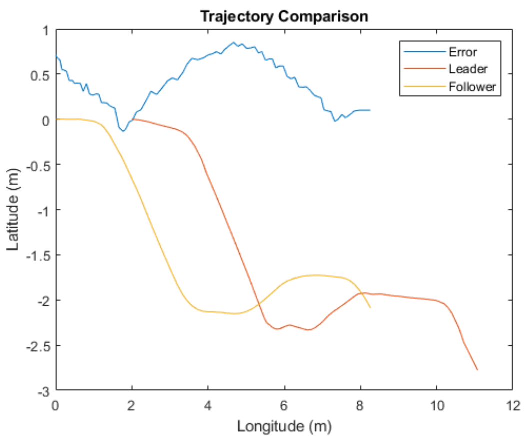

Fig. 9. *Platoon's Trajectory Comparison*

vehicles is defined as the Euclidian distance between the global position of the vehicles in the time t. The measured error is the difference between the desired distance and the measured distance. This graph demonstrates that the followers are fed with the leader's information within acceptable latency to perform the control action while keeping the safety distance thus avoiding collisions.

The leader also performs a "S" movement and the follower is able to repeat the movement with minor differences. Figure 10 presents the comparison between the Leader and the Follower's heading values. Here, it is possible to observe that the follower performed the orientation adjustments roughly in parallel with its leader vehicle due to the timed arrival of information. However, this will be changed in future implementations, so that corrections are made only when the right leader position at which the information originated is reached.

In order to compare the performance of the cooperative platooning with a non-cooperative version, we implemented non-cooperative platooning i.e. restricted to local sensing only, and had the platoon follow the same path. In this case, we relied upon the the stereo camera to detect a stop sign that is attached at the rear-end of the leader using Yolo V3 [18], compute its position in the image and then to input it to a PD controller. This local information was also aided by a

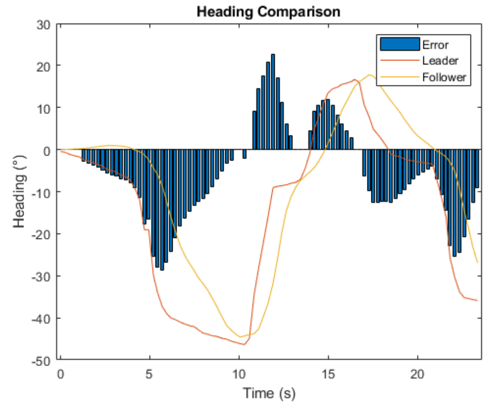

Fig. 10. *Comparison of the Leader and Follower's Heading Values over time.*

Fig. 11. Summary of Distance Errors

set of additional range finders and sonars to complement the camera vision strategy, when for some reason e.g. vibration, the vehicles were not able to detect the sign. The tests with non-cooperative platooning where carried out with the leader at a constant speed of $1.0m/s$.

The longitudinal and lateral errors in the tests with co-operative and non-cooperative platooning can be analysed in Figures 11 and 12 respectively. Both average errors are smaller when using communications instead of local sensor information. In longitudinal error, the improvement is in the order of 1.5% and in the lateral error is in the order of 20% in the same conditions - the leader running in $1.0m/s$. The improvement in longitudinal errors in cooperative scenarios is not significant given the constant speed of the leader. We

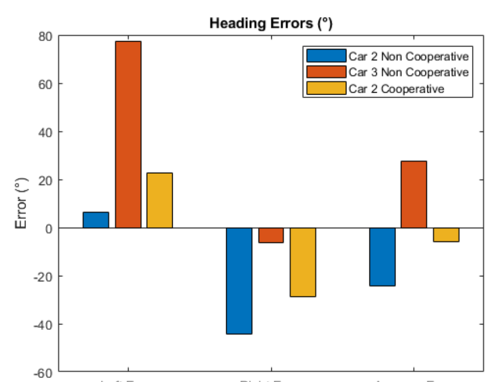

Fig. 12. Heading Errors Summary

believe that by varying the speed in the leader, we would obtain a more significant results in favor of the cooperative strategy. However, the improvement in the lateral control is much more perceptive. By receiving the leader's information via the OBUs, the delay between detecting a change in orientation and actuation upon the follower's lateral control is much shorter. In several platooning studies, lateral control is not taken into account, given the different techniques to keep the vehicle in a lane. However, often, lane markings are not visible, which can jeopardize the correct operation of system. Therefore, we believe it will be quite useful to explore the proposed algorithm, or similar, as this will increase the stability and the safety of the platooning.

V. Conclusions and Future Works

This work presented a flexible and scalable Robotic Testbed Platform called RoboCoPlat, directed at the development of a cooperative platooning solutions based on the ETSI ITS-G5 protocol, and encompassing real OBUs. In order to provide a stable communication between the ROS and the OBUs, a bridge software module was developed and implemented. The cooperative capability of the testbed was successfully demonstrated showing that it is possible to implement a successful cooperative platooning solution, by only relying on communications instead of local information. In parallel, we implemented a non-cooperative solution and demonstrated that cooperative platooning presents lower absolute errors, which leads to a stabler behavior.

We believe this testbed has the potential to support a series of important research activities in the near future, by enabling the analysis of the interdependence between communications and control, in particular by pushing the performance limits of the ITS-G5 stack. In addition, there is an ongoing project with and industrial partner, towards the integration and validation of a runtime monitoring architecture for increased safety in this kind of ITS scenarios. To accomplish this, other more complex scenarios such as higher speed slalom are being implemented.

References

[1] B. Chen, D. Robinette, M. Shahbakhti, K. Zhang, J. Naber, J. Worm, C. Pinnow, and C. Morgan, "Connected Vehicles and Powertrain Optimization," *Mechanical Engineering*, vol. 139, no. 09, Sep. 2017.

[2] L. Jin, M. Čičič, S. Amin, and K. H. Johansson, "Modeling the Impact of Vehicle Platooning on Highway Congestion: A Fluid Queuing Approach," in *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (part of CPS Week) - HSCC '18*. Porto, Portugal: ACM Press, 2018, pp. 237–246.

[3] A. Talebpour and H. S. Mahmassani, "Influence of connected and autonomous vehicles on traffic flow stability and throughput," *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 143–163, Oct. 2016.

[4] European Telecommunications Standards Institute, "ETSI TR 103 299 V2.1.1 Intelligent Transport Systems (ITS); Cooperative Adaptive Cruise Control (CACC); Pre-standardization study," European Telecommunications Standards Institute, Tech. Rep., Jun. 2019.

[5] O. Karoui, E. Guerfala, A. Koubaa, M. Khalgui, E. Tovard, N. Wu, A. Al-Ahmari, and Z. Li, "Performance evaluation of vehicular platoons using Webots," *IET Intelligent Transport Systems*, vol. 11, no. 8, pp. 441–449, Oct. 2017. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/iet-its.2017.0036

[6] T. Acarman, Yiting Liu, and U. Ozguner, "Intelligent cruise control stop and go with and without communication," in *2006 American Control Conference*, Jun. 2006, pp. 6 pp.–.

[7] Zhiwu Li, Oussama Karoui, Anis Koubaa, Mohamed Khalgui, Emna Guerfala, and Eduardo Tovar, "System and method for operating a follower vehicle in a vehicle platoon," Polytechnic Institute of Porto (ISEP-IPP), Portugal, Tech. Rep. CISTER-TR-181203, 2018.

[8] N. T. Tangirala, A. Abraham, A. Choudhury, P. Vyas, R. Zhang, and J. Dauwels, "Analysis of Packet drops and Channel Crowding in Vehicle Platooning using V2X communication," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, Nov. 2018, pp. 281–286.

[9] F. Rossi, "Safecop," Feb. 2020. [Online]. Available: http://www.safecop.eu/

[10] B. Vieira, R. Severino, E. V. Filho, A. Koubaa, and E. Tovar, "COPADRIVe - A Realistic Simulation Framework for Cooperative Autonomous Driving Applications," in *ICCVE 2019*, Graz, Austria, Nov. 2019, p. 6.

[11] O. Karoui, M. Khalgui, A. Koubâa, E. Guerfala, Z. Li, and E. Tovar, "Dual mode for vehicular platoon safety: Simulation and formal verification," *Information Sciences*, vol. 402, pp. 216–232, Sep. 2017.

[12] Z. Szendrei, N. Varga, and L. Bokor, "A SUMO-Based Hardware-in-the-Loop V2X Simulation Framework for Testing and Rapid Prototyping of Cooperative Vehicular Applications," in *Vehicle and Automotive Engineering 2*, K. Jármai and B. Bolló, Eds. Cham: Springer International Publishing, 2018, pp. 426–440.

[13] S. Wei, Y. Zou, X. Zhang, T. Zhang, and X. Li, "An Integrated Longitudinal and Lateral Vehicle Following Control System With Radar and Vehicle-to-Vehicle Communication," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1116–1127, Feb. 2019.

[14] European Telecommunications Standards Institute, "ETSI TR 102 638 V1.1.1 Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions," European Telecommunications Standards Institute, Tech. Rep. V1.1.1, 2009.

[15] B. Vedder, J. Vinter, and M. Jonsson, "A Low-Cost Model Vehicle Testbed with Accurate Positioning for Autonomous Driving," *Journal of Robotics*, vol. 2018, pp. 1–10, Nov. 2018.

[16] A. Belbachir, "An embedded testbed architecture to evaluate autonomous car driving," *Intelligent Service Robotics*, vol. 10, no. 2, Apr. 2017.

[17] J. P. Jansch-Porto and G. E. Dullerud, "Decentralized control with moving-horizon linear switched systems: Synthesis and testbed implementation," in *2017 American Control Conference (ACC)*, May 2017, pp. 851–856, iSSN: 2378-5861.

[18] A. Rupp, M. Tranninger, R. Wallner, J. Zubača, M. Steinberger, and M. Horn, "Fast and Low-Cost Testing of Advanced Driver Assistance Systems using Small-Scale Vehicles," *IFAC-PapersOnLine*, vol. 52, no. 5, pp. 34–39, 2019.

[19] D. Lu, Z. Li, D. Huang, X. Lu, Y. Deng, A. Chowdhary, and B. Li, "VC-bots: a vehicular cloud computing testbed with mobile robots," in *Proceedings of the First International Workshop on Internet of Vehicles and Vehicles of Internet - IoV-VoI '16*. Paderborn, Germany: ACM Press, 2016, pp. 31–36.

[20] D. Lu, Z. Li, and D. Huang, "Platooning as a service of autonomous vehicles," in *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Jun. 2017, pp. 1–6, iSSN: null.

[21] H. Cao, S. Gangakhedkar, A. R. Ali, M. Gharba, and J. Eichinger, "A Testbed for Experimenting 5G-V2X Requiring Reliability and Low-Latency," in *WSA 2017; 21th International ITG Workshop on Smart Antennas*, Mar. 2017, pp. 1–4, iSSN: null.

[22] M. O'Kelly, V. Sukhil, H. Abbas, J. Harkins, C. Kao, Y. V. Pant, R. Mangharam, D. Agarwal, M. Behl, P. Burgio, and M. Bertogna, "F1/10: An Open-Source Autonomous Cyber-Physical Platform," *arXiv:1901.08567 [cs]*, Jan. 2019, arXiv: 1901.08567.

[23] JetsonHacks, "NVIDIA Jetson TX2 J21 Header Pinout," Feb. 2020. [Online]. Available: https://www.jetsonhacks.com/nvidia-jetson-tx2-j21-header-pinout/

[24] Cohda Wireless, "MK5 OBU," Feb. 2020. [Online]. Available: https://cohdawireless.com/solutions/hardware/mk5-obu/

[25] StereoLabs, "ZED Stereo Camera | Stereolabs," Feb. 2020. [Online]. Available: https://www.stereolabs.com/zed/

[26] InvenSense Inc., "MPU 92-50, Product Specification," Jan. 2014. [Online]. Available: https://cdn.sparkfun.com/assets/learn_tutorials/5/5/0/MPU9250REV1.0.pdf