CISTER

Research Centre in
Real-Time & Embedded
Computing Systems

PhD Thesis

An Evaluation Framework for Safe Cooperative Vehicle Platooning

Enio Filho

CISTER-TR-230605

2023/05/29

An Evaluation Framework for Safe Cooperative Vehicle Platooning

Enio Filho

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

https://www.cister-labs.pt

Abstract

The development of cooperative vehicle systems is one of the industry's choices to mitigate traffic and transportation problems in urban centers and roads. However, the development of these systems is increasingly complex due to their multiple facets encompassing control, communications, and safety. The high level of integration between communication networks, sensors, actuators, and the dynamic characteristics of individual and group vehicles demands coordinated systems capable of responding in real-time to environment variations.

This Thesis addresses the building of a design framework for validation of the safety and performance aspects of these cooperative Cyber-Physical Systems (Co-CPS). We choose the study of cooperative vehicular platoons (Co-VP), since such applications are of great interest, due their potential to reduce energy consumption, improve traffic flow and increase transportation capacity. However, the literature shows a gap space in the integrated study of Co-VP control dynamics regarding communication issues. These systems are prone to safety failures when threatened by communication errors and delays. Moreover, the difficulty in consolidating a validation tool capable of jointly analyzing these aspects, showing the impacts of communication on control systems, was also observed.

The efficient validation of Co-VP systems demands a deep understanding of multiple topics. We begin by presenting a review of Co-VP research in terms of control and compile the most important characteristics of ETSI ITS-G5, the chosen communication infrastructure. Next, we conduct a multidimensional survey on the advances related to this subject, evaluating the control models and the impacts that network constraints cause on these vehicles. Finally, we present strategies to minimize security problems involving these applications.

Seeking to mitigate the lack of a Co-VP simulation tool that meets the needs of safety validation, this Thesis describes the construction of an integrated framework for developing, testing, and evaluating these systems. Encompassing microscopic aspects of communication and control, CopaDrive uses ROS as an integrative tool, extending the framework from simulation to implementation on a robotic testbed through a hybrid environment, in a Hardware in the Loop (HIL). Using the 3D robot simulator and a communication network simulator, we evaluate how the use of Cooperative Awareness Messages (CAMs), defined by ETSI ITS-G5, impacts the ability of the platoon to perform a u-turn. Furthermore, this Thesis presents how the same control and communication model was used in HIL, to validate a Control Loss Warning (CLW) module. Finally, we also show how the control model can be validated using a robotic testbed, using real On-board Units (OBUs).

The vehicles participating in a cooperative platoon are subject to quick conditions changes. Thus, considering only their longitudinal motion is not a reasonable option since the cars will inevitably have to make turns and face obstacles on a usual path. Therefore, we propose a Look-Ahead controller capable of integrating the lateral and longitudinal controls of the vehicles in the platoon in a distributed way. By propagating the trajectory of the lead vehicle through predecessor-follower communication, we reduce heading and distance errors, increasing the system's safety. We also propose using a lateral adjustment to correct the effect of cutting corners caused by the distance between the leader and the follower when a turn is performed.

The evolution of Co-VP systems is intrinsically dependent on communications, which are responsible for ensuring message delivery. Regarding the ITS-G5, this Thesis restricts the CAMs triggering mechanism by proposing a Platoon Service Profile (PSP). We show how this restriction contributes to increased platoon performance by reducing lateral and longitudinal errors in realistic scenarios. We also compare other message firing models established by ETSI ITS-G5 and conclude that using PSP does not significantly increase network throughput. We also show its applicability for urban and freeway scenarios.

Since the scenarios and technologies surrounding CPS are highly dynamic, we also show how CopaDrive can be used to validate cooperative vehicular applications for different environments. In this Thesis, for convenience and without loss of generality, we apply the proposed framework to a learning academic system using an inter-vehicular sensor network based on the IEEE 802.15.4 communication protocol.

Faculdade de Engenharia da Universidade do Porto

An Evaluation Framework for Safe Cooperative Vehicle Platooning

Enio Prates Vasconcelos Filho

U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Programa Doutoral em Engenharia Electrotécnica e de Computadores

Advisor: Prof. Dr. Eduardo Manuel Medicis Tovar

Co-Advisor: Prof. Dr. Anis Koubaa

Co-Advisor: Prof. Dr. Luis Miguel Pinho de Almeida

May 26, 2023

An Evaluation Framework for Safe Cooperative Vehicle Platooning

Enio Prates Vasconcelos Filho

Programa Doutoral em Engenharia Electrotécnica e de Computadores

Approved by:

President: Prof.Dr. José Nuno Moura Marques Fidalgo, Universidade do Porto
Referee: Prof. Dr. Rahul Mangharam, University of Pennsylvania, USA
Referee: Prof. Dr. António Casimiro Ferreira da Costa, Universidade de Lisboa
Referee: Prof. Dr. Joaquim José de Castro Ferreira, Universidade de Aveiro
Referee: Prof. Dr. Paulo José Lopes Machado Portugal, Universidade do Porto
Advisor: Prof. Dr. Eduardo Manuel Medicis Tovar
Co-Advisor: Prof. Dr. Anis Koubaa
Co-Advisor: Prof. Dr. Luis Miguel Pinho de Almeida

May 26, 2023

Abstract

The development of cooperative vehicle systems is one of the industry's choices to mitigate traffic and transportation problems in urban centers and roads. However, the development of these systems is increasingly complex due to their multiple facets encompassing control, communications, and safety. The high level of integration between communication networks, sensors, actuators, and the dynamic characteristics of individual and group vehicles demands coordinated systems capable of responding in real-time to environment variations.

This Thesis addresses the building of a design framework for validation of the safety and performance aspects of these cooperative Cyber-Physical Systems (Co-CPS). We choose the study of cooperative vehicular platoons (Co-VP), since such applications are of great interest, due their potential to reduce energy consumption, improve traffic flow and increase transportation capacity. However, the literature shows a gap space in the integrated study of Co-VP control dynamics regarding communication issues. These systems are prone to safety failures when threatened by communication errors and delays. Moreover, the difficulty in consolidating a validation tool capable of jointly analyzing these aspects, showing the impacts of communication on control systems, was also observed.

The efficient validation of Co-VP systems demands a deep understanding of multiple topics. We begin by presenting a review of Co-VP research in terms of control and compile the most important characteristics of ETSI ITS-G5, the chosen communication infrastructure. Next, we conduct a multidimensional survey on the advances related to this subject, evaluating the control models and the impacts that network constraints cause on these vehicles. Finally, we present strategies to minimize security problems involving these applications.

Seeking to mitigate the lack of a Co-VP simulation tool that meets the needs of safety validation, this Thesis describes the construction of an integrated framework for developing, testing, and evaluating these systems. Encompassing microscopic aspects of communication and control, CopaDrive uses ROS as an integrative tool, extending the framework from simulation to implementation on a robotic testbed through a hybrid environment, in a Hardware in the Loop (HIL). Using the 3D robot simulator and a communication network simulator, we evaluate how the use of Cooperative Awareness Messages (CAMs), defined by ETSI ITS-G5, impacts the ability of the platoon to perform a u-turn. Furthermore, this Thesis presents how the same control and communication model was used in HIL, to validate a Control Loss Warning (CLW) module. Finally, we also show how the control model can be validated using a robotic testbed, using real On-board Units (OBUs).

The vehicles participating in a cooperative platoon are subject to quick conditions changes. Thus, considering only their longitudinal motion is not a reasonable option since the cars will inevitably have to make turns and face obstacles on a usual path. Therefore, we propose a Look-Ahead controller capable of integrating the lateral and longitudinal controls of the vehicles in the platoon in a distributed way. By propagating the trajectory of the lead vehicle through predecessor-follower communication, we reduce heading and distance errors, increasing the system's safety.

We also propose using a lateral adjustment to correct the effect of cutting corners caused by the distance between the leader and the follower when a turn is performed.

The evolution of Co-VP systems is intrinsically dependent on communications, which are responsible for ensuring message delivery. Regarding the ITS-G5, this Thesis restricts the CAMs triggering mechanism by proposing a Platoon Service Profile (PSP). We show how this restriction contributes to increased platoon performance by reducing lateral and longitudinal errors in realistic scenarios. We also compare other message firing models established by ETSI ITS-G5 and conclude that using PSP does not significantly increase network throughput. We also show its applicability for urban and freeway scenarios.

Since the scenarios and technologies surrounding CPS are highly dynamic, we also show how CopaDrive can be used to validate cooperative vehicular applications for different environments. In this Thesis, for convenience and without loss of generality, we apply the proposed framework to a learning academic system using an inter-vehicular sensor network based on the IEEE 802.15.4 communication protocol.

Keywords: Cooperative Vehicular Platooning, Vehicular Networks, Simulation Framework, ETSI ITS-G5, Vehicular Control.

Resumo

O desenvolvimento de sistemas cooperativos veiculares é uma das apostas da indústria para mitigação de problemas de trânsito e transporte em centros urbanos e nas estradas. Entretanto, o desenvolvimento destes sistemas é cada vez mais complexo, em função de suas múltiplas facetas, que englobam controle, comunicações e segurança. Assim, o alto nível de integração entre redes de comunicação, sensores, atuadores e características dinâmicas individuais e do conjunto de veículos demanda sistemas coordenados e capazes de responder em tempo real as variações do ambiente.

Essa tese focou-se na construção de um ambiente que permitisse validar diferentes aspectos de segurança e desempenho destes sistemas cooperativos cyber-físicos (Co-CPS). Assim, optou-se por desenvolver o estudo dos pelotões veiculares cooperativos (Co-VP), por conta de seu potencial de redução de consumo de combustíveis, melhoria do tráfego e aumento da capacidade de transporte. Contudo, a literatura da área mostra um espaço não preenchido no estudo integrado das dinâmicas de controle Co-VP quando sujeito aos problemas causados pela comunicação. Assim, estes sistemas estão sujeitos a falhas de segurança quando ameaçados por perdas e atrasos de pacotes. Além disso, observou-se ainda a dificuldade em consolidar uma ferramenta de validação capaz de analisar estes aspectos de maneira conjunta, apontando os impactos da comunicação sobre os sistemas de controle.

A validação eficiente dos sistemas Co-VP demanda um profundo conhecimento das dinâmicas envolvidas e dos modelos utilizados. Assim, apresentamos uma revisão dos trabalhos que estudam as aplicações Co-VP em termos de controle e fazemos uma compilação das características mais importantes do ETSI ITS-G5, escolhido como modelo de comunicação a ser implementado. Em seguida, realizamos um survey multidimensional sobre os avanços relacionados a esse assunto, avaliando os modelos de controle e os impactos que os problemas de rede ocasionam sobre estes veículos. Como um aspecto mandatório, apresentamos ainda os estudos realizados para minimizar problemas de segurança que envolvem estas aplicações.

Buscando mitigar a falta de uma ferramenta de simulação Co-VP que atenda às necessidades de validação de segurança, essa tese descreve a construção de um framework integrado de desenvolvimento, teste e avaliação destes sistemas. Englobando aspectos microscópicos de comunicação e controle, o CopaDrive utiliza o ROS como ferramenta integradora capaz de estendê-la desde a simulação até a implementação em uma testbed robótica, passando por um ambiente híbrido, definido com Hardware in the Loop (HIL). Por meio de um simulador 3D e de um simulador de redes de comunicação, avaliamos como o uso das Cooperative Awareness Messages (CAMs), definidas pelo ETSI ITS-G5 impacta na capacidade do pelotão de realizar uma curva fechada. Essa tese apresenta ainda como o mesmo modelo de controle e de comunicação foi validado no HIL, utilizando ainda ferramentas externas de segurança, como o Control Loss Warning (CLW). Finalmente, mostramos ainda como o modelo de controle pode ser validado por meio de uma testbed robótica, utilizando On-board Units (OBUs).

Os veículos participantes de um pelotão cooperativo estão sujeitos a diversas variações de condições e mudanças de cenário. Assim, considerar apenas o movimento longitudinal do mesmo não é uma opção razoável, já que em um trajeto comum, os veículos fatalmente terão de fazer curvas e encarar obstáculos. Assim, propomos um controlador Look-Ahead, capaz de integrar os controles laterais e longitudinais dos veículos do pelotão de forma distribuída. Ao propagar a trajetória do veículo líder por meio da comunicação predecessor-seguidor, reduzimos os erros de direção e de distância, aumentando a segurança do sistema. Propomos ainda o uso de um ajuste lateral para corrigir o efeito de cortar os cantos das curvas, provocado pela distância entre o líder e o seguidor no momento da realização da curva.

A evolução dos sistemas Co-VP depende intrinsecamente das comunicações, responsáveis por garantir a entrega das mensagens. Nesta tese, restringimos os gatilhos que disparam as CAMs, propondo um Platoon Service Profile (PSP). Mostramos como essa restrição contribui para o aumento do desempenho do pelotão, reduzindo erros laterais e longitudinais em cenários realísticos. Comparamos ainda outros modelos de disparos de mensagens, estabelecidos pelo ETSI ITS-G5, e concluímos que o uso do PSP não aumenta significativamente o throughput na rede. Desse modo, mostramos sua aplicabilidade para cenários urbanos e de autoestradas.

Sendo os cenários e as tecnologias que envolvem os CPS altamente dinâmicas, mostramos ainda como o CopaDrive pode ser utilizado na validação de diferentes ambientes. Nesta tese, provemos os mecanismos para essa expansão, aplicando-o ao contexto acadêmico estudantil e o modificamos para a validação de uma rede de sensores intra-veicular baseada no protocolo IEEE 802.15.4.

Keywords: Pelotão Veicular Cooperativo, Redes de Comunicação Veicular, Framework de Simulação, ETSI ITS-G5, Controle Veicular

Acknowledgements

The achievement of a Ph.D. is seen as an individual goal. However, it is achieved after a long marathon where many people around him support the runner. People who also give up their activities to be with us and help accomplish the dreamed-of objective. So, I take this space to thank them.

Initially, I thank my wife, **Francieli**. Without her, none of this would be possible. My safe harbor, support, and strength in the most challenging times. Thank you for being with me on this journey, my love. And along with her, I thank my sons, **Gabriel** and **Heitor**, for being the source of inspiration and greater motivation to continue growing and being an example to them. I cannot fail to thank my parents **Albene** and **Ênio**, for supporting me early. If it weren't for them, I wouldn't have even started. Special thanks to my sister, **Layanna**, for all her help when I was most desperate.

I am very grateful to my advisors, who helped me construct and elaborate on the Thesis, for their guidance and teachings. I thank my Advisor **Eduardo Tovar** for the structure and for providing everything necessary for this Thesis to be possible. I thank my Co-Adviser, **Anis Koubaa**, for his incredible knowledge and quick thinking in showing the ways to follow. I thank my other Co-Advisor, **Luis Almeida**, for his advice and clear vision of how things work and how to apply them. I am also grateful for the advice, discussions, learning, and working together with **Pedro Santos**. Finally, a special thanks to **Ricardo Severino** for starting this work in the ICARUS project, working with me and pointing out directions to follow, and paths to take, and broadening my research horizons.

Inside CISTER, I met lots of people, learned like never before, and had the opportunity to work with excellent researchers and engineers. Among these, I especially thank **Bruno Vieira**, **Nuno Guedes**, **Daniel Almeida**, **Giann Nandi**, and **Sergio Penna** for the conversations, laughs, problem-solving, and random ideas that helped me during these years of research. I thank them also for the support in the work done and for all the help. And a big thank you to fellow Ph.D. students **Miguel**, **Jatin**, **Yilian**, **Javier**, **Youseff**, **Mubarak**, and **Harrison**!

Last but not least, I thank **God** for opening the doors he opened, for the stones and the opportunities I encountered along the way, which made me more than yesterday and less than tomorrow.

This work was partially supported by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology), within the CISTER Research Unit (UIDP/UIDB/04234/2020); by FCT and the EU ECSEL JU under the H2020 Framework Programme, within project ECSEL/0010/2019, JU grant nr. 876019 (ADACORSA). The JU receives support from the European Union's Horizon 2020 research and innovation programme and Germany, Netherlands, Austria, France, Sweden, Cyprus, Greece, Lithuania, Portugal, Italy, Finland, Turkey. The ECSEL JU and the European Commission are not responsible for the content on this Thesis or any use that may be made of the information it contains.

Enio Prates Vasconcelos Filho

Publications

Book Chapters

1. Vasconcelos Filho, Enio; Severino, Ricardo ; Rodrigues, João ; Goncalves, Bruno ; Koubaa, Anis ; Tovar, Eduardo. CopaDrive: An Integrated ROS Cooperative Driving Test and Validation Framework. Robot Operating System (ROS). VI ed.: , 2021, v. 962, p. 121-174.

2. Vasconcelos Filho, E., Yudi, J., Abdelkader, M., Koubaa, A., Tovar, E. (2023). RosDrive: An Open-Source ROS-Based Vehicular Simulator for STEM Control Systems Classes Tutorial. In: Koubaa, A. (eds) Robot Operating System (ROS). Studies in Computational Intelligence, vol 1051. Springer, Cham. https://doi.org/10.1007/978-3-031-09062-2_5

Journals

1. Vasconcelos Filho, Enio; Santos, Pedro M. ; Severino, Ricardo ; Koubaa, Anis ; Tovar, Eduardo. Improving the Performance of Cooperative Platooning With Restricted Message Trigger Thresholds. IEEE Access, v. 10, p. 45562-45575, 2022.

2. Enio Vasconcelos Filho, Ricardo Severino, Pedro M. Salgueiro dos Santos, Anis Koubaa, Eduardo Tovar (2023): Cooperative vehicular platooning: a multi-dimensional survey towards enhanced safety, security and validation, Cyber-Physical Systems, DOI: 10.1080/23335777.2023.2214584

Conferences and Workshops

1. Vasconcelos Filho, Enio; Severino, Ricardo ; Koubaa, Anis ; Tovar, Eduardo. An Integrated Lateral and Longitudinal Look Ahead Controller for Cooperative Vehicular Platooning. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. 364ed.: Springer International Publishing, 2021, v. , p. 142-159.

2. Kurunathan, Harrison ; Severino, Ricardo ; Filho, Ênio ; Tovar, Eduardo. WiCAR - Simulating Towards the Wireless Car. Lecture Notes in Computer Science. 1ed.: Springer International Publishing, 2020, p. 136-147.

3. Filho, Enio Vasconcelos; Guedes, Nuno ; Vieira, Bruno ; Mestre, Miguel ; Severino, Ricardo ; Goncalves, Bruno ; Koubaa, Anis ; Tovar, Eduardo. Towards a Cooperative Robotic Platooning Testbed. In: 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), 2020, Ponta Delgada, p. 332.

4. Vieira, Bruno ; Severino, Ricardo ; Filho, Enio Vasconcelos ; Koubaa, Anis ; Tovar, Eduardo. COPADRIVe - A Realistic Simulation Framework for Cooperative Autonomous Driving Applications. In: 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE), 2019, Graz. p. 1-10.

Poster and Extended Abstracts

1. Vasconcelos Filho, Enio; Severino, Ricardo ; Koubaa, Anis ; Tovar, Eduardo. A Wireless Safety and Security Layer Architecture for Reliable Co-CPS. In: 4th Doctoral Congress In Engineering (DCE21), 2021, Porto. DCE21- Symposium on Electrical and Computer Engineering: Book of Abstracts. Porto: FEUP Edições, 2021. v. 1. p. 27-29.
2. Vasconcelos Filho, Enio; Severino, Ricardo ; Koubaa, Anis ; Tovar, Eduardo. A Real-Time QoS Monitor Architecture Proposal for Cooperative Vehicular Platooning. In: 4rd Doctoral Congress in Engineering (DCE21), 2021, Porto. Book of Abstracts of the Symposium on Transport Systems and Mobility. Porto: FEUP Edições, 2021. v. 1. p. 17-20.

Related Works

1. Sanhudo, Luís ; Rodrigues, João ; Vasconcelos Filho, Enio. Multivariate time series clustering and forecasting for building energy analysis: Application to weather data quality control. Journal Of Building Engineering, v. 1, p. 101996, 2020.
2. Vasconcelos Filho, Enio; Lopes Dos Santos, Paulo. A Dynamic Mode Decomposition Approach With Hankel Blocks to Forecast Multi-Channel Temporal Series. IEEE Control Systems Letters, v. 3, p. 739-744, 2019.
3. Vasconcelos Filho, Enio; Gomes, Filipe; Monteiro, Stephane; Severino, Ricardo ; Koubaa, Anis ; Tovar, Eduardo. A Drone Secure Handover Architecture validated in a Software in the Loop Environment," in 12th EASN International Conference, vol. 12, (Barcelona, Spain), p. 8, EASN, Oct. 2022.
4. Vasconcelos Filho, E., Mendes, B., Santos, P.M., Severino, R., Tovar, E. (2023). Development of a Hardware in the Loop Ad-Hoc Testbed for Cooperative Vehicles Platooning. In: Martins, A.L., Ferreira, J.C., Kocian, A., Tokkozhina, U. (eds) Intelligent Transport Systems. INTSYS 2022. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 486. Springer, Cham. https://doi.org/10.1007/978-3-031-30855-0_4
5. Marcia Cunha Rocha, Enio Vasconcelos Filho, Fernando Alves, Sergio Penna, Pedro M. Santos, and Eduardo Tovar. 2023. A WSSL Implementation for Critical Cyber-Physical Systems Applications. In Cyber-Physical Systems and Internet of Things Week 2023 (CPS-IoT Week Workshops '23), May 09–12, 2023, San Antonio, TX, USA. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3576914.3587507
6. Pereira J, Marta D, Vasconcelos Filho E, et al. Edge-Aided V2X Collision Avoidance with Platoons: Towards a Hybrid Evaluation Toolset. In: 9th International Workshop on Safety and Security of Intelligent Vehicles; Jun.; Porto, Portugal; 2023. p. 4.

"One day your life will flash before your eyes.
Make sure it's worth watching."

The Bucket List

Conteúdo

1 Introduction **1**
- 1.1 Motivation and Challenges . . . 3
- 1.2 Approach . . . 4
- 1.3 Thesis Statement . . . 6
- 1.4 Contributions . . . 7
- 1.5 Thesis Outline . . . 8

2 Cooperative Vehicular Platooning Background **11**
- 2.1 Overview . . . 11
- 2.2 Related Surveys . . . 13
  - 2.2.1 Co-VP Architecture Surveys . . . 14
  - 2.2.2 Co-VP Controller and Efficiency Surveys . . . 15
  - 2.2.3 Co-VP Test and Validation Surveys . . . 15
- 2.3 Co-VP Formal Model . . . 16
- 2.4 Co-VP Stability Analysis . . . 17
- 2.5 Co-VP Inter-Space Policy . . . 19
  - 2.5.1 Co-VP Information Flow Topology . . . 20
  - 2.5.2 Co-VP Controller Strategies . . . 20
- 2.6 Conclusion . . . 24

3 V2V Communications **25**
- 3.1 Overview . . . 25
- 3.2 V2X Literature Review . . . 26
- 3.3 The ITS-G5 Standard . . . 27
  - 3.3.1 Applications . . . 29
  - 3.3.2 Facilities . . . 29
  - 3.3.3 Management . . . 31
  - 3.3.4 Security . . . 31
  - 3.3.5 Network and Transport layer . . . 32
  - 3.3.6 Access layer . . . 33
  - 3.3.7 Co-VP Settings . . . 35
- 3.4 Dedicated Short-Range Communication (DSRC) - WAVE . . . 38
- 3.5 LTE: C-V2X . . . 39
- 3.6 Conclusion . . . 41

4 Co-VP Overview **43**
- 4.1 Introduction . . . 43
- 4.2 Co-VP Reliability . . . 44

4.2.1 Network Threats 44
4.2.2 Transmission Parameters Adjust 46
4.3 Security Analysis of Co-VP applications 47
4.3.1 Vehicular Network Vulnerabilities 48
4.3.2 Co-VP Stability under Security Attacks 48
4.4 Validation Tools 50
4.4.1 Simulation 51
4.4.2 Experimentation Tools 56
4.5 Open Challenges 63
4.6 Conclusions 64

5 CopaDrive: an integrated ROS cooperative driving test and validation framework 67
5.1 Introduction 67
5.2 Co-VP Safety Tools 69
5.3 CopaDrive - Integrated System 72
5.4 CopaDrive Simulator (CD-S) 74
5.4.1 CD-S Central Components 74
5.4.2 Framework Architecture 75
5.4.3 Experimental Results 77
5.4.4 Traffic Analysis 84
5.5 CopaDrive Hardware-in-the-Loop - CD-HIL 85
5.5.1 ROS-MK5 Bridge 89
5.5.2 Experimental Results 90
5.6 CD-RoboCoPlat 91
5.6.1 Testbed's Architecture 92
5.6.2 Localization 95
5.6.3 System Validation 96
5.7 Conclusions 97

6 An Integrated Look Ahead Controller for Co-VP 99
6.1 Introduction 99
6.2 Problem Formulation 101
6.2.1 Control Model 101
6.2.2 Metrics of Platoon Safety 102
6.3 Controller Model 103
6.3.1 Longitudinal and Lateral Controllers 103
6.3.2 Look Ahead Controller - LAC 104
6.4 Simulation Environment 105
6.5 Simulation Scenarios 106
6.5.1 Scenario 01 - Bearing test 107
6.5.2 Scenario 02 - LAC 108
6.5.3 Scenario 03 - Complex Circuit 109
6.5.4 Scenario 04 - Obstacle Avoidance 110
6.6 Conclusions 111

7 Improving the Performance of Cooperative Platooning With Restricted Message Trigger Thresholds 113
7.1 Introduction . . . 113
7.2 Related Works . . . 114
7.3 Exploring Trigger Thresholds for CAM Messages . . . 116
7.3.1 Current ETSI ITS Triggers . . . 116
7.3.2 Tentative Service Profiles . . . 117
7.4 Evaluation of the Service Profiles . . . 118
7.4.1 Evaluated Scenarios . . . 119
7.4.2 Metrics . . . 120
7.4.3 Straight Line Scenario - SC1 . . . 122
7.4.4 Multi-Curve Scenario - SC2 . . . 124
7.4.5 Obstacle Scenario - SC3 . . . 126
7.4.6 Network Performance - SC4 . . . 128
7.5 Conclusions . . . 129

8 CopaDrive Extensions 133
8.1 Introduction . . . 133
8.2 RosDrive: An Open-source ROS-Based Vehicular Simulator for STEM Control Systems Classes Tutorial . . . 134
8.2.1 Simulator Architecture . . . 136
8.2.2 Control Algorithms . . . 140
8.2.3 Vehicle Model . . . 140
8.2.4 Experimental Validation . . . 146
8.3 WiCAR - Simulating towards the Wireless Car . . . 154
8.3.1 Related Work . . . 155
8.3.2 Co-simulation Framework . . . 156
8.3.3 Network Specification . . . 157
8.3.4 Performance analysis . . . 158
8.4 Conclusion . . . 161

9 Conclusions and Future Work 163
9.1 Main Conclusions . . . 163
9.2 Future Research Directions . . . 165

A CopaDrive Install Instructions 167
A.1 Repository . . . 167
A.1.1 Main Requirements: . . . 167
A.1.2 Setup Project . . . 167
A.1.3 How to run CopaDrive Simulator . . . 168

Referências 169

Lista de Figuras

2.1 Driving Patterns for Platooning 12
2.2 Driving Applications 13
2.3 Basic Car-Following (B-CF) schematic 16
2.4 Section 2.3 general diagram 16
2.5 Co-VP Information Flow Topology (IFT) 20

3.1 Protocol Stack for ITS-G5 28
3.2 Vehicle ITS station in a vehicle sub-system [73] 29
3.3 CAM Message Format 31
3.4 Security Services [74] 32
3.5 ETSI ITS-G5 Channel allocation [76] 34
3.6 DCC state machines and CBR measurement procedure (m is the number of channel probes within the measurement period T): a) ETSI TS 102 687 configuration; b) ETSI TS 103 175 configuration; c) ETSI TR 101 612 configuration, ; d) Channel Busy Ratio (CBR) measurement [194] 35
3.7 Co-VP time gap (Δt) 36
3.8 Co-VP functional architecture 37
3.9 TR 103 299 CAM generation Rule 37
3.10 DSRC protocol Stack 38
3.11 DSRC Spectrum Division [154] 38
3.12 LTE-V2X Architecture [69] 39
3.13 LTE-V2X usual Application 40

4.1 Experimentation Tools Architecture 52

5.1 CopaDrive Main Architecture View 68
5.2 Control Loss Warning 70
5.3 OBU Logical Architecture 71
5.4 CopaDrive toolset and validation stages 73
5.5 CD-S Simulation Environment 75
5.6 Framework Architecture 77
5.7 CAM Sender Fluxogram 78
5.8 CAM Receiver Fluxogram 79
5.9 Data workflow 80
5.10 Platooning Trajectory 81
5.11 Vehicle inter-distances - Scenario A 82
5.12 Distance Error - Scenario A 82
5.13 Steering Angles - Scenario A 82
5.14 CAM Exchanging Delay - Scenario A 83

5.15 Longitudinal distances analysis in scenarios B, C and D . . . 83
5.16 Steering Analysis for scenarios B, C and D . . . 83
5.17 Period CAM-BSP . . . 84
5.18 Period CAM-BSP Platoon . . . 84
5.19 Period CAM-CSP . . . 84
5.20 Throughput Analysis for different scenarios . . . 85
5.22 CD-S Traffic Simulation Scenario . . . 86
5.23 Messages Delivery Time in Vehicular Traffic . . . 87
5.24 Network Throughput in Vehicular Traffic . . . 87
5.25 CD-HIL Architecture . . . 88
5.26 CD-HIL Deployment . . . 88
5.27 Bridge Architecture . . . 89
5.28 FDS scenario with and without CLW . . . 91
5.29 CD-RoboCoPlat . . . 93
5.30 Hardware Architecture . . . 93
5.31 Software Architecture . . . 94
5.32 Odometry Error . . . 95
5.33 *Platoon's path* . . . 96

6.1 Platooning View . . . 100
6.2 Co-VP Model with PF-IFT . . . 101
6.3 Platoon Model . . . 101
6.4 Bearing Error . . . 101
6.5 Longitudinal Controller Model . . . 104
6.6 The LAC consider the difference between the current position of SV_{i-1} and the desired position . . . 105
6.7 City Circuit . . . 106
6.8 Simulation Architecture . . . 106
6.9 Scenario 01 - Bearing Test . . . 107
6.10 SVs Position with/without bearing Controller . . . 107
6.11 Scenario 01 - Longitudinal alignment . . . 108
6.12 Scenario 01 - Heading Alignment . . . 108
6.13 Scenario 02 - Look Ahead Controller (LAC) . . . 108
6.14 Scenario 02 - Platooning curve performance . . . 108
6.15 Scenario 03 - Platooning curve performance . . . 109
6.16 Scenario 03 - Vehicles Path . . . 109
6.17 Scenario 03 - Vehicles inter Distances . . . 109
6.18 Scenario 04 - Vehicles Path . . . 110
6.19 Scenario 04 - Vehicles Inter Distances . . . 110
6.20 Scenario 04 - Obstacles Distances . . . 111

7.1 ETSI ITS-G5 Stack . . . 116
7.2 Time requirements for CAM generation and CAM processing . . . 117
7.3 Track for Simulations . . . 119
7.4 SC1 - SV_5 Speed Error Comparison . . . 121
7.5 SC1 - SV_5 Distance Error Comparison (e_5^{δ}) . . . 121
7.6 SC1 - Co-VP Stability Check . . . 121
7.7 SC1 - Total Triggers per Profile . . . 121
7.8 SC1 - Message Profile Delay Analysis . . . 121

7.9 Scenarios Throughput . . . 121
7.10 SC2 - SV_5 Distance Error (e_5^δ) . . . 123
7.11 SC2 - Co-VP Stability Check . . . 123
7.12 SC2 - SV_5 Heading Error (e_5^θ) . . . 123
7.13 SC2 - SV_5 Trajectory Comparison . . . 123
7.14 SC2 - Message Profile Delay Analysis . . . 123
7.15 SC2 - Total Triggers per Profile . . . 123
7.16 SC3 - SV_5 Distance Error (e_5^δ) . . . 124
7.17 SC3 - Co-VP Stability Check . . . 124
7.18 SC3 - SV_5 Trajectory Comparison . . . 124
7.19 SC3 - SV_5 Heading Error (e_5^θ) . . . 124
7.20 SC3 - Message Profile Delay Analysis . . . 125
7.21 SC3 - Total Triggers per Profile . . . 125
7.22 SC1 - Sent Messages per Vehicle . . . 126
7.23 SC2 - Sent Messages per Vehicle . . . 126
7.24 SC3 - Sent Messages per Vehicle . . . 126
7.25 SC4 - Packet Deliver Ratio . . . 126

8.1 General Simulator Architecture . . . 135
8.2 Publish/Subscribe Model . . . 136
8.3 Track Model . . . 138
8.4 Prius Gazebo Model [280] . . . 139
8.5 Prius Information Structure . . . 139
8.6 Vehicle 2D Model and Coordinates . . . 141
8.7 PID CC . . . 142
8.8 Line detection process . . . 143
8.9 Vehicle Heading Error (ε^θ) . . . 144
8.10 Line detection and Driving controller . . . 144
8.11 Sonar Visualization . . . 145
8.12 General Vehicle Architecture, with Line Detection and Detection and Avoidance modules . . . 146
8.13 Vehicle Speed Response to different PID parameters . . . 147
8.14 Vehicle Trajectory Analysis (Curve 7) under different HC PID Settings . . . 149
8.15 Vehicle Heading Error(ε^θ) at Curve 7 under different HC PID Settings . . . 149
8.16 Vehicle Trajectory Analysis (Full Lap) under different HC PID Settings . . . 150
8.17 Vehicle Trajectory Analysis (Static Obstacle Avoidance) under different HC PID Settings . . . 151
8.18 Vehicle Trajectory Analysis (Dynamic Obstacle Avoidance) . . . 152
8.19 Integration Architecture . . . 157
8.20 DSME superframe . . . 158
8.21 Scenario taken for evaluation . . . 158
8.22 Maximum acceptable delay for the braking force applied . . . 159
8.23 Impact of static scheduling and braking force on the crash rate . . . 160
8.24 Delay with Static scheduling for MO=4 and SO=4 . . . 161

Lista de Tabelas

2.1 Related Surveys Comparison 14
2.2 Summary of Control Models for Co-VP 23

3.1 V2V communication standards 26
3.2 Types of Messages 30

4.1 Summary of Co-VP Network Analysis 45
4.2 Security Requirements and Attacks 48
4.3 Co-VP Cybersecurity Research 49
4.4 Summary of Simulation Frameworks for Co-VP 53
4.5 Summary of CACC HIL Platforms 58
4.6 Summary of Platooning Robotic Testbeds 61

5.1 Trigger to CAM messages 78
5.2 Comparison between Scenarios - Number of messages and Safety Guarantee . . . 83
5.3 HIL simulations - Analysis of CLW alert systems 92
5.4 CD-RoboCoplat Components 93

6.1 Definition Terms 102
6.2 Model Parameters 107

7.1 Service Profiles 118
7.2 Model Parameters 120
7.3 Triggers Profile per Vehicle 131

8.1 Cruise Controller PID Settings 147
8.2 HC PID parameters 148

Abbreviations

ACC	Adaptative Cruise Control	InSecTT	Intelligent Secure Trustable Things
ADACORSA	Airborne Data Collection on Resilient System Architectures	ITS	Intelligent Transport Systems
AV	Autonomous Vehicles	LAC	Look Ahead Controller
B-CF	Basic Car-Following	LDM	Local Dynamic Map
BSA	Basic Set of Applications	LTE	Long Term Evolution
BSP	Basic Service Profile	MPC	Model Predictive Controller
BSP-P	Basic Service Profile for Platooning	OBU	On-Board Unit
C2C-CC	Car-2-Car Communication Consortium	PDU	Protocol Data Unity
CACC	Cooperative adaptive cruise control	PF	Predecessor Follower
CAM	Cooperative Awareness Message	PID	Proportional Integral Derivative
CAV	Cooperative Autonomous Vehicles	PSP	Platoon Service Profile
CF	Car Following	RHW	Cooperative Road Hazard Warning
CF	Car-Following	RM	Robot Middleware
CLW	Control Loss Warning	ROS	Robot Operating System
Co-CPS	Cooperative Cyber-Physical Systems	RSU	Road Side Unit
Co-VP	Cooperative Vehicular platooning	SafeCop	Safe Cooperating Cyber-Physical Systems using Wireless
CPS	Cyber-Physical Systems	SB	Symmetric Bidirectional Communication
CSP	Constant Spacing Policy	SV	Subject Vehicle
CTHP	Constant Time-Headway Policy	TV	Target Vehicle
DENM	Decentralized Environmental Notification Messages	V2I	Vehicle to Infrastructure
DSRC	Dedicated Short Range Communication	V2V	Vehicle to Vehicle
ETSI	European Telecommunications Standards Institute	V2X	Vehicle to Everything
FLOYD	5G/SDN Intelligent Systems For LOw latencY V2X communications	VANET	Vehicular ad hoc networks
GL	Global Leader	VDP	Vehicle Data Provider
HIL	Hardware-in-the-Loop	VP	Vehicular Platooning
IFT	Information Flow Topology	WAVE	Wireless Access in Vehicle Environments

Capítulo 1

Introduction

Cyber-Physical Systems (CPS) have taken on the most diverse roles and applications in increasingly complex scenarios. From the industrial field to the simplest everyday utensils, their ability to integrate themselves increases their importance. It allows them to be used in more significant numbers and with greater security. On the other hand, the advance in communication networks has allowed the development of a new class of CPS capable of working together, called Cooperative CPS (Co-CPS). Co-CPSs presents a new paradigm, increasing their capacity to act, allowing the performance of critical activities with the need for real-time responses for joint action [224]. Along with the new capabilities, new challenges have arisen inherent to the safety and security of their use in different contexts.

Such systems, characterized by unprecedented levels of ubiquity, give rise to research in several areas, including Drones and Industry 4.0. Among these, the study and development of autonomous vehicles (AVs) has become a reality in recent years, bringing an impact on society that is both profound in the way it changes mobility and transportation systems as well as far-reaching in the technological evolution associated with its development process [195]. Large companies invest heavily in their autonomous platforms, seeking to enter and take hold of an emerging market of great potential [310]. According to [275, 105], the global market size of autonomous driving is $24.1B, with a growth expectation up to $173.15B by 2030, with Shared Mobility Services Contributing to 65.31%. However, many issues concerning cost and legal barriers need to be addressed to achieve a more significant reach [270].

The extension of autonomous cars into an integrated transportation and mobility vision defines the Intelligent Transportation System (ITS) concepts. In this scenario, connectivity between vehicles (V2V) and between vehicles and infrastructure (V2I) provides the dissemination of information between agents, enabling a variety of advanced applications. Thus, Cooperative Vehicular Platooning (Co-VP) [128] emerges as a key application that will advance the safety and efficiency of autonomous driving. Road capacity and energy efficiency can be increased by having groups of vehicles traveling close together and constantly exchanging information through vehicle-to-everything (V2X) links. At the same time, accident occurrence is reduced [18, 256]. The development of connected platoons is present in the plans of automakers, developers and governments worldwide [282, 3, 115] with an agenda for implementation until 2030 [118].

The European Telecommunications Standards Institute (ETSI) ITS-G5 [72] and the IEEE 1609 Family of Standards for Wireless Access in Vehicular Environments (WAVE) [67, 132] have become the leading standards defining V2X communication. These operate on top of existing PHY/-MAC technologies, such as the IEEE 802.11p-based Direct Short Range Communications (DSRC) or the 3GPP's C-V2X (that encompass LTE-V2X and NR-V2X) [203, 2]. The studies and applications of these communication technologies have expanded the possibilities of Co-VP applications, defining rules, scenarios, and use cases that validate the applications and determine conditions of use. The European Union has advocated using the ETSI ITS-G5 standards in its vehicle communication studies, while the US has opted for the WAVE solution. ETSI also defines a series of use cases and safety-critical conditions for implementing solutions in various scenarios, including Co-VP applications.

However, Co-VP present several safety challenges, considering that they heavily rely on wireless communications to exchange safety-critical information. For example, frequently in Co-VP, wireless exchanged messages contribute to maintaining the inter-vehicle safety distance or relay safety alarms to the following vehicles. Hence, just as V2X communication can improve platooning safety [149], via the introduction of an additional information source beyond the limit of the vehicle's sensors, its usage also raises concerns regarding the reliability and security of communications and its impact on traffic safety, and efficiency [5, 336]. Furthermore, several adverse effects can be observed in Co-CPS controllers in the presence of communication issues, such as data packet loss and transmission delay [273].

To address many of these problems in safety-critical Co-CPS, the SafeCOP project [66] heavily studied these topics towards the definition of mechanisms and safety assurance procedures that could guarantee safety in those systems-of-systems. One example of such systems with direct applicability to Co-VP is the Control Loss Warning (CLW), as proposed in one of the use cases and described in [227, 202]. This system aims to trigger a specific safety action according to the scenario if one or more of the vehicles involved in the platoon fails to maintain the required speed or distance (longitudinal or lateral) to the preceding vehicle. It was achieved by analyzing the received data from the preceding vehicle and comparing it to the current and predicted behavior of the car.

To understand the safety limits of such proposals in an understandable fashion, extensive testing and validation must be carried out. Nonetheless, the complexity, cost, and safety risks involved in testing with actual device deployments, progressively demand realistic simulation tools to ease the validation of such technologies, helping to bridge the gap between development and real-world deployment. Notably, as accurately as possible, such comprehensive simulation tools must be able to mimic real-life scenarios from the autonomous driving or control perspective and the communications perspective, as both are highly interdependent.

Thus, the demand for a validation system for Co-CPS systems involves evaluating the entire implementation chain of its components in a reliable and integrated manner of its components. Unfortunately, there is no tool capable of analyzing realistic scenarios involving aspects of control and communication networks, examining the impacts that different network problems may cause in control systems, and vice-versa. Such an integrated validation process would also allow for the analysis of platoon performance aspects related to safety and security. This is the strategy we follow in this Thesis.

1.1 Motivation and Challenges

The implementation of Co-VP systems presents challenge since a significant set of adopted models can be as centralized or distributed, equipped or not with delays and with a series of critical safety constraints to reduce the possibility of accidents. For instance, in a connected platoon, vehicles can travel at higher speeds with inferior inter-vehicle distance, thus reducing fuel consumption by taking advantage of the slipstream while retaining all safety guarantees [18, 256, 145], such as improving longitudinal safety [168] and increasing the road capacity [27]. Co-VPs also reduce the risk for soldiers in military theaters by reducing the need for drivers on military convoys [104] and increase passenger capacity in public transport [126].

The modeling of these Co-VP controllers demands the analysis of the follower vehicles' ability to react promptly to the leader car's action, dealing with the inherent communication delays and the constant scenario variations. On the other hand, the network models must be adequate to respond to road conditions and message triggers in real time. This challenging application encompasses different topics, such as cooperative control models [112], V2V and V2I communication [94], energy efficiency [190], safety, interaction with other vehicles and platoons, among others. The V2V and V2I communication have an important role in increasing the performance of the resulting platoons compared to those obtained without communication between the vehicles [341, 141].

Already widely used to design robotics applications, the Robot Operating System (ROS) framework has been steadily addressing autonomous vehicles, easing the development process by providing multiple libraries, tools, and algorithms, and supporting several applications capable of simulating the physics and several of the sensor/actuator and control components of these vehicles. On the other hand, several network simulators are available and capable of carrying out network simulations of vehicular networks. Nonetheless, these tools remain mostly separated from the autonomous driving reality, offering few or minimal capabilities to evaluate complex cooperative autonomous driving systems.

Although using simulation software is the most flexible and economical approach for analyzing these issues, with the ability to scale the system as desired, the fact that such simulators do not encompass the processing characteristics or constraints of real computing and communication platforms reduces their effectiveness. Therefore, several efforts have been carried out to integrate simulations with Hardware in the Loop (HIL) [269, 307], which implements safety mechanisms.

This approach enables the deployment, test, and validation of such safety mechanisms in many scenarios, exploring the performance limits while guaranteeing safety.

Although the value of such methodology is indisputable since it can provide significant additional support to the systems development, there are still limitations, as several vehicle components are not included. In addition, such tests are expensive and challenging to scale. Robotic testbeds appear as a good solution in the middle ground between such simulation-based approaches and full vehicle deployments. Considering their flexibility, they can integrate with different platforms to be deployed in vehicles. They also can be deployed indoors in controlled environments and partially replicate a realistic scenario at a fraction of the cost of an actual vehicle [97]. Therefore, there is no single solution to support the development, testing, and validation of Co-VP systems, as each presents its clear advantages and limitations. The best approach is thus to rely upon the usage of several test and validation tools for each stage of the development process.

On the other hand, the lack of integration between different platforms significantly increases the development time of the Co-CPS system. The effort involved in integrating the system components with the validation platform is repeatedly discarded due to the significant differences between test environments and the prototype system. We believe that a fully integrated framework supported by ROS can serve the purpose of providing standard middleware from the beginning of the development process to the final deployment of the prototype system.

1.2 Approach

We studied four aspects to tackle the challenges identified in the previous section. First, we studied the impact of communications on Co-VP control systems to increase the security of these applications and ensure their applicability in existing systems. The various problems regarding control systems in connected platoons were studied, including different models, control limitations, and real-time systems. Third, we also examined the most used communication networks, their differences, and their applications. Finally, we surveyed the existing validation means, comparing them regarding strengths and weaknesses.

A review of validation tools for Co-CPS systems shows a wide variety. Several tools show capability and potential for modeling and performing tests that allow understanding the operation of these systems. In this context, ROS emerges as a potential solution enabler because of its flexibility and integration capability. However, the limitation of integration between robotic and network simulators hindered the mimicking of realistic models that would allow the validation of control strategies and the use of the communication model. Thus, we developed a 3D vehicle simulation tool integrated with an ETSI ITS-G5 network simulator, which allowed the evaluation of network problems, such as delays and packet losses on the safety and reliability of the Co-VP system. This tool was named CopaDrive, with ROS as the engine, Gazebo as the 3D interface, and OMNET++ as the network simulation tool. Given the critical nature of these systems and the conditions to be evaluated, the focus was on using a simulator capable of validating microscopic aspects of the interactions between agents, such as cornering, dragging, vehicle mechanics, and obstacle avoidance, among others.

The construction of CopaDrive also involved the definition of an architecture that would minimize the effort of implementation in a real platform, reducing the distance between the virtual and real worlds. Opting for ROS allowed the evolution of CopaDrive into a framework that validates cooperative platooning systems. This framework allowed the migration of the models used in simulation to a HIL implementation, integrating software and On-Board Units (OBUs) for vehicle communications validation. Finally, we used 1:10 scale vehicles, with ROS as an enabler and OBUs to perform communication to validate these algorithms in a robotic testbed. It was possible to evaluate the study of the Co-VP system from various perspectives, including software and hardware, going through the ETSI ITS-G5 communication systems.

CopaDrive allows the development of control strategies for platooning, optimizing its operation, and increasing its safety. Platoons require the guarantee that the followers can pursue the same trajectory as the leader, keeping a safe distance. Such control must be stable and robust and thus ensure the safety of the agents involved. Given the various scenarios where platoons can be used, one must consider the presence of curves and obstacles common to any road to be traveled. Although the control literature for Co-VP applications is comprehensive, only a few works emphasize vehicles' lateral and longitudinal control in an integrated manner. Here, we devise a control model enabling platoons to follow their leaders on realistic circuits, representing real scenarios. To this end, we seek to improve the performance of the Co-VP application in the trajectory following and curve execution while increasing the number of vehicles in the platoon through a look-ahead controller.

The ETSI ITS-G5 standard defines the transmission of Cooperative Awareness Messages (CAM) (similarly, WAVE defines Basic Safety Messages - BSM) to enable cooperative perception, augmenting each vehicle's situational awareness and knowledge horizon. CAM messages can be transmitted periodically, at a pre-defined time interval, or can be event-triggered when a kinematic threshold is crossed, e.g., when speed or heading angle crosses a given value. However, few studies analyze how the frequency of sending these messages impacts the platoon's behavior in different scenarios. Therefore, we used CopaDrive to evaluate message trigger profiles within ITS-G5 and their impact on the ability of vehicles to follow the leader in different conditions. This study considered the effect that new profiles on the communication network compared to the existing profiles, and the minimization of longitudinal and lateral errors in the movements of the platooning vehicles, ensuring the system's safety. While not referred to as such in the standard, we call to a set of threshold values of kinematic events as a *service profile*.

The construction of CopaDrive also enabled different studies on autonomous vehicles. Integrating a 3D and network simulator allowed the evaluation of the IEEE 802.15.4e protocol model evaluation. We assessed the use of this network as an enabler of an intra-vehicular sensing network, validating its application capability in real scenarios in the future. We also evaluated its use for accident avoidance in different conditions. This extension of CopaDrive also broadened the horizons for its applicability, such as using it for teaching as learning aids. The vehicle simulator was modified for control applications in different urban scenarios. In these scenarios, it was possible to describe, in a simple way, the use of various control parameters to increase the performance of autonomous vehicle handling and allow students to understand how different control characteristics impact handling under other conditions.

1.3 Thesis Statement

Co-VP applications have an inherent complexity, given their integration between communications and control systems and their need for real-time response for safety-critical tasks. Validating these applications before their use in real environments is critical to ensure their safety. Therefore, using reliable validation tools to integrate control and communication is essential in developing, testing, and proposing new models. The initial goal of this Thesis is to create this framework, showing its capabilities by solving open Co-VP issues caused by network threats. Therefore, this Thesis addresses the following problems:

1. Co-VP control strategy validation problem in a realistic scenario using ETSI ITS-G5 communication.

2. The integration problem between Lateral and Longitudinal Control of Co-VP applications in real moving conditions.

3. The problem of event triggers generating CAMs messages in Co-VP communications.

Following these problems, we want to answer the following research questions:

1. Is it possible to analyze the impacts of ETSI ITS-G5 communication on Co-VP control systems in an integrated and detailed way?

2. Considering several Co-VP application scenarios, is it possible to integrate the lateral and longitudinal controls, increasing the ability to follow the leader's trajectory and the platoon's safety?

3. Is it possible to improve the performance of the cooperative platoon by modifying the CAM messaging triggers without significantly increasing the number of packets transmitted?

Therefore, we state our Thesis as follows:

A realistic framework for Co-VP systems development and testing that integrates simulation, hardware-in-the-loop and testbed reduces solution development time, allows validating different performance optimization strategies, and enables the analysis of the impact of communications on the control, increasing application safety. Using such a framework that we call CopaDrive, we postulate that integrating lateral and longitudinal platooning control with tighter triggers for CAM messages provided by ETSI ITS-G5 improves Co-VP application performance without significantly increasing the load on the data network.

1.4 Contributions

Along the work towards proving our Thesis, we generated the following contributions to the state-of-the-art in cooperative vehicular platooning,

C1 A multidimensional analysis of Co-VP systems, integrating the different constitutive aspects of Co-CPS. We address platoon controller models and the impact of network failures on their performance, including inherent network threats and security issues.

C2 A flexible environment for integrated development, test, and validation, of cooperative driving applications that we call CopaDrive. Using ROS as an enabler and ITS-G5 as the network stack, CopaDrive evaluates Co-VP systems, allowing the study and joint analysis of controller and network aspects.

C3 A Hardware-in-the-Loop (HIL) and robotic testbed that extends CopaDrive while using ROS and integrating real ETSI ITS-G5 communication using OBUs and safety features.

C4 A V2V-enabled Co-VP Look Ahead Controller (LAC) with low complexity that reduces longitudinal and lateral errors, increasing platooning performance and safety, and solving the cutting corner problem.

C5 An improved CAM message profile for Co-VP applications, called Platoon Service Profile (PSP), wich increases platoon performance in critical scenarios without significant extra network payload.

C6 Demonstration of CopaDrive flexibility using it in other AVs applications, including evaluating different network communication standards like the IEEE 802.15.4e and a learning environment for control applications.

We performed a thorough Co-VP applications overview to achieve contribution C1. As presented in Chapter 4, we demonstrate the most impactful research about Co-VP applications and the current gaps in this study area. We also evaluated the different validation tools for these applications, including simulation and realistic frameworks. In this research, we figured out the vantages and disadvantages of these tools and highlighted the most critical challenges in Co-VP development. This research is submitted to Cyber-Physical Systems Journal (ISSN: 2333-5785).

Regarding C2 and C3, we developed the CopaDrive framework as an integrated tool to evaluate Co-VP systems. Initially, we developed CopaDrive to fill the gap of a simulator that would meet the need for studies on the impact of vehicular communication models on the control of these devices, using ROS as an enabler, Gazebo as a 3D visualization tool, and OMNET++ as a network simulator. We present the first [294] microscopic control Co-VP simulator integrated with ETSI ITS-G5.

Furthermore, the use of ROS has allowed us to present in [101] the integration of the simulator with a real systems validation platform, both from a HIL perspective and in a Robotic Testbed.

This implementation mitigates the limitations of validation tools for Co-VP systems based exclusively in simulations. It allows us to evaluate the impact of data transmissions using OBUs equipped with the ETSI ITS-G5 on vehicle control systems in a hybrid scenario (HIL) and on a fully embedded testbed. These results are presented in Chapter 5.

Contribution C4 addresses a problem often overlooked in Co-VP solutions: lateral control of vehicles. Several researchers consider that platooning uses complementary solutions such as cameras to follow the lateral movement of the leader, which may not be enough in different track conditions. We implemented the lateral control of the platoon, seeking to solve the problem defined as "cutting the corner" so that the following vehicle makes the same curve as the platoon leader, respecting its trajectory and keeping its distance. In this work, we extend the capacity of cars in the same platoon using a look-ahead controller. This controller uses part of the accumulated error between each vehicle to minimize the total error between the last follower and the platoon leader, thus increasing the safety of their movement. This contribution was published in [99] and is presented in Chapter 6.

ETSI ITS-G5 uses CAM messages as the primary way to disseminate V2V messages. However, not using specific message trigger profiles for Co-VP applications can be a risk factor due to potentially inappropriate communication patterns. Contribution C5 introduces a new profile for sending CAM messages, restricting the message-generating triggers. To analyze the impact of this change on the Co-VP system, we use CopaDrive to evaluate different message-sending profiles in different scenarios. We demonstrate the cooperative platoon's performance improvement achieved by reducing lateral and longitudinal errors by adopting PSP. We also evaluate the impacts of this change on the communication network and the number of packets sent and received, showing that adopting the more restricted profile increases application safety without significantly increasing network throughput. This work has been published in [98] and is presented in Chapter 7.

The integration of ROS with OMNET++ performed in CopaDrive has opened several possibilities for studies with different communication protocols and control algorithms. Contribution C6 demonstrates this flexibility with the use of IEEE 802.15.4e in an intra-vehicular sensor network to prevent accidents and increase the safety of applications with AVs. This work was published in [164]. Also, because of its ease of use, we demonstrated in [290] how the CopaDrive can be used to study new control techniques and algorithms in a learning environment. Both variations of CopaDrive are discussed in Chapter 8.

1.5 Thesis Outline

The organization of the rest of this dissertation was already presented, in detail, in the previous section when explaining the contributions, particularly Chapter 4 through Chapter 8. Before these, Chapter 2 presents the Co-VP main characteristics, models, and features. This chapter introduces the platoon stability concept and reviews the most common controllers used in this application. Next, Chapter 3 shows the most used vehicular networks and their main differences, emphasizing the ETSI ITS-G5. We also review some related works that indicate the advantages and disadvantages of each one. At the end, Chapter 9 presents our conclusions about the validation of the Thesis and associated contributions. We also propose potential future research topics regarding Co-VP applications and their validation tools.

Capítulo 2

Cooperative Vehicular Platooning Background

This chapter presents a brief review of Co-VP, including the most relevant surveys found in the literature that summarizes the recent research in this area and correlated ones. We also present, in detail, the mathematical model to be used and the metrics that indicate platoon safety. Finally, we present the main control models used, emphasizing their complexities and solutions.

2.1 Overview

With the increasing interest in cooperative platooning in the industrial and academic areas, many studies have been performed. Several are related to the implications of using platooning in real-life situations and its impacts on society. In contrast, others are concerned with this kind of application's communication and cooperative aspects. Finally, others focus on the control models of the platoon, aiming to guarantee the response time of the vehicles in different situations.

One of the biggest challenges of governments relies on the constant growth of the number of vehicles in the cities and roads. Even in a fully connected world, where many people can work from home, many products still need to be transported and delivered at different points. Moreover, as the construction of new roads is not sustainable, changing the driving approach from manual driving to a platoon-based driven model can improve the mobility and traffic in cities and on roads [121].

For instance, as the vehicles in a platooning are near, the road capacity is increased at the same time as the traffic congestion declines. For the same reason, the energy efficiency and the emissions fall due to the reduction of the air resistance [287]. This driven pattern also improves many applications that rely on communication, given the almost static distance between the vehicles. Regarding the driver, driving in a platoon can be safer and more comfortable since the vehicle follows other cars and guarantee the safety of the members [152].

In literature, the platoon pattern is defined in [142] as a group of vehicles with common interests, where a vehicle follows another one, maintaining a small and safe distance from the previous one. Those vehicles assume a cooperative driving pattern, following the platoon leader. An example of those platoon patterns can be seen in Figure 2.1. The management of the platoon can have many operations, like formation, merging, maintaining, and splitting, among others. Those operations demand synchronized behavior and control between the vehicles.

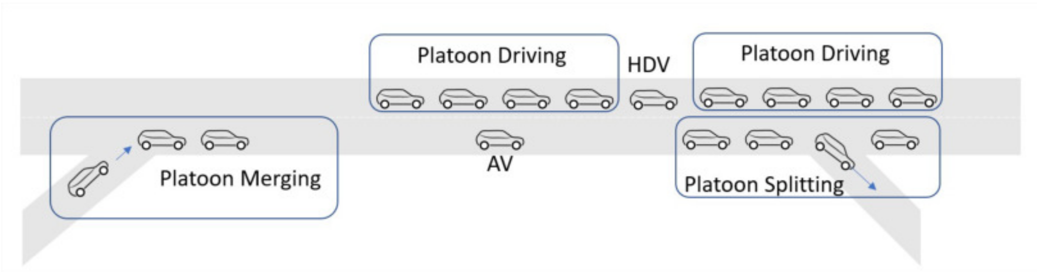


Figura 2.1: Driving Patterns for Platooning

Many sensors can manage the control and behavior of the AVs. However, wireless communication can improve the safety and reliability of Co-VP applications. ITS are defined as "*advanced applications which without embodying intelligence as such aim to provide innovative services relating to different modes of transport and traffic management and enable various users to be better informed and make safer, more coordinated and 'smarter' use of transport networks*," by the European Union [254]. Each vehicle can carry an OBU, responsible for collecting the data from the sensors and sending them to the neighbors or to *Road-Side-Units* (RSU). Those RSU can be accountable for analyzing the data and redistributing them or even alerting the vehicles to the road conditions.

Considering the vehicles that form a platoon as a complex system that defines the course of action based on sensors and communication with others, it is possible to describe them as Co-CPS [299]. They can be analyzed in two points: intra-vehicle CPS, where the main goal is to optimize the response time and the performance of each car, and inter-vehicular, where the objective is to improve the traffic or the network behavior of the platoon. In both cases, the platoon's safety is the most critical parameter to be guaranteed. The design of the applications and complete visualization of platoons as Co-CPS is illustrated in figure 2.2.

The most common Vehicle Platooning (VP) applications focus on optimizing traffic, increasing energy efficiency, and service delivery to vehicles, like information about the road. There are many studies on those areas, such as [112], where the authors present an analysis of the traffic flow, considering an autonomous platoon and a mix of human-driven vehicles and autonomous vehicles. Regarding the fuel consumption, [70] evaluates the reduction caused by platooning, given the small distance between the cars. There is also a study developed in [279], where the decrease in emissions of CO2 is analyzed. Enabling communication between vehicles can also support other services, like internet access and local cooperative services [329, 61], where vehicles contribute part of their buffers to replicate data for others in the same platoon and share data

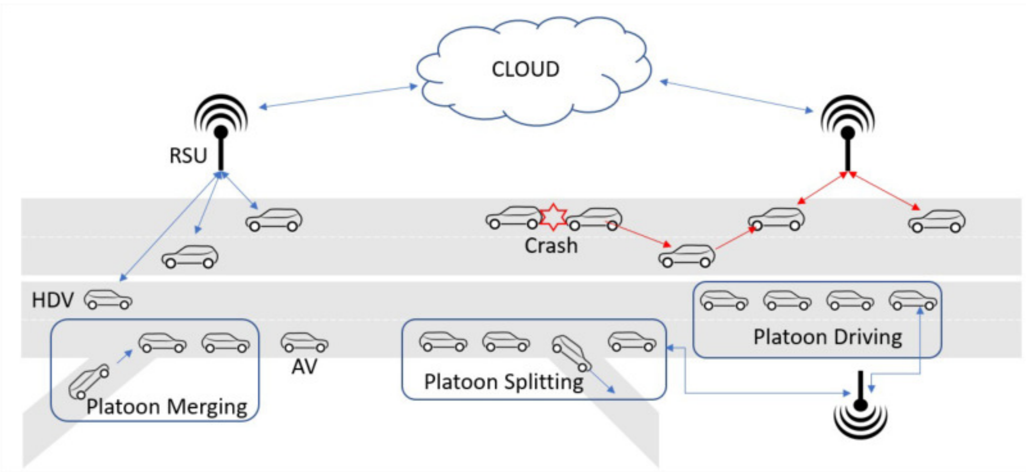


Figura 2.2: Driving Applications

with them.

2.2 Related Surveys

Co-VP is a complex and multi-disciplinary subject. A true example of the cooperative CPS paradigm in which the physical and cyber aspects of the system become highly integrated. To further increase the complexity, each vehicle consists of a system interacting with the remaining platoon members via different communication transactions to form Systems-of-Systems (SoS). With this in mind, we looked for surveys that addressed this complexity in both Co-VP and complementary topics. These works encompass a fundamental multi-disciplinary perspective to accurately describe, model, develop, implement, and validate these SoS's. That is the approach we follow and present in what follows.

We identified surveys covering the background of Cooperative Autonomous Vehicles (CAV), Vehicular ad-hoc networks (VANETs), Co-VP techniques and controllers, and vehicular validation frameworks. Table 2.1 summarizes the topics addressed by each work and positions our work concerning these. We cross-checked these with topics we covered in our work, such as the relationship with Co-VP applications, communication infrastructures, control systems, safety and security, and validation tools. We adopted the criterion of '$-$', indicating that the topic was not covered, '$+/-$' for incomplete coverage, and finally, '$+$', indicating a more complete and integrated topic coverage.

As shown, we found that none of these completely addresses these topics, and quite often, neither does their interdependence. In what follows, we highlight our findings from this analysis of the state-of-the-art.

Tabela 2.1: Related Surveys Comparison

Topic	Reference	Year	Address Co-VP	V2X Study	Controller Analysis	Safety Analysis	Security Threats	Validation Tools
Co-VP Architecture	[152]	2011	+	+/−	+/−	+/−	−	−
	[119]	2018	+	−	+/−	−	−	−
	[162]	2019	+	−	−	+/−	−	−
V2X Communication	[322]	2020	+	+	−	−	+/−	−
	[188]	2019	−	−	−	−	+	
	[169]	2017	+	+	−	+/−	+/−	−
	[142]	2016	+	+	+/−	+/−	−	+/−
Controller Strategies	[175]	2017	+	−	+/−	−	−	
	[326]	2020	+	−	+	−	−	−
	[258]	2018	+	−	+/−	−	−	−
	[304]	2018	+	−	+/−	−	−	−
	[35]	2016	+	−	+/−	+/−	−	−
Validation	[301]	2019	−	+	−	+/−	+/−	+/−
	[85]	2020	+	+/−	−	−	−	+/−
Our Work		2021	+	+	+	+	+	+

2.2.1 Co-VP Architecture Surveys

One of the first surveys in vehicular platooning was presented in [152]. This survey introduces several concepts such as string stability and considers Co-VP a natural development of vehicular platooning with Adaptative Cruise Control (ACC), introducing V2I and V2V communication. In addition, they gather several works in Obstacle Detection and Collision avoidance, Inter-vehicle communication, string stability, and control strategies. However, there is no reference to the current protocols regarding inter-vehicle message transmission.

In [119], the focus was on a control and planning architecture for Cooperative Autonomous Vehicles (CAVs), observing techniques to improve energy efficiency. First, they defined the Co-VP as one of some Real-Time motion planning techniques for CAVs and the ACCs evolution, removing the limitations of perceptions systems. Then, they performed a brief review of the vehicular communication protocols. In this survey, the authors defined the control analysis of the Co-VP as a one-dimensional networked dynamic system, decoupling the lateral and longitudinal Co-VP controllers. Considering the inter-vehicular distance as the primary metric of the platooning, they suggested some control strategies, like Model Predictive Controllers (MPC) and Linear Consensus. Finally, the authors demonstrate that some basic level of centralized coordination is still necessary for the geometry and information flow network regarding platoon coordination.

A general view of the platoon coordination in most common car maneuvers is presented in [162]. The authors explain the movements, e.g., join, merge, leave and split, as illustrated in 2.2. They also overview some intra-vehicle and inter-vehicle connectivity works, mainly regarding DSRC/WAVE[67] for V2V.

The authors of [142] provide a wide survey on Co-VP, which is described as a complex physical system that integrates modern wireless communication technologies and can be classified as a CPS due to its integration of computing, communication, and control technologies. The survey is organized into several fundamental topics, including modeling, management, stability analysis, platoon driving models, and V2X communication models, and also discusses validation methods through simulators. While the survey offers an integrated view of Co-VP by establishing links between Co-VP architectures, their control methods, and network communication protocols, it does not address security concerns and only briefly covers validation tools.

2.2.2 Co-VP Controller and Efficiency Surveys

The authors of [304] primarily examine Cooperative Adaptive Cruise Control (CACC) and its architecture, which is structured around perception, planning, and actuation layers. Their work is mainly focused on longitudinal controllers that ensure string stability. They discuss MPC, Consensus, Optimal controllers, and Co-VP as key aspects of CACC technology. Meanwhile, the authors of [175] take a complementary approach, providing an overview of the performance of four distributed control models - linear consensus control, distributed robust control, distributed sliding mode control, and distributed MPC - in terms of internal stability, stability margin, string stability, and coherence behavior specifically for Co-VP.

The significance of MPC in the Co-VP literature is apparent, as evidenced by [35], which surveys the outcomes obtained using distributed MPC for Co-VP and offers a real-time MPC implementation. In contrast, [258] focuses on formation control of Co-VP and surveys various distributed and decentralized methods for vehicle formation control, highlighting the technical and implementation difficulties associated with these control methods through different topologies, which are presented in Figure 2.5. Another area of interest in Co-VP research is energy consumption efficiency, explored in [326]. The authors of this paper examine fuel economy in truck platooning and analyze contributing fuel consumption factors, such as various coordination methods and look-ahead control strategies.

The surveys on this topic do not consider the inter-dependencies with the communications topics in-depth. In most cases, there is no debate on limitations the communications infrastructures may impose over the control systems or even if current V2X communication standards can support some of the proposals. Often, the approach only considers control practices, completely neglecting that such controllers will take part in a complex SoS, and the communication interactions and their constraints must be analyzed to understand their effectiveness correctly.

2.2.3 Co-VP Test and Validation Surveys

Unfortunately, just a few surveys related to Co-VP cover the validation process of a Co-VP system as a central theme. Two highlighted surveys in this area are presented in [301] and [85]. In the first one, the authors reinforce the importance of the reliability and maturity of the technology that should be tested and verified. In the first survey, they summarized the testing methods for

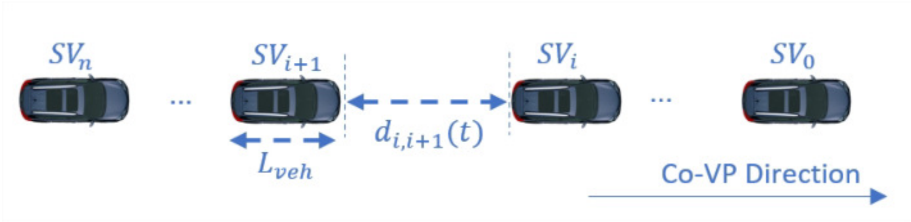


Figura 2.3: Basic Car-Following (B-CF) schematic

the V2X communication process using LTE-V and DSRC. This work also emphasizes the important network challenges such as latency/reliability and security. The second survey [85] is more dedicated to Co-VP Validation Strategies based on simulation, real experimentation, formal verification, and testing. However, both fail to address the fundamental topic of validating a complete SoS in all its fundamental aspects and interactions.

As seen, there is a lack of surveys that address the multi-disciplinary properties of Co-VP in a competent fashion. Most take on Co-VP from a single topic-based and thus quite limited perspective. In our survey, we consider this topic of inter-dependency, fundamental to all cooperative CPS, a core of our survey by overviewing the recent advances in the different relevant topics and highlighting how these overlaps are considered in the literature. In the next section, we introduce the first topic of the sour survey.

2.3 Co-VP Formal Model

The formal definition of a Co-VP system involves the mathematical definition of its components and the relationships that define the safety of its movement in the form of functional and non-functional parameters. This section presents the basic mathematical definition of Co-VP applications, including the most used IFTs. In addition, it introduces the concepts of platooning stability based on distance measurements between vehicles. Importantly, by introducing these basic tenets of Co-VP control concepts, we ease the understanding of the control strategies. Finally, the section also presents the advances in the control models used in centralized or distributed architectures that allow the implementation of these cooperative systems. This session diagram is illustrated in Figure 2.4.

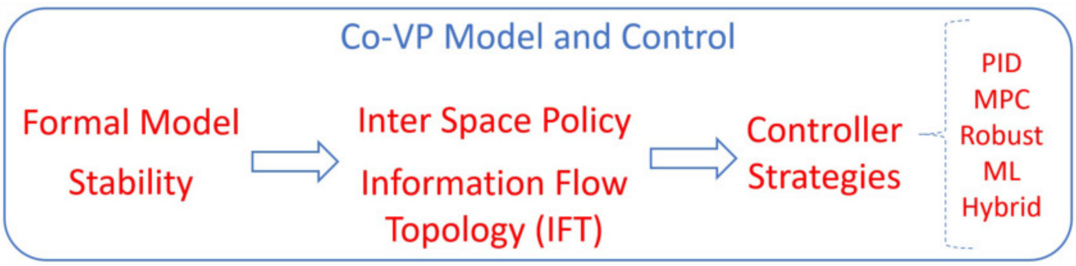


Figura 2.4: Section 2.3 general diagram

The vehicle controller has a crucial role in the Co-VP implementations. Even under perfect network communication, the received data should be analyzed and processed together with the own sensor's information in time to guarantee the platoon's safety. These controllers can be modeled in several ways with different approaches. This section will introduce the central platoon characteristics, with the most used controllers, comparing their applications in several works.

Those models use to emulate human behavior, enable other drivers how to interact with the vehicle, and provide passengers with comfortable driving [337]. Furthermore, to analyze the immediate interactions among vehicles in Co-VP, it is common to apply microscopic traffic models to explore different vehicle dynamics, including response time, transient and steady response of a car, regarding space between vehicles, velocity, and acceleration, among others [142].

The car-following (CF) microscopic traffic model is one of the most used theoretical references for autonomous car-following systems [251]. It models the strong interaction between vehicles with tight space between them. These CF models can be grouped in three categories: basic CF models (B-CF), time-delayed CF models (TD-CF), and multi- anticipative/cooperative CF models (MAC–CF) [267]. The typical car following schematic is illustrated in Figure 2.3, where identical vehicles follow each other in a single line with no overtaking.

The platoon is composed of $i \in \mathbb{N}$ vehicles. The full set of vehicles can be defined as $SV_i = \{i \in \mathbb{N} | 0 \leq i \leq n\}$, with a set of SVs, where SV_0 is the first vehicle and the platoon's Leader (TV). Each SV_i can be a local leader of SV_{i+1} and a follower of SV_{i-1}. The SV_i vehicle's position, speed and acceleration at time t is denoted as $x_i(t)$, $v_i(t)$ and $a_i(t)$, respectively. The distance - $\Delta d_{i+1}(t) = x_i(t) - x_{i+1}(t)$ - and the speed difference - $\Delta v_{i+1}(t) = v_i(t) - v_{i+1}(t)$ - are crucial to the CF model.

It is possible to represent the vehicle response in in a time-continuous model using the acceleration a_{i+1} in terms of $\Delta d_{i+1}(t)$ and $\Delta v_{i+1}(t)$ to SV_i and v_{i+1} and a set of ordinary differential equations (ODEs) [309]:

$$\dot{x}_{i+1}(t) = v_{i+1}(t) \tag{2.1}$$

$$\dot{v}_{i+1}(t) = f(d_{i+1}(t), v_{i+1}(t), \Delta v_{i+1}(t)) \tag{2.2}$$

In this model, the current state of SV_i defines the mobility of SV_{i+1}. It is crucial to notice that the B-CF model does not address the vehicle's lateral controller, considering only the longitudinal aspects of the Co-VP.

2.4 Co-VP Stability Analysis

The CF model's instability is usually responsible for traffic congestion, stop/slow-and-go oscillations, and even accidents with CAVs. Furthermore, with changes in speed and distance between the vehicles, the increase of perturbation causes instability and decreases the Co-VP safety. Considering the Co-VP longitudinal controller, the stability analysis studies how the errors of SV_i position evolve [335].

The linear stability analysis focus on the influence of minor disturbances over the Co-VP. Considering that the Co-VP applications were developed for road traffic, applying this model to platooning studies with almost constant speed is suitable. The authors of [267] reviewed and mathematically defined several stability models, like local and asymptotic stability. The first is Lyapunov stability, where any sufficiently small initial perturbation remains small. The second one is where any small concern tends to zero as time advances to infinity. Considering the traffic flow, the authors consider the local stability as the stability of a single-vehicle over some slight disturbance. In contrast, asymptotic stability considers the Co-VP stability (or string stability).

In this work, we will use the string stability concept definition from [277, 314, 207, 213], that is based on the spacing error between the real and the desired inter-distance between SV_i and SV_{i+1} [246]. The string stability requires that the disturbance strictly attenuates between each leader-follower pair as it propagates away from the SV_0. The spacing error for the $SV_{i+1}th$ vehicle can be determined using:

$$\varepsilon_{i+1}(t) = SV_i(t) - SV_{i+1}(t) + d_{des}, \tag{2.3}$$

were d_{des} is the desired intra-platoon distance. The steady-state error transfer function is defined by:

$$H_i(s) = \frac{\varepsilon_{i+1}}{\varepsilon_i}, \tag{2.4}$$

where the platoon string stability is guaranteed if $\|H_i(s)\|_\infty \leq 1$ and $h(t) > 0$, where $h(t)$ is the impulse response corresponding to $H(s)$. These string stability definition use the $\mathscr{L}_2$ norms, where $\|H_i(s)\|_\infty = max_t|\varepsilon_i(t)|$ is the maximum magnitude of the perturbation within infinite time. This metric characterizes the Co-VP string stability worst-case performance, using the maximum of the frequency response of the transfer function from the perturbation to ε_{i+1}. The authors of [333] propose a more flexible stability analysis, defining the string stability as $\mathscr{L}_\infty$, to guarantee the absence of overshoot for a signal while it propagates throughout the platoon. In this approach, it is possible to guarantee the Co-VP local stability in a string with n vehicles if:

$$H(s) = \frac{\varepsilon_n}{\varepsilon_1} < 1. \tag{2.5}$$

Several factors have direct influence over the Co-VP stability. Namely, the *vehicle parameters*, that include delays in response time, maximum heading and speed, the *spacing policy*, that refers to the distance between SV_i and SV_{i+1}, the *control model* and the *communication structure*.

Regarding the Vehicle parameters, the uncertainties about the model are the most concerning problem. The authors of [43] illustrate a solution, using centralized Co-VP control, using an MPC strategy, where the accelerations of the vehicles are generated considering the worst-case scenario for each one of the vehicles. The use of a stochastic approach to predict the behavior of the cars demonstrates a flexible way to maintain platoon stability, even mixed with human drive vehicles. In this work, all the followers send data to the platoon leader that defines the strategy and sends it back to the followers in a *Leader-followers* approach. The authors from [319] propose a new delayed feedback MPC scheme to deal with sensors with limited measurement range and actuator time delay. They also present controllers that adapt their parameters online, solving a receding horizon optimal control problem.

In [340], the authors consider a Co-VP with a nonlinear CAVs model, with parametric uncertainty and unknown external disturbance. So, they proposed an online estimator based on V2V communication with an adaptive backstepping control scheme. This work employs asymmetric time-varying constraints to avoid spacing error growth. Another approach was presented in [91], where the authors distinguished types of disturbances and used different ways to handle them. This work applied a feedforward controller for large yet infrequent perturbations and a feedback controller for slight yet frequent variations. They demonstrated that this controller outperforms the standard MPC implementation.

2.5 Co-VP Inter-Space Policy

The majority of Co-VP adopt one of two common inter vehicles space methodologies: *constant spacing policy* (CSP), which is independent of the speed of the controlled vehicle [40]; and *constant time-headway policy* (CTHP), that uses the current speed of the vehicle to define the safety distance [57]. The CTHP is usually recognized as a safe practice for human drivers. The objective range (d_{ref}) in this policy is $d_{ref}(t) = SD + t_h v_{i+1}(t)$, where $SD > 0$ is the safety distance and t_h is the defined time headway, generally between 0.5 and 2 seconds.

While using CTHP is common in Co-VP, [146] propose a new method that improves on this approach by using a non-linear range policy. This policy decouples t_h from the time constant of the vehicle's mechanical control loop and is obtained through an optimization procedure with traffic flow and stability constraints. Tests show that this method achieves stable traffic flow up to a significantly higher traffic density, even with different vehicle models. In a different study, [39] examine inter-platoon stability by extending the CTHP approach to study the flow of many platoons, considering the whole traffic flow as the interaction between cooperative platoons. Platoon leaders receive information from other platoon leaders via the V2I communication strategy, and a virtual leader is used for the entire platoon.

Other variations have also been explored in different studies. For instance, in [191], vehicles use onboard sensors to maintain distance between them and receive a_i from the local leader via V2V communication. In [300], a flexible, safe distance constraint ensures safe distance and communication connectivity in the platoon. This method allows a CAV to meet with the platoon within any preset time without being bound to initial requirements or system parameters.

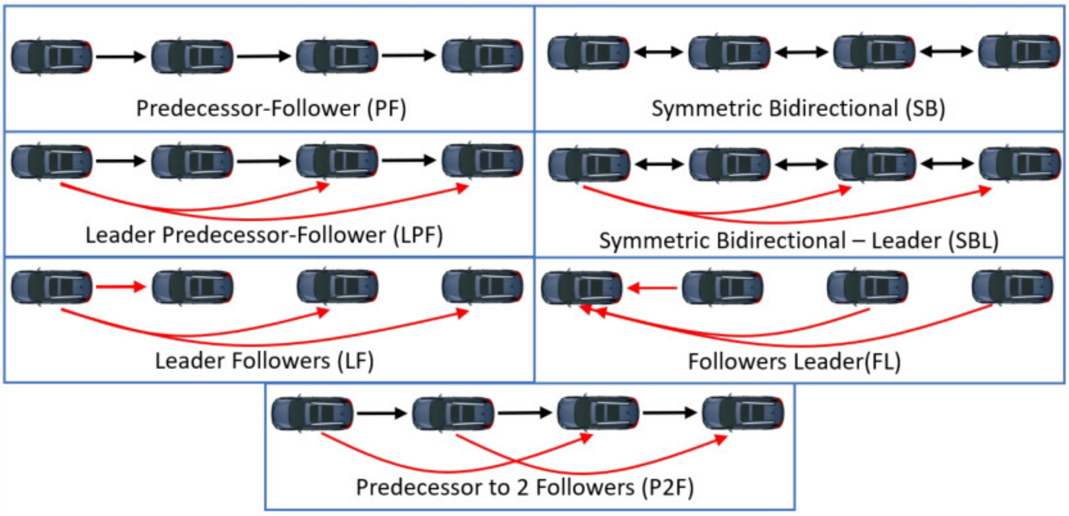


Figura 2.5: Co-VP Information Flow Topology (IFT)

2.5.1 Co-VP Information Flow Topology

The communication structure has a significant role in platoon stability, as explained in [258]. Messages size, information type to be transmitted, and the message distribution's topology all matter in the stability control. The most common strategies are the *predecessor-follower* (PF) method [150], and the *symmetric bidirectional communication* (SB) [266], although others can be used, as presented in Fig. 2.5. In the first approach, the SV_i sends messages to SV_{i+1} and receives no messages from them. In the *bidirectional* communication, the messages are also sent from SV_{i+1} to SV_i. The PF communication also has some different implementations, like in [177], where the authors investigate a merging algorithm for the vehicles to join a platoon. In this work, the vehicle receives data from the previous two vehicles in the platoon to perform a *consensus* algorithm, defining a *predecessor to 2 followers*, or P2F. The authors of [109] proposed a granulated *predecessor leader-follower*, in order to create a scalable platoon. The leader transmits its messages to the next two vehicles in this work. Then, the last one became a *G-leader* and is responsible for reproducing the leader information to the next two vehicles and the next *G-leader*. The authors could keep the stability of 8 vehicles platooning in several conditions.

Based on several studies, it is possible to improve the platoon stability using some methods: broadcast the information of the leader to all the vehicles [40]; using CTHP instead of CSP [57]; non-linear spacing policies and non-identical controllers [155]; sending messages in both ways - from previous cars to next cars and in opposite direction [110]. However, broadcasting information to all vehicles and providing bidirectional communication between vehicles is a strategy that decreases its benefits as the platooning size increases [123].

2.5.2 Co-VP Controller Strategies

Several control models, ranging from simplified controllers to very complex ones, can be applied in Co-VP. Such models directly influence the response time of the platooning applications and are also responsible for guaranteeing the platoon's stability in different situations.

2.5.2.1 Co-VP PID Controllers:

Although considered simple, the Proportional Integral Derivative (PID) controller is a solution widely used in Co-VP applications. For instance, in [289, 150], two PID controllers are integrated to perform lateral and longitudinal control of a platooning. A maximum vehicle number was determined to guarantee the platoon's safety in both works.

In [113], a modified version of the PID control is implemented. The authors use an adaptive Proportional Derivative (PD) controller to ensure platoon stability. In addition, they propose a dynamic information exchange mechanism between vehicles based on a predecessor-follower mode. Under this scheme, the information about the previous vehicle is transmitted to the next two vehicles in the platoon. Furthermore, the sensors detect the distance and position of the preceding vehicle. Thus, when communication is lost, the controller does not immediately switch to a degraded mode but rather attempts to compensate for the communication failures.

2.5.2.2 Co-VP MPC Controllers:

Another usual controller for Co-VP applications is the MPC and its variations. For example, the authors of [327] compare the PID with the MPC to maintain inter-vehicular distance and headway time between the vehicles. They used the VISSIM simulator combining CSP and CHTP for safety analysis. The authors showed that MPC improves the platoon's control performance in this case. However, in this work, the communication prerogative is that all the vehicles can share their data with the surrounding vehicles without data loss. Therefore, the authors also conclude that a well-tuned PID can maintain platoon stability with less computational power, but the tuning can be very empirical and inefficient. Otherwise, the parameters adjusted in MPC are more comfortable and can also help with lost packets in communications.

In [127], an MPC is used to allow the platoon interaction with an HDV, *joining* and/or *spliting* from a Co-VP, using CHTP as a safety condition. This approach uses a centralized node (an RSU or another vehicle) to publish to all vehicles on the road. This work does not consider the communication range, allowing the centralized node to receive data from all the platoon vehicles.

However, in [334], centralized MPC is considered challenging to implement in real Co-VP applications, given the system's dynamics. This work proposes a Distributed MPC (DMPC), where the algorithm in each car does not need the leader's information but only from its neighbors. There is an optimal local solution for each vehicle that does not need to *a priori* know the entire platoon's desired set point. In this case, they consider that only the followers directly communicating with the leader know the desired path. Then, they introduce a constraint in the follower's position based on the neighbor's information. The authors address their tests in different unidirectional topologies, like predecessor-following, predecessor-leader following, two-predecessor following, and two-predecessor-leader following.

In [125], the authors conducted numerical simulations to compare the fuel efficiency of a new distributed EMPC strategy with a commonly used distributed target-tracking MPC strategy. The proposed strategy aims to enhance fuel efficiency while ensuring the platoon's stability and string stability. To achieve this, they utilized the fuel consumption functions of the vehicles as the objective cost of the distributed MPC.

2.5.2.3 Co-VP Robust Controllers:

The authors of [153] propose a decentralized control approach to address platoon stability by formulating a multi-objective H_∞ control. The objective of this control is to guarantee string stability of a vehicle platoon in ACC and CACC while allowing tradeoffs between vehicle following performance, system robustness, and string stability. Two scenarios are considered in this work: one with ACC using only local sensors and another with CACC using vehicle communications. In case of communication failure, the vehicle operates as an ACC. Similarly, [225] also employs H_∞ control to support multiple-vehicle look-ahead CACC design, considering linear platooning. This work uses a novel definition of $\mathcal{L}_2$ string stability. The communication approach uses a predecessor-follower and two-predecessor follower to analyze how communication complexity impacts the system's performance.

2.5.2.4 Co-VP Controller design with Machine Learning:

The use of machine learning (ML) models in Co-VP applications is still restricted, but some studies have explored their use as an alternative to control parameters. For example, a study presented in [174] aimed to reduce fuel consumption by using an IFT-PF to transmit the state of the agents globally and a specific channel to establish rewards of the DRL model. The proposed approach considers the multi-agent variation inherent in platooning, including vehicle inputs and outputs. Additionally, [41] introduced a path planning scheme that utilizes DRL on the network edge node for improving the driving efficiency of autonomous vehicular platoons in terms of fuel consumption. The proposed approach considers a joint optimization problem that factors in the task deadline and fuel consumption of each vehicle in the platoon.

The authors of [318] focused on the longitudinal controller and proposed the use of a longitudinal PID controller for platooning. The optimal parameter tuning was performed as a goal of a deep reinforcement learning (DRL) model. The authors claimed a reduction in stability time and distance error using a Hardware-in-the-Loop (HIL) as a validation tool. Similarly, [183] proposed an integrated approach that combines DRL and dynamic programming (DP) to develop efficient vehicle tracking policies in Co-VP scenarios. The proposed system, FH-DDPG-SS, uses three key ideas to improve efficiency: transferring network weights backward in time, approximating stationary policies, and scanning through reduced state space. However, this paper did not compare its results with other scenarios.

Tabela 2.2: Summary of Control Models for Co-VP

Controller Strategy	Cite	Year	Description	Model Applied Issues	Space Policy	IFT
Linear	[146]	2005	Distributed	Actuator Lag	CHTP	-
	[39]	2019	Linear	Communication delay	CHTP	BSF
	[191]	2020	Distributed	-	CHTP	PF
	[300]	2019	Distributed	-	CSP	PF
	[266]	2017	Distributed	Degraded Communication	CSP	PF
	[177]	2016	Distributed Consensus	Communication Delay	CSP	PF
	[109]	2019	Distributed	-	CSP	-
	[40]	2017	Distributed	Actuator Lag	CHTP	PF
	[57]	2015	Distributed Consensus	Communication Delay	CHTP	LF
	[155]	2004	Distributed	-	CSP	PF
	[110]	2013	Distributed	-	CSP	SBF
	[123]	2013	Distributed Non Linear	-	CHTP	SBF
PID	[150]	2017	Distributed	Degraded Communication	CSP	PF
	[99]	2020	Distributed	Actuator Delay	CHTP	PF
	[113]	2018	Distributed	Degraded mode with compensation	CHTP	P2F
MPC	[43]	2018	Centralized	Actuator Lag	CSP	FL
	[319]	2018	Centralized and Delayed	Lag Sensors	CSP	LF
	[327]	2018	Distributed	-	CHTP	SBF
	[127]	2018	Centralized	-	CHTP	FL
	[334]	2017	Distributed	-	CHTP	PF
	[125]	2020	Distributed	-	CSP	PF
	[91]	2020	Distributed Feedforward and Feedback	Unmodeled Dynamics and Initial Tracking Error	CTHP	PF
Robust	[340]	2020	Distributed Adaptive Backstepping	Parametric uncertainty and disturbance	CTHP	PF
	[153]	2017	Distributed H_∞	Degraded Communication	CHTP	PF
	[225]	2014	Distributed H_∞	Real vehicles with ETSI-G5	CHTP	P2F
ML/ DRL	[318]	2020	Centralized	PID tuning	CTHP	PF
	[183]	2023	Distributed	Dynamic Program	CSP	PF
	[171]	2022	Distributed	V2V Communication	CHTP	PF

Furthermore, the work presented in [171] investigates the impact of V2X communications on platoon control performance using DRL. The study explores the tradeoff between the gain of including exogenous information in the system state for reducing uncertainty and the performance erosion due to the curse of dimensionality. The study determines the most appropriate state space for platoon control under different information topologies and quantifies the value of each piece of information to establish the most optimal policy. Additionally, [44] proposes a model-based DRL algorithm for the CACC of connected vehicles, including a platoon of both human-driven and connected AVs via V2V and vehicle-to-cloud communication. However, these implementations are theoretically validated, and their effectiveness in real-world scenarios needs further investigation.

2.6 Conclusion

The choice of controller model for Co-VP applications greatly influences the system's performance and safety. Responsible for ensuring the platoon's stability, its complexity can directly affect its applicability in the real world, considering communication errors, processing time, and these algorithms' responses. Thus, the choice depends heavily on the mechanical systems' responsiveness and the vehicular network's communication capacity. Table 2.2 summarizes the works presented in this section, comprehensively comparing the controller strategies.

The works presented in sections 2.3, 2.4, 2.5, 2.5.1, and 2.5.2 are summarized in Table 2.2, providing a comprehensive comparison between the controller strategies.

Capítulo 3

V2V Communications

This chapter introduces the main models of vehicular communication networks, facilitating the implementation of Co-VP applications. We emphasize the ETSI ITS-G5 model because it is the most widely used in Europe and was later adopted in our experiments during the development of this dissertation.

3.1 Overview

V2X communication refers to a communication paradigm that exchanges data between vehicles and road infrastructure over a dedicated network, and it includes V2V and V2I communication. This work will focus mainly on V2V communication and its application to Co-VP. A crucial requirement for allowing inter-vehicular communication is the efficient and fast exchange of relevant messages that increase road safety. Several technical challenges must be addressed to meet these requirements, like low latency, high reliability, guaranteed data rate, and security [81].

Some organizations have been working towards standardizing these V2X networks, and two of the most prominent standards are Wireless Access in Vehicular Environments (WAVE) [67] and European Telecommunications Standards Institute (ETSI) ITS-G5. They have been developed respectively in North America and Europe, and both are supported by IEEE 802.11p [72], and DSRC, which has been mandatory for light vehicles in the U.S. since 2020 [282]. In addition, however, other technologies like LTE-V2X had been tested for Co-VP applications with excellent results.

As will be presented in section 3.2, the majority of related surveys in V2X and Co-VP uses WAVE and DSRC as the primary communication standard. However, since 2019, Europe has defined the ITS-G5 standard as the base for direct V2X communication [285]. So, in this section, we will present a comprehensive review of the ITS-G5 standard, with the essential characteristics. Then, we will give a brief review of DSRC-WAVE and LTE-V2X, offering some comparisons between the technologies. Finally, as a summary, we present, in table 3.1, a brief resume of the main characteristics of these standards.

Tabela 3.1: V2V communication standards

Standard	Frequency (GHz)	Band. (MB)	Range (m)	Base Model	Main Consortia
ITS-G5	5.85-5.915	10	300	IEEE 802.11p	ETSI and C2C-CC
DSRC-WAVE	5.85-5.925	10	500	IEEE 802.11p	SAE
C-V2V	5.85-5.925	20	300	LTE Cellular Network	3GPP

3.2 V2X Literature Review

Several surveys have reviewed the V2X literature. For instance, [221] studies the existing inter-vehicle communication techniques, analyzing how these models deal with an unstable network environment. So, each vehicle should inform periodically nearby about their conditions. Then, if the number of vehicles increases, the number of messages increases, raising the packet collision probability and inducing the latency problem. Then, the survey compares some techniques that support inter-vehicle communication and the factors that better sustain the platoon. This survey focuses on Medium Access Control (MAC) techniques for efficient use of shared wireless transmissions, divided into contention-based and contention-free methods. Using the IEEE 802.11p standard [131], the authors compare the Carrier Sense Multiple Access – Collision Avoidance (CSMA/CA) as a contention-based method and the Frequency Division Multiple Access (FDMA) and Time Division Multiple Access (TDMA) as contention-free methods. Nevertheless, this work does not analyze network problems' impact on the platoon's stability or the congestion treatment methods provided by vehicular networks protocols.

Architectural problems and wireless technologies that enable V2V communications are the focus of the work presented in [322]. This research also highlights these networks' security issues and discusses the use of vehicular fog computing in enabling V2V communications. The authors reviewed several V2V trend topics, namely safety applications, mainly regarding collision avoidance and security, and compared certificated and non-certificated solutions for V2V. The general V2V architecture is presented as an integration of sensors that detect the vehicle conditions, processor, and decision-making, responsible for coordinating the actions, Global Positioning System (GPS), and the communication radio. Some V2V communication protocols are briefly introduced: DSRC, WAVE, 4G LTE, and 5G. Regarding V2V networks for platooning, the authors define the main challenges, like the formation, management, efficiency, information flow topology (IFT), and reliability in highly dense areas. However, they did not address the network impact over the control methods or the validation strategies of the Co-VP implementation.

An extensive review about Cooperative Vehicular Networks (CVN) was introduced in [63], where the authors cover relevant characteristics of CVN, like physical, MAC, routing protocols, link scheduling, and security. This work aggregates several research topics: physical layer cooperation, MAC protocols, routing/forwarding mechanisms, cooperative link scheduling, performance analysis, power/resource allocation, cooperative group communication, and secure cooperative communication. Furthermore, the authors proposed a taxonomy for CVN based on objectives, cooperative transmission modes, cooperation-based network functions, cooperating device types, and communication technologies. Finally, they defined some requirements for CVN improvement based on adaptive transmission power control, optimal cooperative relay selection, minimal coordination overhead, friendly cooperative transmission, fair resource allocation, and effective incentive mechanisms. However, those requirements need further proof and validation since there is no analysis of their impact on a Co-VP.

The authors of [272] present an innovative view of vehicular networks, empowered by several Machine Learning (ML) techniques. Those techniques, applied in communications, networking, and security of vehicular CVN, allow the evolution and implementation of intelligent radio and network intelligence to be used in artificial intelligence (AI) enabled next-generation (6G) networks. In this work, the authors stated the current challenges in vehicular communication with DSRC and the LTE-V - the extension of LTE - proposed by 3GPP. These challenges are divided into multi-radio access, dynamic radio configuration, network resource allocation, and network traffic control and the suggested solutions presented in the literature for those challenges. Furthermore, the security challenge for vehicular networks is also addressed in this work, with the same challenges as a wireless network plus the specific ones provided by vehicular networks, like the presence of malicious vehicle behaviors, Co-VP applications, and the vehicle's particular complex security.

Regarding the security issues in vehicular networks, [188] divides this topic into three main challenges: security, privacy, and trust. With the main focus on anonymous authentication schemes, this work briefly reviews VANETs, defining system architecture, communication patterns, and V2X standards - DSRC, Wave, and IEEE 802.11p. They list the security keys as availability, confidentiality, authenticity, data integrity, and non-repudiation, explaining the services that should be provided and their corresponding threats and attacks. However, this work does not address specific cooperative vehicular applications, like Co-VP. This study is fundamental, as the cooperative nature of these SoS introduces additional security vulnerabilities which expose new risks.

Addressing Co-VP applications, the authors of [169] highlight some works about platooning in an adversarial environment, where an attacker modification of some of the control parameters can lead to string and system instabilities, reducing the platoon's safety. Hence, this work presents techniques to detect and mitigate this attack. Still, this work presents preliminary results, with few variations of attacks and defense mechanisms.

3.3 The ITS-G5 Standard

The ITS-G5 standard has been developed in Europe by integrating several consortia. Two of them are ETSI's Center for Testing and Interoperability [71] and Car-2-Car Communication Consortium (C2C-CC) [34]. Communications and information exchanges are done because of the complexity existing in all possible ITS scenarios. Different ITS sub-systems should support them, and those systems can have a wide variety of IFTs. For instance, RSU system architectures can provide helpful information regarding the current status of its surroundings or even road walk pedestrians carrying a smartphone that can be integrated into multiple scenarios. To define initial scenarios and a joint base platform, allowing communication interoperability among implementations by several manufacturers, C2C-CC has created a Basic System Profile with minimal characteristics to be deployed in all the V2V devices using the ITS-G5 standard [33]. As the technology implementation does not occur in a single step, some development and implementation conditions were defined by C2C-CC but not made standard. In this roadmap, Co-VP is expected to be deployed in phase three of day two and beyond applications [253].

ETSI EN 302 665 [73] is the base for the current ITS-G5 standard, with a selected set of options of the IEEE 802.11 model. In ITS-G5, the G5 stands for the 5 GHz frequency band. ETSI EN 302 665 defines an ITS *station* (ITS-S) as the central unit for V2V, V2I, personal (mobile personal devices), and central nodes (for traffic management and back-end systems) applications. The protocol stack and reference architecture for V2X ITS-S is shown in Figure 3.1 [95] and lists the nucleus standards of the European ITS-G5 pattern.

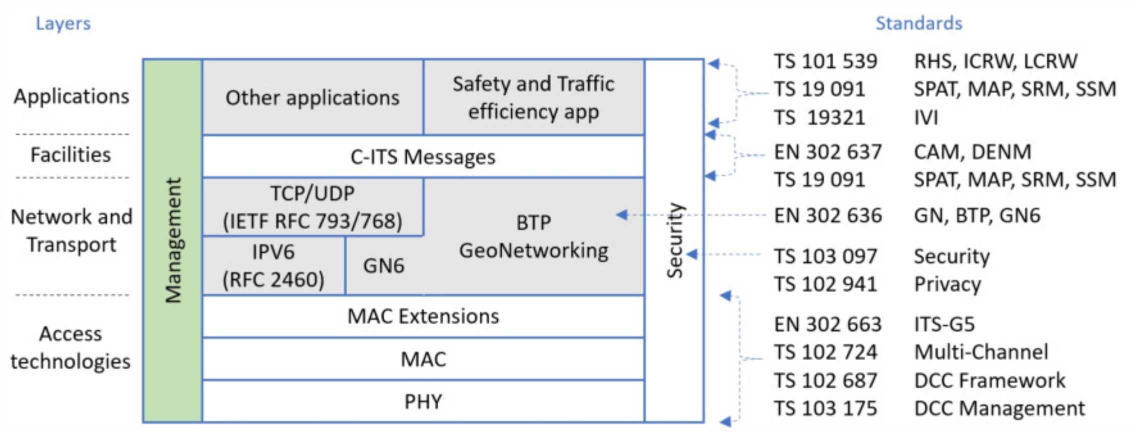


Figura 3.1: Protocol Stack for ITS-G5

Figure 3.2 presents a more in-depth overview of the expected vehicle stack architecture, focusing on V2V communications. According to the usually used as the OSI model, the OSI's Application layer can be incorporated within both the Applications and Facilities layers specified by ETSI, just like the Access layer defined by ETSI, which brings together the Physical and data link layers from the OSI model.

The ITS-S host is responsible for most of the implementation required by ETSI ITS-G5 from the application layer until the lower layers that are, as well implemented on the ITS-S *router* component. The Vehicle ITS-S *gateway* is responsible for the interface of both previously referenced modules with the Proprietary in-vehicle Network that should vary within different manufacturers. In most cases, this gateway will connect the ITS-S components to a CAN Network to get access and exchange information with the car's ECUs responsible for the vehicle actuators and gather some valuable data.

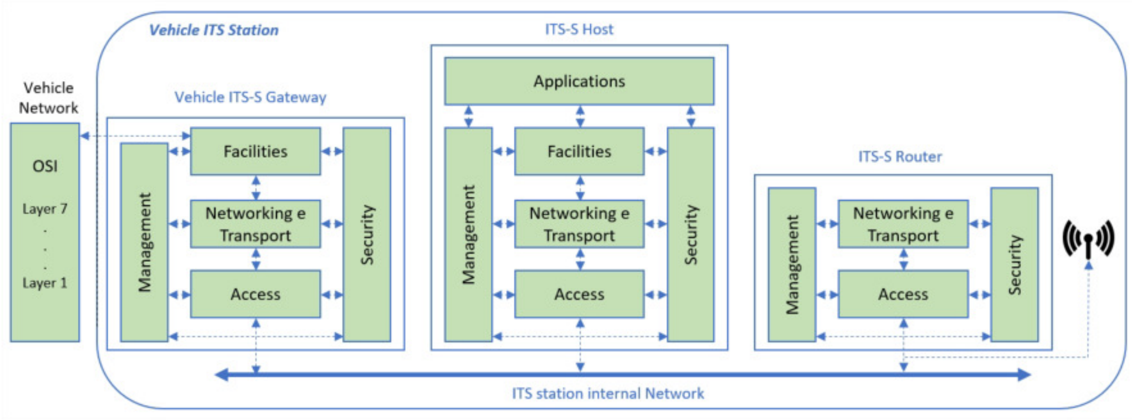


Figura 3.2: Vehicle ITS station in a vehicle sub-system [73]

3.3.1 Applications

Applications of ITS scenarios can vary in a broad spectrum. However, ITS-G5 divides them into three major groups: *Road Safety*, *Traffic Efficiency*, and *Other Applications*. Platooning can be considered a Road Safety application since the platoon members should implement safety measures to guarantee that they do not cause any harm to the normal traffic workflow.

For most of these ITS applications, specific requirements should be guaranteed by their supportive communication services. Therefore, ETSI ITS-G5 defines the following requirements as the ones to be more or less strictly imposed: Reliability, Security, Latency, and general performance parameters. Considering this, ETSI ITS-G5 defines a set by the name of Basic Set of Applications (BSA) [72], that aggregates the most relevant ITS applications to be deployed on vehicles. This setlist was created by ETSI with help from a wide variety of users and stakeholders of the automotive industry, taking into account different criteria, like strategic, economic, system performance, system capabilities, and legal requirements. The BSA from ETSI defines four main application classes: active road safety, cooperative traffic efficiency, local cooperative services, and global internet services.

3.3.2 Facilities

This layer is the one that manages all the high-level information exchanged between vehicles and other ITS stations [73]. This layer functionality from different OSI layers is present since the session layer goes through the presentation layer(e.g., ASN.1 encoding and decoding) until the application layer. Thus, this layer supports the application, information, communications, and sessions. It also provides interfaces with all the other ITS-G5 layers.

This layer also provides different facilities, like the support for joint message management for data exchange between ITS-S applications. These messages can be both Periodic or Event-triggered ones and are called: Cooperative Awareness Messages (CAMs) and Decentralized Environmental Notification Messages (DENMs).

Usually, the applications quoted in section 3.3.1 focus on four types of messages: state monitoring, control packets, services data, and safety messages [142]. Those messages and some of their main characteristics are summarized in Table 3.2. It is possible to check that the data latency can vary from milliseconds to seconds regarding the application's specification. Furthermore, the data can be triggered by motion, time, or road events, distributed differently, like broadcast, unicast, or multicast.

Tabela 3.2: Types of Messages

Message	**Latency**	**Dissemination**	**Use Cases**
State Monitor	100ms-1s	periodic, broadcast	Kinetics information, Road conditions
Control	100ms	periodic, broadcast, multicast	Cooperative Driving
Services	second level	event, unicast	News, Media
Safety	100ms	event, broadcast	Line-change, Over-taking, Collisions

CAM Messages According to ETSI, CAMs [82, 49] are sent periodically between ITS stations to all the neighbours' stations within communication range (Single-hop and Broadcast). ITS hosts use these messages to improve and update their sensing (e.g., to evaluate the distance between two vehicles), adding a redundancy layer for ITS vehicles that should feature other ways of sensing their surroundings (e.g., sensors and cameras). Data sent through CAMs is usually about the current position and status of the ITS host source.

As CAMs are sent periodically, sending time is an essential quality requirement for the Applications that might use this message to improve their services. Therefore, the CAM generation service should follow some generation rules to fulfill ETSI's requirements for the generality of ITS scenarios. These generation rules will be explored in chapter 7.

The CAM frame is shown in Figure 3.3. Both RSUs and OBUs can send CAM messages, but the set of possible CAM messages can be different, given the implemented fields of the data. The average CAM size is typically around 350 Bytes. All the RSU and OBU CAM messages need to include at least a High-Frequency Container (HFC) and, from time to time, it should also send a message with a Low-Frequency Container (LFC). The HFC contains all fast-changing (dynamic) status information of the vehicle ITS-S, such as heading or speed. The LFC contains static or slow-changing vehicle data like the status of the exterior lights. The basic CAM generation frequency is also defined in [49].

DENM Messages DENMs [78] are mainly used by the Cooperative Road Hazard Warning (RHW) application to alert road users of the detected events. The RHW application is an event-based application composed of multiple use cases. The general processing procedure of an RHW use case is as follows:

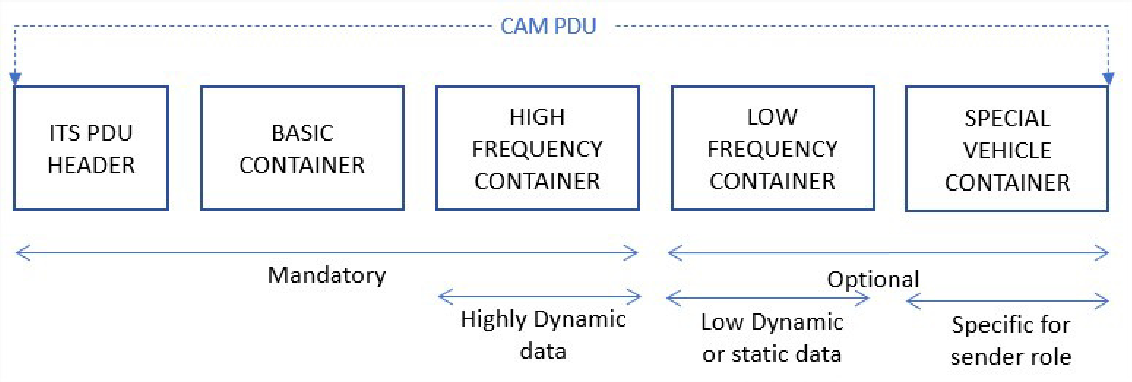


Figura 3.3: CAM Message Format

- Upon detecting an event that corresponds to an RHW use case, the ITS station immediately broadcasts a DENM to other ITS stations located inside a geographical area concerned by the event.
- The transmission of a DENM is repeated with a specific frequency.
- This DENM broadcasting persists as long as the event is present.
- The DENM broadcasting stops automatically once the event disappears after a predefined expiry time or when an ITS station generates a special DENM to inform that the event has disappeared.
- ITS stations, which receive the DENMs, process the information and decide to present appropriate warnings or information to users, as long as the information in the DENM is relevant for the ITS station.

All the data gathered by DENM messages, similarly to what happens with CAM messages, is used for the ITS station Facilities to update their LDM, using the ASN.1 representation [77].

3.3.3 Management

The management layer guarantees that all the other layers are working as intended and naturally cooperating. It belongs to the group of the only two layers (Security and Management) with interconnections with all the other ITS-G5 layers. In addition, this layer identifies available ITS services, provides a general congestion control, maps the ITS-S application on communication interfaces, maintains the information of neighboring stations, and provides regulatory information [73].

3.3.4 Security

The following are the most relevant security functionalities implemented on the ITS-G5 standard: Firewall and intrusion management; Authentication, authorization, and profile management;

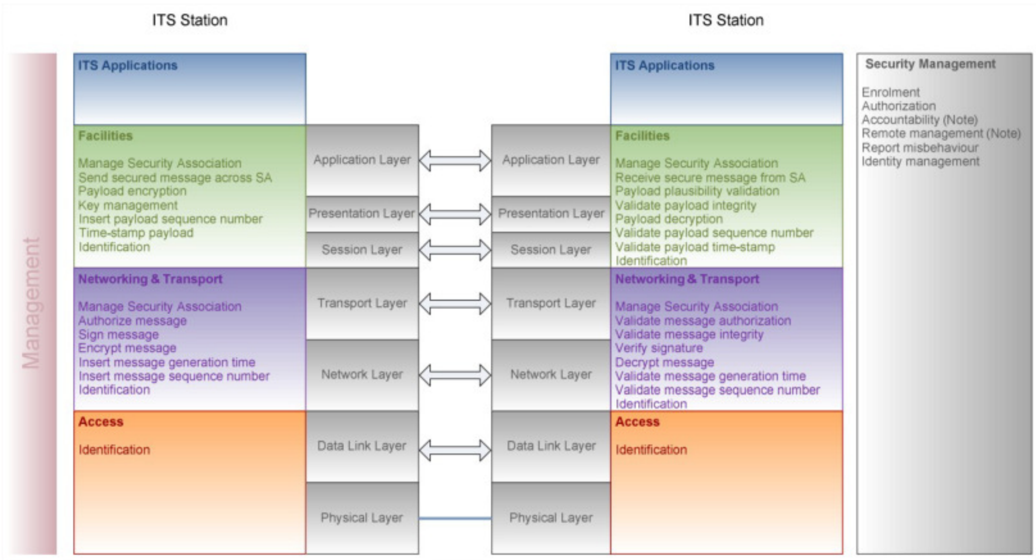


Figura 3.4: Security Services [74]

Identity, crypto key, and certificate management; A joint security information base (SIB); Hardware security modules (HSM); and Interface and security support for all the other ITS-G5 layers.

The ITS security architecture is defined in [83]. The security services are provided on a layer-by-layer basis, then each of the security services operates within one or several ITS architectural layers or a Security Management layer. The main security services, based on [74], are presented in Figure 3.4. The Technical Specification presented in [80] defines the central security standard for the security header, certificate format for asymmetric cryptography, and the private key infrastructure (PKI) enrollment and authorization management protocols. C2C-CC specifications complement PKI standards, trust assurance levels (TALs), and protection profiles (PP). The PKI standard defines the protocols among the certificate authorities and ITS stations, while TAL defines the security levels of an in-vehicle C-ITS system. The PP then comprises all measures for the security and privacy of a given TAL [95].

3.3.5 Network and Transport layer

The networking and transport layer contains functionality from the OSI network and the OSI transport layers, with some amendments dedicated to Intelligent Transport System Communications (ITSC), like Networking and Transport protocols and layer management and interface to all the other ITSC layers.

Within the different Network protocols, these are the ones that are currently supported by ITS-G5: GeoNetworking protocol, IPv6 networking[241] with IP mobility support specified in RFC 6275 [31] and optionally support for network mobility (NEMO) as defined in RFC 3963 [283] or other approaches depending on the deployment scenario. And, IPv4 support for transition into IPv6 [55].

In the case of Transport protocols that ITS-G5 supports: Basic Transport Protocol (BTP)[79], User Datagram Protocol (UDP) as defined in RFC 768 [136] and Transmission Control Protocols (TCP) as specified in RFC 793 [56]. The widely used UDP/TCP protocols are a frequent choice. However, ETSI ITS-G5 mainly uses BTP as its preferred transport protocol. For instance, BTP is used over the GeoNetworking protocol, both UDP and TCP are planned to be integrated as well, on top of this protocol [77].

GeoNetworking The ETSI GeoNetworking protocol controls the transport of data packets from a source node to the destination. The transport can be either to an individual node (GeoUnicast), all nodes, any node inside a geo-area (GeoBroadcast/GeoAnycast), or all nodes in a one-hop/n-hop neighborhood (single-hop/ topologically-scoped broadcast). Every ad hoc router has a location table that maintains the position of its known neighbors and is used to make forwarding decisions; it also has packet buffers for location service, store-carry-and-forward, and forwarding algorithms.

Two packet transport types are relevant for safety and traffic efficiency use cases: first, single-hop broadcast for the transmission of periodic CAMs, and second, Geo-Broadcast for the multi-hop distribution of event-driven messages within a geo-area, DENMs. The ETSI GeoNetworking protocol specifies three main forwarding algorithms. These algorithms distribute information in a geo-area: Simple Geo-Broadcast and contention-based forwarding as base schemes. Furthermore, advanced forwarding combines base schemes and comprises a set of protocol mechanisms to improve their performance. More forwarding algorithms and a detailed description were presented in [161]

3.3.6 Access layer

The access layer provides the means to access the medium. This layer incorporates both the PHY and Data link layer (DLL). The first one is responsible for physically connecting to the communication medium. The DLL can be sub-divided into two sub-layers, MAC and LLC. The MAC layer is responsible for managing access to the medium, while the LLC is working to provide logical link control.

This layer is defined by IEEE 802.11p [131]; this standard was created based on IEEE 802.11a to focus on and serve vehicular scenarios, with some objectives in mind: Increase the maximum distance of operation (around $1km$); provide High mobility and speed of the network nodes; provide a way to control and attenuate the Multipath effect - the existence of multiple signal echos received; and try to guarantee the best QoS, regarding the amount of different ad-hoc networks existing in these environments. Other characteristics imposed by this technology achieve these goals. For instance, the channel distribution allocated within the 5.9GHz band for the ITS-G5 standard uses a 10MHz channel spacing, as presented in Figure 3.5. Furthermore, it is crucial to notice that CAM messages, which are the main application focus of these implementations, are sent under the G5-CCH channel [161].

Due to the MAC protocol of IEEE 802.11-2012 [131], and the limited bandwidth of ITS-G5, the data load on the wireless channels, can exceed the available capacity in some situations.

Therefore, Decentralized Congestion Control (DCC) methods, as specified in [75], are required in ITS-G5 stations to control the channel load and avoid unstable behavior of the system.

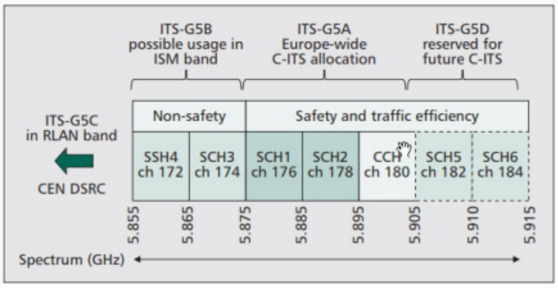


Figura 3.5: ETSI ITS-G5 Channel allocation [76]

Decentralized Congestion Control - DCC This DCC mechanism [75] is a crucial part of access layer. DCC's main goal of implementation and usage is to maintain network stability, fair resource allocation, and throughput efficiency for ITS-G5 hosts. The DCC implementation demands components on several layers of the stack protocol, and these components jointly work together to fulfill all the mandatory requirements. Important to note that DCC does not have any control over frequency channel selection on DENM or CAM. The only implementation goal of DCC is to control message delivery and guarantee a satisfactory QoS, only limiting message providing within certain time constraints and keeping the Channel Busy Rate (CBR) at a safe level.

The DCC layer dynamically adapts the frequency of data sending accordingly to the CBR. It can change MAC and Physical layer parameters like transmit power, minimum packet interval, data rate, and radiosensitivity. Each transition in the state machine has a corresponding CBR value as a threshold. Figure 3.6 shows the three basic algorithms for DCC, considering CBR thresholds [194], and CBR measurement, where the transition to a more restrictive state is performed when the CBR value was above the threshold during the last measuring interval, and the change to a less stringent state happens when the CBR value, in the previous measuring interval, was below the threshold.

The DCC application over a Co-VP scenario is over evaluation since restraining the communication in safety-critical applications, which usually are delay-sensitive, could guide to unacceptable performance degradation [296].

Medium Access Control(MAC) The MAC algorithm decides when a node can transmit based on the current channel status. The MAC schedules transmission to minimize the interference in the system to increase the packet reception probability. The MAC algorithm deployed by 802.11p is found in the IEEE 802.11-2012 [131], and it is called enhanced distributed coordination access (EDCA). It is based on the essential distributed coordination function (DCF) but adds QoS attributes. DCF is a carrier sense for multiple access with collision avoidance (CSMA/CA) algorithm. This MAC layer grants priorities to various messages such that the notes with higher priority have shorter deference in channel contentions [302].

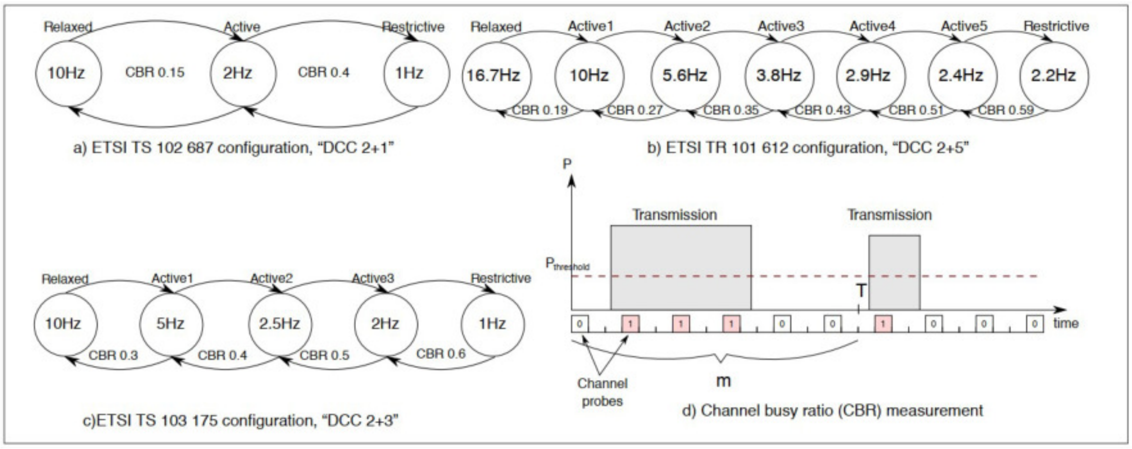


Figura 3.6: DCC state machines and CBR measurement procedure (m is the number of channel probes within the measurement period T): a) ETSI TS 102 687 configuration; b) ETSI TS 103 175 configuration; c) ETSI TR 101 612 configuration, ; d) Channel Busy Ratio (CBR) measurement [194]

In CSMA/CA, a node starts to listen to the channel before transmission, and if the track is perceived as idle for a predetermined listening period, the node can begin to transmit directly. However, if the channel becomes occupied during the listening period, the node will perform a backoff procedure, i.e., the node has to defer its access according to a randomized period. The predetermined listening period is called either arbitration interframe space (AIFS) or distributed interframe space (DIFS), depending upon the mode of operation (EDCA or DCF). In EDCA, the MAC layer defines different (AIFS, backoff) pairs. Thus, the frames with other priorities own different defer access periods. In general, the higher priority, the shorter the deferred period.

Physical Layer (PHY) The PHY in 802.11p is Orthogonal Frequency Division Multiplexing (OFDM) [231] detailed in clause 18 of 802.11 [131]. The basic idea is to divide the available frequency spectrum into narrower subchannels (subcarriers). The high-rate data stream is split into several lower-rate data streams transmitted simultaneously over many subcarriers, where each one is narrow banded. There are 52 subcarriers, where 48 are used for data, and 4 are pilot carriers. The OFDM PHY layer has support for eight different transfer rates, of which three are mandatory; 3 Mbit/s, 6 Mbit/s, and 12 Mbit/s [265].

3.3.7 Co-VP Settings

As explained in section 3.3.1, several use cases define specific settings for BSP, changing some ITS-G5 parameters, like the minimum frequency of the periodic message, maximum latency, or demanding a specific security requirement, as protection and authentication of the CAMs. The [72] defines the Co-VP as one of these use cases, called *Co-operative vehicle-highway automation system (Platoon)*. The Co-VP communication mode is classified as V2X cooperative awareness associated with unicast in this use case. They also define a maximum latency of 100*ms*, with a

CAM minimal frequency of $2Hz$ and a vehicle relative position system better than $2m$. This Co-VP profile can be called a BSP-Platoon [294].

However, taking into account the relevance of Co-VP applications, ETSI has defined a specific document with a CACC pre-standardization study [84]. ETSI defines several requirements impacting the vehicles' hardware and software to compose a platoon in this study. One of these requirements is the *CACC target time gap* (Δt_{target}): the time gap set by the Co-VP application to follow a target vehicle (TV). The Co-VP controller should adjust acceleration to maintain the time gap (Δt) with the TV as close as possible to Δt_{target}. The Co-VP Δt is the interval time when a TV's rear end and a follower's front end pass the same point of the road. This study defines the follower vehicle as the Subject Vehicle (SV). Figure 3.7 shows the Co-VP Δt, where d is the distance between TV and SV, and v_{SV} and v_{TV} represents, respectively, the SV and TV speed.

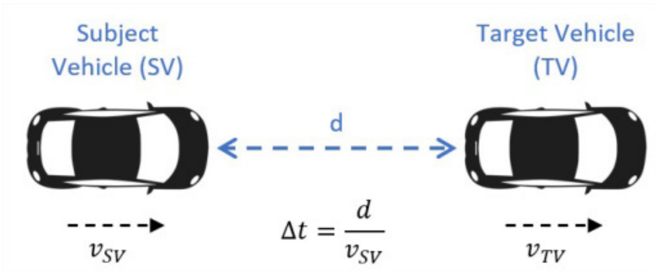


Figura 3.7: Co-VP time gap (Δt)

The Δt_{target} should be proportional to v_{SV} when it is equal to or higher than a predefined value. However, when v_{SV} is smaller than this value, a minimum distance (d_{min}) should be sustained. There is also the possibility of configuring a minimum safety time gap (Δt_{min}). The Δt_{min} should be equal or higher, then the time required for collision avoidance and can be defined as $\Delta t_{min} = \frac{|v_{SV} - v_{TV}|}{a_{SV}}$, where a_{SV} is the maximum SV deceleration.

The TR 103 299 also defines a functional architecture, as presented in Figure 3.8. The core blocs for Co-VP are:

- Message handler: manages the generation, encode/decode, reception, and transmission of C-ITS messages;
- TV identifier: identifies the TV based on data available from the message handler;
- Vehicle status monitor: monitors the vehicle kinematics;
- Environment monitor: monitors vehicle's surrounding environment;
- CACC logic manger: adjusts the CACC logic, e.g. transition between different CACC application, joining/leaving, set up CACC parameters (e.g. Δt_{target});
- Motion planner: based on CACC parameters set by the CACC logic manager, this function decides of vehicle motion and potentially vehicle trajectory or vehicle maneuvering;

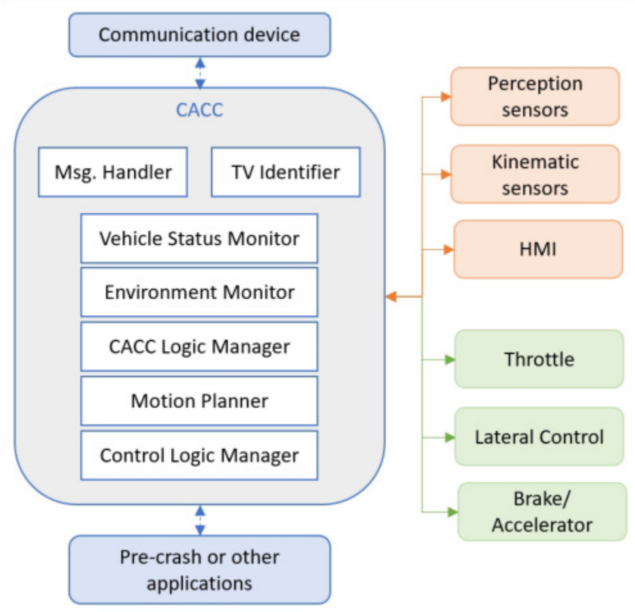


Figura 3.8: Co-VP functional architecture

- Actuator control manager: generates control command to corresponding vehicle actuators according to the motion planner results.

This study proposes a change in the CA basic service triggering conditions with two options:

- Option 1: when the platoon is engaged, the CAM trigger is set to $100ms$ ($10Hz$);
- Option 2: when the platoon is engaged, the CAM trigger should be set depending of the target distance value Δt_{target}, as demonstrated in Figure 3.9. In this figure, $T_{GenCamMin}$ is $30ms$, $T_{GenCamMax}$ is $100ms$, $\Delta t_{TargetMin}$ is $0.5s$ and $\Delta t_{TargetMax}$ is $2.0s$;

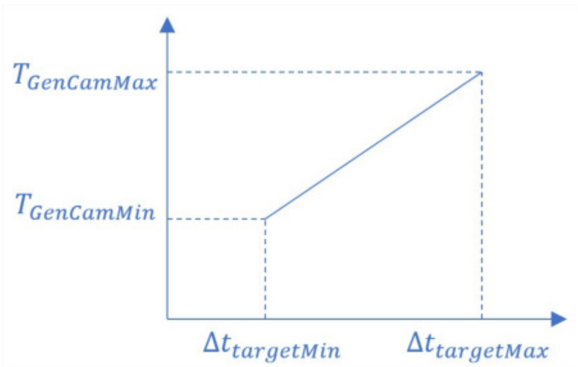


Figura 3.9: TR 103 299 CAM generation Rule

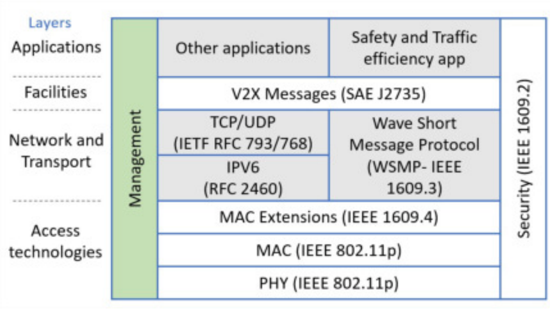


Figura 3.10: DSRC protocol Stack

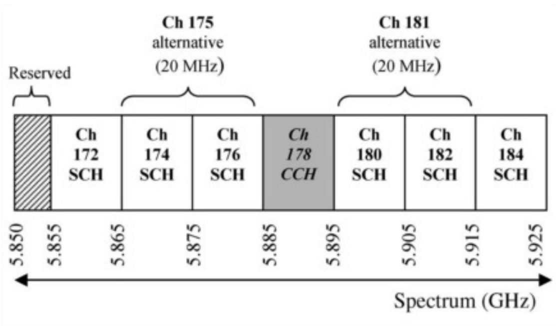


Figura 3.11: DSRC Spectrum Division [154]

3.4 Dedicated Short-Range Communication (DSRC) - WAVE

DSRC technology is the central V2X communication standard in the U.S. It is defined by a bunch of standards and protocols for automobile applications, and it has been mandatory for light vehicles in the U.S. since 2020 [282]. It is also based on IEEE 802.11p, with close technical characteristics with ETSI-G5 in PHY and Acces layer, but with some crucial differences. The DSRC protocol stack in the U.S. is presented in Figure 3.10, together with the standards in each layer, and a complete reference about the layers can be observed in [154]. The DSRC standard is IEEE Wireless Access in Vehicular Environments (WAVE).

In the U.S., DSRC ITS applications have a 75 MHz licensed spectrum in the 5.9 GHz from 5850 MHz to 5925 MHz. The spectrum and channel allocation are presented in Figure 3.11. It uses two channels, including control channels (CCHs) and service channels (SCHs). The available spectrum is divided into one CCH, and six SCHs [313]. DSRC-enabled vehicles can broadcast messages over a long distance ranging from $10m$ to $1km$, with a maximum bit rate of $54Mbps$.

DSRC adopts the IEEE 802.11p for Wireless Access for Vehicular Environment (WAVE) as its PHY and MAC layer standard, similar to ITS-G5. However, unlike ITS-G5, the direct communication between vehicles in DSRC is defined in IEEE 1609.3, with the Wave Short Message Protocol (WSMP). The WSMP is a single-hop network protocol with a small header of some bytes. WSMP provides the multiplexing of messages to upper-layer protocol entities based on service IDs, hence fulfilling the transport protocol's role.

However, to utilize the multiple wireless channels, IEEE 1609.4 standard defines a management extension to the MAC for multi-channel operation [96]. This extension allows the DSRC devices to transmit and receive messages on different channels, without a dual transceiver system, using an *alternating access* method [65]. Thus, the Control Channel can only be utilized half of the time, reducing the available bandwidth for safety messages compared with ITS-G5. The EDCA also manages the prioritization and channel access in DSRC, but with different parameters from ITS-G5.

IEEE 1609.2 standard defines the security in DSRC, providing authentication and optional encryption of the messages with digital signatures and certificates. A great comparison between the security layer of DSRC and ITS-G5 is performed in [93].

The Society of Automotive Engineers (SAE) can be observed as a counterpart in the U.S. from CC2-CC in Europe. It is responsible for several specifications, taxonomies, and definitions for CAV applications. For instance, the SAE J2945/1 and SAE J2735 [332] present the performance specifications for the V2V system that can be considered as a BSP for DSRC. In these documents, SAE also offers the Basic Safety Message (BSM), the periodical message that should be transmitted between vehicles informing about their current states, such as location, speed, or heading. Those messages are equivalent to CAM messages in ITS-G5 and vary from 1 to 10Hz.

3.5 LTE: C-V2X

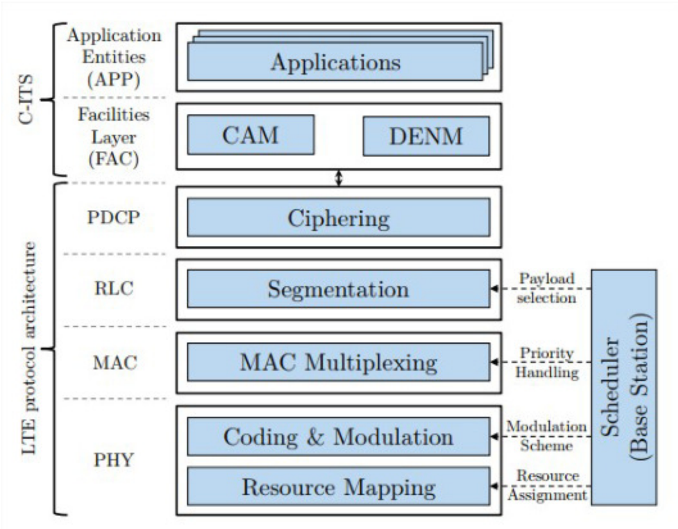


Figura 3.12: LTE-V2X Architecture [69]

Although DSRC and ITS-G5 are the dominant protocols in applications related to V2V communication to date, the use of LTE systems (cellular communication) has emerged as an exciting alternative. The Third Generation Partnership (3GPP) has defined several standards in releases 14 and 15 to support V2X communications, allowing direct communication between nodes instead of the usually centralized control method [14]. The LTE-V2X protocol with LTE architecture, based on [69] is presented in Figure 3.12.

In release 14, 3GPP defined the necessary advances in LTE-V standards to support V2X services over both Users to User (UU) interface and PC-5 interface [313]. The UU interface operates at a 2GHz licensed frequency and supports the communication between the user equipment and a centralized node, called evolved Node B (eNB). This centralized communication is defined in LTE-V2X as *LTE-V-Cell*. A decentralized communication mode, *LTE-V-Direct*, was defined using the PC5 interface [1] and allows direct communication between nodes, like pedestrians, vehicles, and RSUs. Figure 3.13 presents the usual LTE-V2X based application. The LTE-V-Direct operating frequency band is the ITS dedicated frequency of 5.9GHz, supporting road safety applications with a mesh topology.

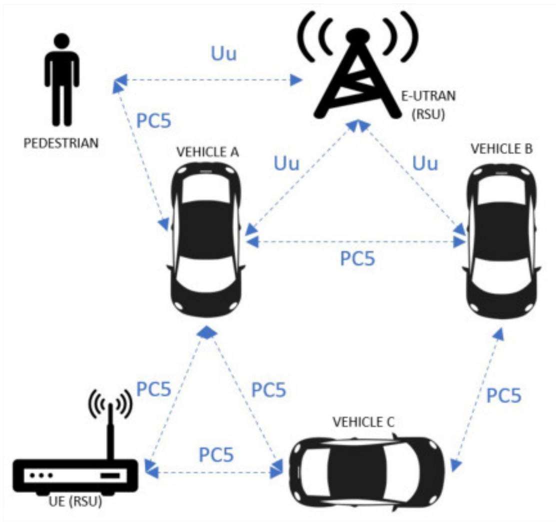


Figura 3.13: LTE-V2X usual Application

The LTE-V-Direct mode, using the device-to-device (D2D) concept [182], supports low-latency and high-reliability communication in high-speed vehicular environments. However, the LTE-V2X standards in Release 14 can only support basic V2X applications, e.g., collision avoidance, hazard warning, and emergency vehicle warning, with no support for advanced applications like Co-VP. With Release 15, 3GPP has started to work on 5G-V2X. This release involves the V2X based on fifth-generation (5G) New Radio (NR), which might use the mmWave bands to access autonomous driving. It has been expected to be able to support Co-VP applications or to be complementary to ITS-G5 standards [4].

When comparing the ITS-G5 and LTE-V2X, it is crucial to observe that the first one can be considered a mature technology, widely tested, while the last one only started to be standardized in 2015. However, LTE-V2X has some advantages that should be considered in future developments [332], like the data rate: LTE-V2X can support a high bandwidth demand since their theoretical bit rate is supposed to be $20Mb/s$ (uplink) and $80Mb/s$ (downlink), compared with a maximum of $12Mb/s$ in ITS-G5 and DSRC. One similar characteristic is the coverage area for highway environments for V2V communications, near to $300m$.

Nevertheless, ITS-G5 and DSRC have advantages over LTE-V2X, like mobility support, channel contention, and costs. Given the Doppler frequency shifts due to the longer symbol duration for LTE-V2X, the mobility support is limited to $140km/h$, while ITS-G5 supports until $250km/h$. Furthermore, in LTE-V2X, there is no mechanism to prevent collisions of messages, as the CS-MA/CA predicted in IEEE 802.11p. The PHY and MAC layer performance of LTE-V2X and ITS-G5 was compared in [200]. In this work, the authors demonstrate that, for a low-level vehicle density, the LTE-V2X outperforms ITS-G5, considering the Packet Error Rate (PER). However, while the congestion increases, the performance difference decreases until ITS-G5 overpass LTE-V2X. This result was reinforced by the study presented in [160], where ITS-G5 outperformed the LTE-V2X considering the end-to-end delay and packet inter-reception time. Finally, since the standard LTE modem does not support LTE-V2V communications, even if dedicated HW is required for both technologies, the LTE-V2X chipset is more expensive than the IEEE 802.11p-based modem of ITS-G5 [103].

3.6 Conclusion

Although vehicle communication models have evolved, their integration and application on a large scale still represent a significant challenge. Among the most critical barriers lies interoperability. Initially, we point out the different implementations of the same standards proposed by different equipment manufacturers. Therefore, the interaction between these devices must be done in the most reliable way to avoid incompatibilities and possible errors caused by subtle differences in the implementations. Furthermore, the definition of complete Co-VP scenarios is lacking even at the application layers. On the other hand, multiple standards (ITS-G5, DSRC-WAVE, and LTE C-V2V) currently contend for the market, making it challenging to apply a single global standard to reduce implementation costs. In this way, adopting a single technology may turn integration with other vehicle brands or road infrastructure impossible.

The adoption of network congestion control mechanisms significantly impacts Co-VP systems. For example, decreasing the frequency of messages between devices on the network limits the availability of information to the platoon, leading to accidents. Thus, the correct configuration of these mechanisms, especially in the ETSI ITS-G5 model, is a challenge that still presents different models to be explored.

Thus, there is no universal inter-vehicle communication model for Co-VP applications. Nevertheless, research points to the IEEE 802.11p developed base, ranging from the WAVE model in the USA to ITS-G5 in Europe. However, the strength of cellular networks, reinforced by the entry of G5, shows how this scenario is not yet definitive. Nevertheless, given the maintenance of European decisions to invest in ETSI ITS-G5 and its applicability to Co-VP models, it was decided in this thesis to use this model as the basis for CopaDrive.

Capítulo 4

Co-VP Overview

This chapter presents an extensive review of the Co-VP applications-related research. We demonstrate how they relate to reliability and safety in a multi-dimensional way. We also analyze the main validation tools for cooperative systems, addressing purely simulated scenarios and embedded in real platforms. Finally, we show the main challenges and points of advance in these areas.

4.1 Introduction

In a cooperative platoon, messages exchanged between platoon members enable collective perception that contributes to improving individual situational awareness, joint maneuverability (e.g., in terms of inter-vehicle safety distance and transverse alignment of the vehicles), or overall safety through fast dissemination of emergency notifications to the other cars of the platoon. The communications layer needs to ensure strict performance requirements regarding reliability and security. When these requirements are not fulfilled, e.g., in the presence of packet loss, transmission delay [273] or security threats [11], negative impacts can be observed in the Co-VP controller.

The study of Co-VP has advanced in several areas, such as control models for platooning [112], V2V and V2I communication [176], energy efficiency [190, 193], interaction with other vehicles and platoons [145], among others. Furthermore, the cooperative platooning applications must be simulated accurately before being implemented in the real world. Advanced simulation tools that can mimic road conditions, the operation of the vehicles' sensors and actuators and control models, and the V2X communication are necessary to evaluate the performance of such cooperative applications. In this line, formal verification methods can also play an important role in proving the safety of these systems [150].

In addition, and support of this, comprehensive toolsets are needed to analyze these interactions. Such tools aim to assess the reliability and efficiency of the developed Co-VP solutions before real-world implementation, reducing errors and costs.

Different facets of the Co-VP problem have been studied in a range of surveys, as presented in chapter 2. However, the literature suggests a gap in the relationship between the related areas, not adequately addressing the full extent of the problem, proposing separate analyzes of each of the parts, and not observing the necessary integration between them to guarantee the system's functionality. Thus, we present this chapter as a general review of the Co-VP body of knowledge, reviewing leading applications, validation systems, and main challenges, whether in safety or security.

4.2 Co-VP Reliability

Co-VP applications are subject to the most diverse interference in communication, such as Packet Loss Ratio (PLR) and delay, which directly influence stable platooning conditions, given that wireless communication quality significantly impacts the safety and performance of platoon control [175, 59]. However, although several studies analyze vehicular networks' performance, few address their impacts on Co-VPs controllers and their consequences on the system's reliability. Therefore, this section will present the recent and most relevant studies about Co-VP reliability based on different network communication threats, such as PLR, inter-message delay, and Transmission Rate Control (TRC), and the designed solutions to mitigate their influence over the platoon. These works are summarized in Tab. 4.1.

4.2.1 Network Threats

One of the first Co-VP network performance analyses was presented in [170], with a study about the impact of PLR on Co-VP's string stability performance within a CSP model. V2V communication was established based on the IEEE 802.11p standard with a fixed time message. The authors concluded that the beacon sending frequency and PLR influence the Co-VP application's performance since the string stability decreases while the messages' frequency decreases. The same network threats were observed in [226], where the authors evaluated the impacts of network properties and controller system specifications on platoon stability. Using a simple communication model and different controller parameters, they assessed the distance error between the platoon vehicles and concluded that the platoon stability decreases while the average PLR increases.

The PLR is also investigated in [58], where the authors observe its impact and the time delay over Co-VP lateral and longitudinal PID controllers. The authors analyzed two communication models, DSRC and LTE-V, with a packet loss model based on the Bernoulli distribution and a fixed time delay. The results show that the longitudinal and lateral errors increase with the growth of both the time delay and the packet loss. However, the tests were performed with a time delay or packet loss and never with both conditions. The authors presented a field test with two vehicles and LTE-V communication with fixed time delays.

The authors of [273] analyzed a 14 vehicle Co-VP application using the WAVE communication protocol, with a fixed time delay between the messages. This work introduced a deliberate

Tabela 4.1: Summary of Co-VP Network Analysis

Network Model	Cite	Year	Controller Model	Model Issues
Generical	[196]	2020	Feed Forward loop with PID controllers	Packet Loss and Time Delay
	[120]	2020	H_∞	random single packet drop
	[226]	2018	PD	Packet Loss, Time Delay and TRC
	[20]	2020	PID	Time Delay and Sensor Faults
	[111]	2019	Linear Controller	Packet Loss
	[308]	2018	Linear Controller	-
WAVE	[15]	2019	Linear Controller	Packet Loss
	[312]	2020	Linear Controller	Packet Loss and Time Delay
	[170]	2011	Linear Controller	Packet Loss
	[273]	2018	MPC	Packet Loss and Channel Crowding
ETSI ITS-G5	[59]	2017	Linear Controller	Time Delay
	[192]	2018	Linear Controller	Packet Loss, Time Delay
	[276]	2019	Linear Controller	Packet Loss, Time Delay
	[296]	2015	PID	Time Delay
	[193]	2016	Linear Controller	Packet Loss, Time Delay and TRC
	[244]	2015	Linear Controller	Time Delay and TRC
	[339]	2019	Longitudinal Vehicle Dynamic Model	TRC
	[194]	2018	Linear Controller	Packet Loss and Time Delay
LTE V2X	[58]	2018	PID	Packet Loss and Time Delay
	[320]	2018	Linear Controller	Time Delay and Throughput
	[298]	2018	Linear Controller	Time Delay

communication failure in one vehicle, and the Co-VP performance was observed after this error. They also evaluated the channel crowding, changing the default message time and demonstrating that the PDR decreases while the CBR increases.

In [196], the authors realized a numerical simulation and a HIL implementation to analyze the Co-VP controller's performance with a constant message delay and a stochastic package loss model. They proposed a feedforward controller integrated with two PID controllers to solve the model uncertainties and defined the system's string stability parameters, evaluating the platooning performance in an urban scenario and on a fuel economy test. The simulation demonstrates that the dropout rate harms string stability and following capacity accuracy when it is up to a limit. However, there were no package dropouts in the HIL implementation since there were just two OBUs. A similar approach is presented in [120], where the Co-VP stability is analyzed against random single packet drop and external disturbances. A robust LMI-based distributed H_∞ controller guarantees the vehicles' longitudinal safety distance within two different IFTs, the bidirectional predecessor-follower and a hybrid solution bidirectional predecessor-follower with leader-followers. The authors analyzed the Co-VP performance under varying packet drop rates (from 0% to 30%) with a different number of follower vehicles.

An analysis of IEEE 802.11p Co-VP communication's time-varying performance was presented in [312], in contrast with the usual steady-state communication analysis. The authors consider the impact of some disturbance in the leader's behavior and derive each follower's time-dependent estate. The authors used packet loss and the message delay to evaluate the communication performance, concluding that the IEEE 802.11p can keep the string stability under disturbance. However, this work considers a leader-followers' IFT communication topology, which reduces the number of sent messages. In [192], a similar evaluation is proposed, but using ETSI ITS-G5 standard [72] and the leader predecessor-followers IFT. This work identifies the phenomena that decrease communication performance based on message synchronization after sequential disturbances.

A Co-VP driving system is presented in [111] using a merged network and control perspective. The authors determine the upper bounds on the acceptable error due to packet losses and tune the real inter-vehicle gap, guaranteeing the platoon string stability. A Symmetric bidirectional IFT is adopted, where the subject vehicle receives data from the predecessor and the next one. They conclude that a flexible inter-vehicle distance associated with safety bounds mitigates the issues of packet losses in Co-VP applications. In [297], the authors also define a worst-case boundary for the latency of DENM in ITS-G5 scenarios to warn vehicles about emergency brakes.

4.2.2 Transmission Parameters Adjust

Some works also analyze the Co-VP communication performance regarding transmission parameters adaptation. For instance, in [193], the Co-VP fuel consumption performance is studied using two DCC configurations in ETSI ITS-G5. This work focuses on the Transmission Rate Control (TRC) adaptation, showing that this adjusts directly influences fuel consumption. Furthermore, the authors of [339] and [194] also demonstrate that the performance of the string stability controller goes down significantly when the message rate is restricted and reduced by TRC. The emergency brake in a Co-VP application was performed in [244] also using the TRC adaptation. However, the authors used fixed frequencies in this analysis, varying from 1 to 20Hz. This experiment analyzed the minimal distance between the vehicles after braking. The work presented in [276] also explores the emergency brake in a Co-VP application proposing the definition of a feasible region of communication delays.

The Co-VP network performance also depends on the definition of the message triggers. There are two standard models in literature: the time-triggered and the event-triggered strategy. Although the ETSI ITS-G5 standard already defines the event-triggered strategy as a standard, many implementations have been done with fixed time messages [244, 276, 196]. This strategy can increase platoon safety using a high message frequency ratio. However, this strategy increases the packet collision due to a crowded network [59] while the event-trigger solution reduces the network CBR.

The authors of [20] apply the event-triggered message controller in a Co-VP application with time-varying delay and sensor faults. In this study, the event triggering strategy is a function of the actual value of sensor fault instead of the standard of ITS-G5. The work presented in [308] also proposes a flexible event-triggering strategy based on tunable parameters for each platooning vehicle, reducing the communication burden. However, both studies do not address an ITS communication standard, which distances their conclusions from actual experiments.

In [296], a performance evaluation of the time delay between messages in a Co-VP application is conducted with ETSI ITS-G5. The authors compare the CAM time delay using ETSI specifications - event-triggering - against a fixed frequency of $10Hz$. In both modes, the authors consider a random transmission delay. The authors conclude that the Co-VP performance with the specified frequency outperforms the ITS-G5 standard, especially at higher speeds. However, they do not address the CBR and its effects on the platoon.

The IEEE 802.11p MAC standard, used in Co-VP networks, is based on the CSMA/CA approach. In [15], the authors identify that this policy is likely to lead to collisions and degraded performance as network load increases. The authors propose an overlay time-division multiple access (TDMA)-based MAC. In this approach, the messages are synchronized between the vehicles, reducing the collisions. This work compares two TDMA algorithms with the IEEE 802.11p MAC CSMA/CA implementation in Co-VP, varying platoon sizes.

The work presented in [298] compares the performance of IEEE 802.11p and LTE-V2V regarding high-density truck-Co-VP scenario conditions. In addition, this work presented the CAM message latency and CAM reception rate as performance metrics and concluded that long platoons could benefit from LTE-V2V due to the better link budget. A similar result was obtained in [36], which pointed to LTE-V2V as a better solution to in-coverage conditions. However, this conclusion is opposed by the works presented in [320, 148]. The authors of [320] realized a performance evaluation for LTE-V2V Co-VP communication, simulating the end-to-end throughput and delay outlines in different layers to analyze different configurations of platooning systems. This simulation demonstrated that the LTE-V2V system could not support Co-VP applications under congested scenarios, as in [148], where the authors demonstrate that the ITS-G5 outperforms LTE-V2V in cases where the LTE-V2V has concurrent data with the V2X communication.

Experience among the various works analyzed shows that Co-VP systems are highly influenced by variations caused by the network QoS. Thus, the controllers of Co-VP systems must be prepared to deal with these variations to guarantee the system's safety. However, this influence has yet to be better studied since the different vehicular network models still need to be better tested in critical situations.

4.3 Security Analysis of Co-VP applications

In addition to the problems inherent to communications, such as delays and packet losses, the Co-VP networks are subject to interference from other agents, whether they are malicious or not, affecting their operation, destabilizing the platooning [54, 252]. Such attacks can be divided into categories: information availability, integrity, authenticity, or confidentiality [8]. Some security requirements and concepts for specific Co-VP scenarios are presented in [259] and summarized in Tab. 4.2. This section presents several Co-VP cybersecurity researcher works that are summarized

Tabela 4.2: Security Requirements and Attacks

Security Requirement	Attack
Availability	Blackhole and Greyhole; Flooding; Denial of Service (DoS); Jamming; Coalition [8]; Malware; Tampering; Greedy Behaviour; Spamming;
Integrity	Falsification; Replay; Spoofing;
Confidentiality	Eavesdropping; Location Tracking;
Autenticity	Certificate replication; Sybil ; Masquerading; Tunneling; Free-Riding [188];

in Tab. 4.3. These works highlight the impact of security threats over the Co-VP application and propose strategies to mitigate them.

4.3.1 Vehicular Network Vulnerabilities

Several works focus on the physical layer security (PLS) regarding the confidentiality of the shared information. For instance, in [316], the performance of the PLS is studied over fading channel regarding data secrecy. Furthermore, the authors of [68] establish a series of challenges for PLS in vehicular communications, proposing a case study based on the coexistence of hybrid technologies. Finally, [199] applies a reconfigurable intelligent surface (RIS) in the PLS, proving that the PLS secrecy is affected by the number of RIS cells and their location. However, none of these works analyze the specific case of platooning and the consequences or applications of these techniques to Co-VP systems.

Taking into account that the most accepted standards for vehicular communication are based on IEEE 802.11p and LTE-V2X, the authors of [8, 93, 338] carry out an analysis of the vulnerabilities of these technologies to cyber-attacks. Still, only the work presented in [338] analyzes the use case of Co-VP applications using the LTE-V2X standard. In this scenario, attacks coming from an RSU, a vehicle, and two agents simultaneously are analyzed. Thus, the impacts caused on PDR, inter-vehicle distance, and speed change during attacks are studied.

On another side, an improvement for RSU's security based on an Internet of Things project called SerIOT is proposed at [106]. This solution implements a monitor for RSU, connecting them to the SerIOT Software Defined Network (SDN). In this way, as the outgoing information for the RSU is monitored, any anomalies can be detected. Furthermore, this solution also implements a honeypot to detect malicious vehicles. However, this work does not describe the impacts and the actual gain of this solution.

4.3.2 Co-VP Stability under Security Attacks

A quantitative analysis of the platooning stability under attack is performed in [11]. The authors investigate the risks of a cyber-attack on platooning stability, analyzing countermeasures and their weaknesses. Finally, the authors proposed a system to detect attacks such as message falsification and jamming, through observation of previous values or voting, based on information

Tabela 4.3: Co-VP Cybersecurity Research

Network Model	Cite	Year	Simulation	Attack Model	Attack Type	Solution
Non Co-VP	[316]	2020	Numeric	Confidentiality	Physical Layer Attack	-
	[68]	2019	Numeric	Confidentiality	Physical Layer Attack	-
	[199]	2019	Numeric	Confidentiality	Physical Layer Attack	Reconfigurable Intelligent Surfaces
	[106]	2019	-	Confidentiality, Authenticity	Falsification	SerIOT Extension
	[93]	2018	Numeric	Confidentiality	-	Public Key
	[8]	2019	-	Availability, Integrity, Confidentiality, Authenticity	Several	-
Generic Co-VP Network	[54]	2018	Numeric	Authenticity	Falsification	Gain Limit
	[259]	2019	-	Confidentiality, Authenticity	Generic	Key distribution
	[208]	2020	Numeric	Authenticity	DoS; Replay; Falsification;	Distributed attack detection
	[323]	2020	Numeric	Availability	DoS	-
Zigbee	[24]	2020	ROS + Testbed	Authenticity	Falsification	Monitoring Sensors
	[23]	2019	ROS + Testbed	Authenticity	Falsification	Monitoring Sensors
DSRC WAVE	[252]	2018	VENTOS	Availability, Confidentiality Authenticity	Falsification; Replay; DoS; Man in the Middle	Voting
	[10]	2015	VENTOS	Availability, Authenticity	Falsification; Jamming;	Voting
	[22]	2020	Numeric	Availability	DoS	Distributed Nonlinear MPC
	[281]	2016	VENTOS	Availability, Authenticity	Falsification; Replay;	Two Network Models
	[28]	2018	Plexe	Authenticity	Falsification	Proof of Location
	[222]	2018	Plexe	Availability, Confidentiality Authenticity	Spoofing; DoS Falsification; Burst Transmission;	Collaborative Controller
LTE C-V2V	[338]	2020	Plexe	Availability, Confidentiality, Authenticity	Falsification; DoS Man in the Middle	-

provided by various vehicles. In jamming detection, the procedure adopted is like detecting a degraded network with platooning output.

A Denial of Service (DoS) attack provided by jamming is studied in [22]. In this case, a restricted attack between two consecutive vehicles is performed. A secure distributed nonlinear MPC algorithm detects and mitigates the attack using local sensors and previous information, keeping the platooning stable. An alternative solution for the attack problem in Co-VP systems is proposed in [281], using the IEEE 802.11p together with the Visible Light Communication (VLC) solution. Furthermore, the authors compare the speed error in the platooning over a packet falsification and a replay attack in a scenario with just the DSRC communication and another with both communication models. Although both situations present errors, the one with both communication modules suffers less oscillation within the attacks.

In [23], a Bias injection Attack is used to cause a slowly time-varying attack signal in a predecessor-follower platoon and on a bidirectional platooning. The authors proposed an attacker-detector game based on a centralized detector that defines the best vehicles to add a sensor and detect the attack. In this scenario, the bidirectional data in the Co-VP application increase the system's security. This work is extended in [24] with scalable vehicles. Otherwise, the authors of [28] propose several reaction actions to mitigate a position falsification attack using a proof location scheme. This work also demonstrates a solution to avoid collisions by detecting false messages.

The authors of [208] presented a distributed attack detection, where each vehicle has its detection mechanism, estimating the local leader position and evaluating the received information. They also propose two recovery methods based on the state estimation of the system. The distributed Co-VP controller is also considered over an adversarial environment with the DoS attack [323], where the delay limits are estimated to determine the platooning safety.

In a more general scenario, the authors of [222] propose a distributed collaborative strategy to avoid longitudinal instability in the platoon formation under an adversarial environment. This strategy is evaluated using Plexe against Spoofing, Message Falsification, DoS, and Burst Transmission. In addition, this algorithm uses a Vote Strategy based on other vehicles' information to detect an anomaly and mitigate it.

All of the works cited allow us to observe the safety impacts of Co-VP applications. However, there is still a great deal of space to explore since the types of attacks can vary, and the control conditions can also be the most diverse. Also, it is possible to observe that there are no testbeds focused on this type of test, which means that it is difficult to evaluate the impacts of some of the proposed solutions.

4.4 Validation Tools

The complexity of Co-CPS implies the need for extensive validation capacity tools, able to test the most diverse conditions to which the devices may be subjected during their operation to synthesize the real world. Such tools make it possible to understand the safety limits of these devices and their applications, reducing costs, development time, and risks of carrying out tests with real models. Therefore, the Co-VP tests should be performed to analyze possible failures in specification, design, and implementation problems over the several project components [311]. The authors of [301] enunciate several V2X testing methods and describe their main focus and some standards. For instance, the latency and reliability of V2X should be tested with function, performance, and communication conformance testing. In addition, the application vulnerabilities and security risks can be mitigated with gateway, penetration, and accelerated testings. Finally, after the V2X validation with *in-lab* testing, the field tests should evaluate the V2X application's performance and function requirements in a natural environment.

As previously stated in Section 2.2, the Co-VP validation strategies can be divided into *simulation*, *real experimentation*, *formal verification*, and *testing*. The validation strategies allow analyzing how an algorithm works under defined performance criteria in several situations. This section will be divided between the *simulation* and *real experimentation* validation tools, with a very brief description of the other methods.

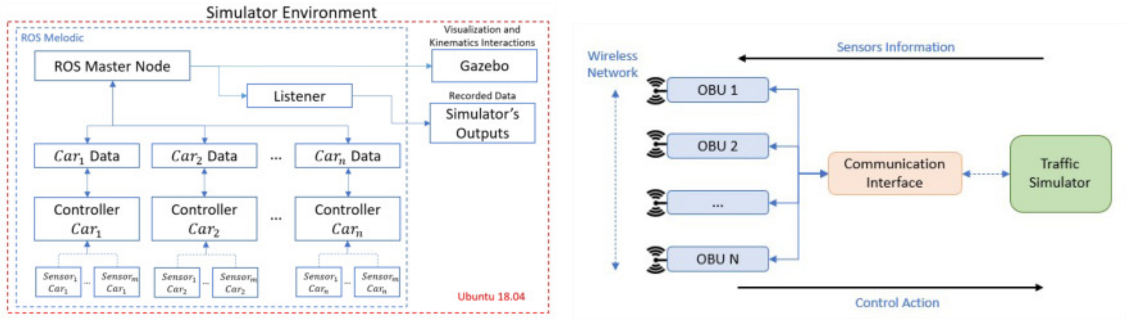
The formal foundation, used in *Formal Verification*, is an essential tool to check the system's correctness, using an accurate model description. The formal validation can be divided into model checking, theorem proving, and handwritten proofs. This validation method provides a way to avoid errors and evaluate the system behavior before the implementation, specifying the system properties to be analyzed with an agnostic approach to the scenarios. The *Testing* validation process consists of executing a system model to detect errors which can cause software failure [85].

4.4.1 Simulation

Simulative verification is a universally used validation strategy in Co-VP applications, given the costs and difficulties of implementing a large-scale environment with CAVs and infrastructures. It allows the development of a complex environment that mimics the system behavior and main external conditions. Furthermore, the simulated scenarios can be replicated and controlled, providing extensive testing fields. Nonetheless, the costs, complexity, and safety risks involved in testing with actual vehicle deployments, progressively demand realistic simulative verification tools to ease its validation, helping to bridge the gap between development and real-world deployment. Notably, such comprehensive simulation tools must be able to, as accurate as possible, mimic the real-life scenarios from the autonomous driving or control perspective and the communications perspective, as both views are highly interdependent. So, these simulations must be performed in an integrated environment between traffic mobility simulators and network simulators.

4.4.1.1 Traffic and Network Simulators

Traffic mobility simulators can be classified into macroscopic and microscopic models, considering the traffic flow granularity and the vehicle's properties. An extensive review of these simulators is presented in [223], where the most famous ones are Gazebo [218], a time-driven robotic simulator that provides support for multiple physics engines with ROS (Robot Operating System) integration, Carla [60], also a time-driven simulator specifically designed for autonomous driving research based on Unreal Engine, and SUMO [185], an event-driven platform for traffic simulation with support for a large number of vehicles and with a powerful integrating interface called TraCi. The network simulators model and test the network performance with different protocols, from the physical to the application layer. The most used ones are the discrete-event simulators NS-3 [210] and OMNET++ [215]. The first one has an implemented 802.11p MAC entity and IEEE 1609 standards, while the second one has an ITS-G5 implementation based on Artery project [232].



(a) Simulation General Architecture (b) HIL General Architecture

Figura 4.1: Experimentation Tools Architecture

Currently, several simulation frameworks converge on enabling integration between traffic mobility simulators and network simulators to support the evaluation of ITS. However, the precision in a Co-VP simulation depends on tight integration between the simulators. Figure 4.1a presents a general view of this architecture, while a summary of all the frameworks presented here is illustrated in Table 4.4. This work split the simulators into 2D and 3D simulators. The 3D simulators give more in-depth data about vehicles and Co-VP conditions, including better visualization. However, the 2D simulators are lighter systems to be handled by the computers where the simulations will be processed.

4.4.1.2 2D Co-VP Frameworks:

2D traffic simulators are the most common basis for vehicular simulations and, by extension, for Co-VP frameworks. Such systems tend to simplify the interactions between objects, generally neglecting some physical aspects, usually referring to the lateral movements of the vehicles. However, they are simulators capable of representing large-scale systems with lower computational costs, without great graphic demands. Some examples include iTETRIS [239] which integrates SUMO and NS-3, but, despite the project potential, it is finished, and there is no available support for new developments.

The VSimRTI [243] uses an ambassador concept to support the integration of virtually any simulator. Different traffic and network simulators have already been integrated, such as SUMO and PHABMACS and NS-3 and OMNeT++. The authors of [230] present a Platooning Management

Tabela 4.4: Summary of Simulation Frameworks for Co-VP

Simulator Type	Cite	Year	Framework	Traffic Simulator	Network Simulator	Network Model
2D Co-VP Framework	[13]	2019	VTI's	SUMO	OMNET++	DSRC/WAVE
	[232]	2015	Artery	SUMO	OMNET++	ITS-G5, LTE-V2V
	[233]	2019	Artery	SUMO	OMNET++	LTE-V2V
	[234]	2019	Artery	SUMO	OMNET++	ITS-G5
	[239]	2013	iTetris	SUMO	NS-3	-
	[245]	2014	Plexe	SUMO	OMNET++	DSRC/WAVE
	[143]	2016	Plexe	SUMO	OMNET++	DSRC/WAVE
	[194]	2018	Plexe	SUMO	OMNET++	ITS-G5
	[19]	2019	Plexe	SUMO	OMNET++	ITS-G5
	[185]	2018	SUMO	SUMO	-	IEEE 802.11p
	[320]	2018	SUMO + NS-3	SUMO	NS-3	LTE-V2V
	[257]	2019	Veins	SUMO	OMNET++	DSRC/WAVE
	[10]	2015	Ventos	SUMO	OMNET++	DSRC/WAVE
	[230]	2017	VSimRti	SUMO	NS-3	DSRC/WAVE
	[243]	2011	VSimRti	SUMO; PHABMACS	NS-3; OMNET++	DSRC/WAVE
3D Co-VP Framework	[60]	2017	-	Carla	-	IEEE 802.11p
	[42]	2019	-	Carla	-	IEEE 802.11p
	[86]	2020	-	Carla	-	IEEE 802.11p
	[294]	2019	Copadrive	Gazebo	Artery and OMNET++	ITS-G5
	[184]	2017	-	Webots	NS-3	DSRC/WAVE
	[150]	2017	-	Webots	-	ITS-G5
	[149]	2017	-	Webots	-	ITS-G5
Co-VP Potential Framework	[144]	2020	-	Webots and SUMO	OMNET++	DSRC/WAVE; LTE-V
	[271]	2016	Acceleration Framework	Acceleration Model	NS-3	DSRC/WAVE
	[325]	2019	QoS-CITS	-	-	TCP/UDP
	[302]	2019	Matlab	Simulink	Matlab	DSRC/WAVE

Protocol (PMP) using VSimRT, with Sumo and NS-3. This work tested the required maneuvers and proper communication behaviors with NS-3 configuration similar to ITS-G5 standards, based on IEEE 802.11p.

Another open-source framework for CAVs is Veins [257]. This framework integrates SUMO and OMNET++, coupling both simulators in a bi-directional way and performing online simulations. Veins extend the OMNeT++ with a complete communication stack based on IEEE 802.11p, creating a network node in OMNET++ for all the vehicular nodes in SUMO. The coupling between traffic and network simulation frameworks is performed using the TraCI interface. Finally, Plexe [245] is a Co-VP extension of Veins, implements protocols to support platooning applications and CACC, with several cruise control models. These control models include a longitudinal controller that uses a linear acceleration control method and a simplified transversal control (i.e., steering) to change lanes and perform Co-VP dynamics appropriately. The Plexe has also been used in [143] with a consensus-based controller for the Co-VP application regarding an intelligent traffic flow of several platoons moving together. This work focuses on the effects of poor vehicular communication, using an IEEE 802.11p stack with beaconing distribution, similar to WAVE 1609.4. In [194], the Co-VP example is used to show how even minor variations in the configuration of ITS-G5 communications may affect the performance of safety and time-critical C-ITS applications. The authors implemented the ETSI standards regarding the CAM messages with 3 DCC configurations in this work. Another work regarding DCC algorithms using Plexe is presented in [19]. This work benchmarks DCC-3 against STB (no congestion control mechanism), DynB, LIMERIC, and DCC-7, considering CBR, Inter-Reception Time (IRT), Fairness, and Safe Time Ratio.

In [13], the authors integrate Plexe with the driving simulation software from the Swedish National Road and Transport Research Institute (VTI), creating a VTI driving simulator. This work presents several use cases for CACC, like simple platooning on the road, platoon merging scenario, and a Human driver. The authors highlight significant challenges for the Plexe simulation, like adding an HDV, and there is no way to change the drive mode from cooperative to autonomous. Another issue is the lateral controller of the vehicles in SUMO. In SUMO, the lane-changing occurs instantaneously in a one-time step, not representing real-world scenarios. Another framework that integrates Sumo and OMNET++ is *Vehicular Network Open Simulator* (VENTOS) [9]. Designed for vehicular traffic flow analysis, it allows the development of new control logic, such as self-driving capability, intelligent traffic controller, dynamic routing, and collaborative driving. Using DSRC, Ventos supports V2V and V2I implementations. In [10], a PMP algorithm was tested with VENTOS, merging V2V communication with radar measurements. The WSMP carries beacon and micro-command messages on the control channel (CCH), and the resulting message is directly sent to the data-link layer, with channel access based on IEEE 1609.4. VENTOS has also been used in studies about the security of connected vehicles [281] and dynamic traffic routing [37].

Given the NS-3 high flexibility, the authors of [320] developed an integrated platform that combines SUMO and a modified version of NS-3 with the V2V transmission capability, according to LTE-V2V specification. They compared several platooning systems using network metrics, like end-to-end throughput and delay profiles. This work adjusted the frame structure, channel modeling, and performance evaluation to reproduce LTE-V2V standards.

Artery [232], provides an integration of Veins and the Vanetza ITS-G5 [235] implementation. The Vanetza provides generic ITS-G5 networking features, operating as an Application Layer to Veins. In this way, the Artery extends Veins, incorporating ETSI ITS-G5 standard protocol stack. This framework also integrates SUMO and OMNET++, but with the possibility to analyze the behavior of vehicles with different capabilities, like different sets of VANET applications. The Artery's core works on top of Vanetza, with a configurable set of VANET applications. The current Artery's version [233] has released a new model that does not extends Veins. In this model, the physical and MAC layers are provided by INET instead of Veins. This modification opened a new set of variations that allows the implementation of different stacks, like LTE-V2X, as presented in [233].

4.4.1.3 3D Co-VP Frameworks:

The need to mimic reality in a simulated environment of CO-VP systems has been rising, given the importance of increasing the likelihood between the validation performed and the actual implementation. In this context, 3D frameworks have been gaining ground in vehicle simulations, which naturally expanded their horizons for Co-VP applications. These simulators allow analyzing the details of the systems in a microscopic view, including the differences in heights, weight, and even interaction between objects. This way, the simulations become more realistic, better representing the application scenario.

For instance, a novel Co-VP simulator framework was presented in [294], where the authors carried out the integration of Gazebo with OMNET++ by extending Artery. They joined the Gazebo support for multiple physics engines with OMNET++ capabilities and Artery ITS-G5 basis to implement a microscopic simulator to represent realistic Co-VP scenarios. For instance, different from other frameworks, this one allows the analysis of the lateral controller of the vehicles regarding heading and steering angles. Furthermore, the integration between Gazebo and OMNET is provided by Robot Operating System (ROS) through a publish/subscribe method using topics. Moreover, this framework was extended with an Advanced Driving Assistance System (ADAS) using the DSME communication stack to create an intra-vehicular network [164].

Other robot simulators, like Carla, can support this kind of simulation. Although the Carla simulator has a high engine power and ROS integration possibility, to the best of our knowledge, there is no literature about this traffic simulator as a part of a Co-VP framework integrated with a network simulator. In [42], and [86], the Carla simulator was used to study, respectively, the Co-VP overtaking behavior in a two-lane highway and to implement a decentralized novel model-free controller for platooning. However, the communication was not simulated using a realistic network protocol in both works.

Another prominent robot simulator is Webots [52]. The Webots was originally designed as a research tool to investigate mobile robots' control algorithms, and since 2018, it has become an Open-source project. Several microscopic vehicular models and possible integration tools have been used within a large physical background. This simulator has been used in some Co-VP frameworks, allowing multiple studies. For instance, in [184], the authors integrate Webots and NS-3 to demonstrate the capabilities of the simulation tool, using an ideal and realistic communication channel, but with no proper stack model. In [149], and [150], the Co-VP performance is evaluated under several conditions, revealing its weakness in real scenarios, like normal and degraded network models, speed changing, and full brake. Given the vehicular model, the Webots allow the analysis of longitudinal and lateral controller models. Nevertheless, there is no integration with a network simulator in those works.

4.4.1.4 Potential Co-VP Simulators:

In addition to the frameworks presented so far, other validation frameworks for vehicle control integrated into communication systems have great potential for validating Co-VP systems. In this subsection, we analyze some of these applications that may be modified, adding functionality to the validation of Co-VP scenarios.

In [144], the authors designed a framework that integrates Webots, SUMO, and OMNET++, using a client/server model. The SUMO is the Server in this application, providing traffic demand and representing a 2D system. The Webots allow a 3D visualization and provide the CAVs control, while the OMNET++ provides the V2X structure - using 802.11p or LTE-V communication. This implementation introduces multiple human driving simulators in the CAVs scenario. Nevertheless, this framework is not yet prepared for Co-VP scenarios.

The authors of [271] developed a potential Co-VP simulator called *Acceleration Framework* that consists of a self-built microscopic traffic simulator integrated with NS-3. This traffic simulator contains an *acceleration model* that recognizes different approaches for regular, connected, and autonomous vehicles and a *Lane-Changing Model* that captures the effects of additional information on lane-changing behavior in a connected driving environment, using a game-theoretical approach. In addition, this simulator allows for V2I and V2V analysis. However, this framework has not yet been addressed to Co-VP-specific applications.

The framework was developed in [325], called *QoS-CITS*, is oriented to Quality of Service (QoS) analysis in CAVs, like throughput, safety, and fuel consumption. This simulator also analyzes how long one vehicle is delayed when it travels along its planned trajectory and how the neighboring cars could impact the desired path plan. The CAVs communication is provided by the X2X Sim module, which simulates TCP/UDP protocols, affecting wireless communication. Although this framework has the potential to be used in Co-VP validation, it does not implement any V2X standard communication.

The Matlab/Simulink also powerup up several Co-VP analyses and simulators. In [302], the authors designed a bit-accurate simulation environment for vehicular networks using the MATLAB discrete event system (DES). In this work, the authors use an *integrated simulator*, containing both traffic and network simulator. The developed network simulator includes a DSRC/WAVE implementation with a precise representation of the PHY layer compared to the NS-3 implementation. The authors have a better computational cost within this simulator in terms of events and show a more realistic packet success rate (PSR) than NS-3. This simulator has not been tested in Co-VP applications, and its vehicle movement model is still restricted.

4.4.2 Experimentation Tools

Simulator frameworks are essential in validating Co-VP systems, given the flexibility, scalability, and reduced cost. However, testing on real platforms is the most natural step in development since simulators, no matter how accurate they are, cannot encompass the real-world dynamics and imperfections produced by the process characteristics and constraints. Nevertheless, given the costs and complexity of Co-CPS and Co-VP applications, their large-scale implementation over accurate models is quite complicated. This way, the real validation tests can be divided into two stages to reduce costs and allow the analysis of each system component in a modularized way. These two steps are defined as Hardware in the loop (HIL) simulations and Testbeds. This section will introduce these tools, dividing HILs into those with or do not have Co-VP implementations. At the same time, the Testbeds are separated into potential Co-VP, Co-VP implementations with non-standard vehicular network communication, and the ones with some ITS network. A summary of all the quoted HIL implementations is presented in Tab. 4.5, while the Robotic Testbeds are presented in Tab. 4.6.

4.4.2.1 Co-VP HIL Potential Implementations

HIL testing is frequently used in the car manufacturing process. It provides a well-defined environment for the device under test (DUT), typically used for testing complex physical systems and processes. Compared to real field tests, it is less expensive, and also, the results are much easier to replicate [211]. The HIL approach also allows the experimentation and analysis of a specific component in the Co-VP study, like the OBUs, RSUs, or the real-time vehicle response. The general HIL architecture for Co-VP scenarios is presented in Figure 4.1b, where we have bidirectional information flow between the physical and virtual subsystems. In this architecture, HIL flexibility allows physical test vehicles to interact with virtual vehicles from traffic simulation models, increasing validation scalability and reducing costs [197]. Another advantage of HIL testing is evaluating safety-critical systems and features that generally operate in highly variable environments in a controlled and limited environment. It also allows the parallel development of different system components on time [147].

For instance, the PaTAVTT is a HIL testing platform that performs a trajectory tracking of CAVs [317]. The authors validated the algorithm model and control strategies in Carsim/Simulink and then migrated it to the HIL platform. In the HIL platform, the vehicles communicate with the central network node using the 802.11ac (Wi-Fi) standard and evaluate the performance of several *U-turn* movements. This platform allows the implementation of Co-VP applications, like following, lane changing, and overtaking, but none of them is presented in this work. The Wi-Fi communication network limits the comparison with other Co-VP scenarios, given the differences to IEEE 802.11p, for instance.

The HIL testbed presented in [248] emulates the fuel consumption performance of an actual vehicle with about 1% error, using the VISSIM simulator allowing the performance evaluation of a CAV that follows an on-road testing vehicle driven on real-world roadway circumstances. The V2V and V2I communication is performed using the DSRC/WAVE stack, while the communication between the cars and the VISSIM uses Cellular Network. The VISSIM was also used in [197] to emulate the traffic with a simulated DSRC/WAVE communication and a physical vehicle. This HIL configuration allows the test of vehicle connectivity and automation functions under different virtually created special conditions and evaluates crucial hardware and software components of CAV vehicle platforms. Furthermore, DSRC latency and packet loss are included in the simulation to estimate real-time communication.

The VENTOS framework is extended in [9] with a HIL support (VENTOS-HIL). In this implementation, real OBUs/RSUs are connected to VENTOS. For each physical device, there is a corresponding virtual OBU or RSU, allowing that all the actions on the physical devices reflect on

Tabela 4.5: Summary of CACC HIL Platforms

Co-VP	Cite	Year	Framework	Controller Model	Network Standard	Objective
No	[45]	2017	-	-	ITS-G5	DCC models
	[317]	2017	PaTAVTT	Longitudinal and Lateral	802.11ac	U-Turn
	[211]	2018	OMNET++ and Artery	-	ITS-G5	Network Performance
	[197]	2018	CACC HIL	Simple Mobility	DSRC	Traffic Simulator
	[269]	2018	-	Stop-and-go Re-routing Process	ITS-G5	V2V and V2I test
	[9]	2019	VENTOS-HIL	Longitudinal	DSRC	Emergency Brake
	[248]	2019	Vissim	Longitudinal	DSRC	Fuel consumption and emissions control
Yes	[284]	2014	LabView	Longitudinal Platoon Maneuvres	3G cellular network	Co-VP Controller
	[226]	2018	-	Longitudinal	Abstract network model	Longitudinal String stability
	[339]	2019	PCA Evaluation Framework	Longitudinal	ITS-G5	TRC impact
	[196]	2020	OSU-ADL-CAV HIL	Longitudinal	DSRC	Distributed Co-VP Controller
	[328]	2020	Carla	Longitudinal	LTE V2X	Parallel communication framework
	[293]	2019	Copadrive	Longitudinal and Lateral	ITS-G5	RTM and CLW evaluation

the simulation itself and vice versa. The adopted network standard is the DSRC/WAVE. The HIL capabilities were analyzed using an emergency brake scenario, showing the extended simulator capabilities.

In [269], a HIL Simulation Framework for evaluation and fast prototype of CAV's applications was developed using SUMO as the traffic simulator. The authors claim that the implemented HIL structure is cost-efficient and easily configurable to allow several CAV tests. In this work, an *Orchestrator* was designed to be a systems manager, integrating the SUMO with the attached real HW-based OBU/RSU devices. Each one of these devices has an *gpsfake* instance, producing essential location data required by the HW/SW V2X protocol stack. The communication between the Orchestrator and the remaining modules uses TCP sockets, while the V2X communication uses the ITS-G5 standard. The user can create several scenarios in this HIL implementation, but they are limited to SUMO and TraCI capabilities. The HIL capabilities were evaluated in a system where V2X communication should support dynamic re-routing of a vehicle in a congested traffic area.

The HIL approach can also evaluate specific network conditions or components. For instance, in [45], experimental validation of ETSI DCC models was proposed. This work studied the unfairness and oscillation issues of DCC implementation and analyzed the process stability of the DCC mechanism under different network conditions and CAM parameters. However, although an actual OBU device was used, there where no mobility in the simulation. In [211], a reactive HIL simulation was implemented with a simulated scenario using OMNeT++ and the real-time 802.11p Over the Air (OTA) proxy. The V2V evaluation tests perform the communication analysis of one simulated *physical twin*, that represents the vehicle able to distribute the received messages to/from surrounding simulated vehicles.

4.4.2.2 Co-VP HIL Implementations

The HIL flexibility allows different Co-VP evaluation analyses. For instance, the authors of [284] presented a Co-VP PMP strategy validation using HIL, built over a decentralized controller model, where each vehicle in the simulation has an OBU, collects the primary data, and forwards them to a central manager that stores and reorganizes the cars. These OBUs communicate through the 3G cellular network, suffering from several delays caused by the centralized communication strategy.

The work presented in [226] enables the Co-VP evaluation performance based on stability and risk-of-collision analysis. Extensive simulation using real-world vehicle parameters can examine longitudinal controllers' specifications and network characteristics, allowing the observation of platooning performance boundaries caused by network constraints and control system definitions. However, the network communication model was assumed as abstract and straightforward, with no communication protocol stack, using the IEEE 802.15.4 standard in the 2.4 GHz band. At the same time, the vehicle dynamics are simulated with Matlab. Extending the network constraints analyses, the impact of Transmission Rate Control (TRC) over a Co-VP scenario based on industrial V2X nodes operating in the ITS-G5 channels is the main focus of [339]. It evaluates simulated vehicles' longitudinal distance in congested scenarios, changing the message's frequency, based on a four OBUs vehicle simulation with data logging over the Matlab Software.

The authors of [196] implemented a HIL test platform using the Carsim/Simulink vehicle simulator integrated with real DSRC modems. This HIL enabled a realistic evaluation of a Co-VP model's parameter selection method based on a feedforward controller within a stable string boundary. Furthermore, this platform also evaluates the impact of dropout and communication time delay in the Co-VP longitudinal string stability.

An LTE C-V2X[69] HIL implementation was presented in [328]. Although this work is still under development, the authors presented an exciting platform, based on the CARLA simulator, integrated with SUMO and direct communication between the simulated vehicles through C-V2X Mode 4 modules. This platform implements a Software-Defined-Radio (SDR) based on three radio devices that mimic three cars. Several Co-VP controller models can be evaluated in future HIL implementation developments based on SUMO implementation.

An extension of the Copadrive simulator to a HIL platform was presented in [293]. In this work, the authors integrate the Gazebo with ITS-G5 OBUs to evaluate the impact of several message frequencies in the Co-VP controller. This HIL implementation also introduces critical safety tools evaluations, using a Run Time monitor (RTM) and a Control Loss Warning (CLW) to increase the Co-VP safety. Finally, this implementation allows the lateral vehicle controller evaluation and the longitudinal analysis.

4.4.2.3 Potential Co-VP Robotic Testbeds

Although HIL implementation's value is indisputable and can support a significant portion of the system's development, there are still limitations, like vehicle components evaluations. In the middle-ground between simulation-based approaches and full vehicle deployments, robotic testbeds appear as a great solution, considering that they can integrate with different platforms that are to be deployed in vehicles. Furthermore, these testbeds can also be implemented in controlled environments and partially replicate a realistic scenario at a small part of the cost of an actual vehicle [97].

Several robotic testbeds allow autonomous vehicles' evaluation but are not yet ready for Co-VP analysis. For instance, the authors of [292] developed a low-cost testbed that can be implemented in different vehicle models to test different control algorithms to follow trajectories autonomously. However, the proposed testbed does not support V2X communications and relies on onboard sensors to implement a platooning test. Two other low-cost testbed for platooning with no V2V communication are presented in [89] and in [139]. The last one relies on the HoTDeC hovercraft, increasing the flexibility but using a pretty different vehicle dynamics from a classic car.

4.4.2.4 Co-VP Robotic Testbeds with General Network Communication

There are different equipment combinations for implementing a testbed aimed at Co-VP applications. Thus, some use communications that do not follow a vehicular pattern, often focusing on control issues or validating specific algorithms. For instance, the testbed developed in [240] allows the Co-VP analysis using vehicles on a scale of 1:10 for passenger cars and 1:14 for trucks. This testbed enables implementing different control strategies for several autonomous driving applications and even platooning. In addition, a CACC controller with a predecessor-follower IFT was implemented in the Co-VP experiment, using UDP messages over a WiFi standard. This testbed allows longitudinal and lateral platoon control.

Researchers of Arizona State University [187] developed the vehicular cloud robots (VC-bots) testbed, aiming to enable an open platform for both research experiments and education services on VANET, vehicular cloud computing infrastructures, and future intelligent vehicles applications. The vehicles are set up from different robotic platforms to simulate other cars' models in this work. This platform is flexible, enabling the development of different cooperative platooning strategies [186]. However, V2V communication is based on WiFi networking. Therefore, this

Tabela 4.6: Summary of Platooning Robotic Testbeds

Co-VP Implementation	Cite	Year	Framework	Implementation	Network Standard
Potential Application	[292]	2018	Autonomous Car	Lidar System	-
	[139]	2017	HoTDeC Hovercrafts	Vision System	-
	[89]	2019	Cheap Controller Unitis	Lidar System	-
Generic Networks	[240]	2019	Small scale vehicles	CACC with PF	
	[187]	2016	VC-Bots	PF model with platoon maneuvres	WIFi
	[186]	2017	VC-Bots	PF model with platoon maneuvres	WIFi
	[303]	2018	Zynq/SoC	FPGA Based with Cooperative Sensing and Information Interaction	Zigbee modules
	[173]	2020	WiFiBot ARV	PF	IEEE 812.15.4
	[53]	2020	Remotely Accessible Cars	PF with variable controller models	Centralized WiFi
	[32]	2017	Autonomous Car	ACC	5G emulation
ITS Networks	[137]	2020	Low cost testbed	LF, PF and LPF	Simplified DSRC
	[87]	2020	Carma	Cooperative Driving Automation (CDA) with ROS Topics	DSRC
	[307]	2019	Drive-by-Wire electric vehicle	Linear feedforward longitudinal controller and MPC lateral controller	DSRC
	[59]	2017	Toyota Prius III Executive	Event triggered	ETSI ITS-G5
	[97]	2020	RoboCoPlat	PF with PID	ETSI ITS-G5

project architecture separates control systems for the longitudinal and lateral control, using V2V contact for the longitudinal controller and a camera vision algorithm for the lateral controller.

Some testbeds rely on non-usual communication standards to implement Co-VP applications. For example, in [303], a Zigbee communication module is implemented in each vehicle to provide V2V communication. This testbed uses an FPGA as the vehicle's mainboard and applies cooperative sensing and information interaction between the cars to control the platoon stability. In addition, the ZigBee module on the leader-vehicle works as a coordinator node to supervise the whole network.

The cybersecurity in Co-CPS also can be analyzed through Co-VP testbeds, as presented in [173]. This work built a platooning testbed with WIFIBOT autonomous robotic vehicles, using the IEEE 812.15.4 standard for V2V communication. This work's primary focus is introducing a cooperative secret key agreement, called CoopKey, a scheme for encrypting/decrypting the control messages. The algorithm's efficiency is evaluated regarding the longitudinal distance of the platooning members.

A cyber-physical testbed for wireless networked control systems is presented in [53]. The author proposes a testbed composed of *Remotely Accessible Cars (RAC)* that uses a Wireless LAN as a communication link. Each vehicle has several sensors to detect a line in the track and send it to a central node. This node analyses the vehicle's data, like position and speed, and sends back the vehicles' commands to guarantee Co-VP stability. This testbed evaluates three controllers with a predecessor-follower IFT: PID, Linear Quadratic, and MPC. However, the centralized approach does not attend to the requisites for a high-demand Co-VP network application with a high delay.

In [32], the authors developed a system that uses 5G ultra-reliable and low-latency communications (uRLLC) emulation for deploying vehicular cooperative demands. This project objective was to design a V2X communication platform that enables flexible reconfiguration within a small frame structure, rapid real-time processing, and flexible synchronization. This system was integrated into an autonomous vehicle to test cooperative driving scenarios, such as semi-simultaneous emergency brakes. However, this testbed has some limits, as it targets the communications platform and is not clear about the potential impacts on a cooperative controller.

4.4.2.5 Co-VP Robotic Testbeds with ITS Network Communication:

A low-cost robotic testbed is presented in [137]. This testbed presents a flexible IFT focusing on evaluating a Co-VP emergency brake situation under communication losses. The Co-VP analysis can be performed with a Leader-Follower, Predecessor-Follower, and Leader Predecessor-Follower IFTs and different communication loss parameters, using TDMA communication over a simplified DSRC standard. As the vehicles have a low cost, the testbed can be easily escalated, respecting the radios' communication ranges for different IFT conditions.

An important robot testbed for testing Co-VP applications has been developed in the CARMA [87] project. The CARMA project is an initiative led by the Federal Highway Administration (FHWA) to enable Cooperative Driving Automation (CDA) research and development. This initiative includes cloud-based transportation systems and a vehicle-based platform for automated vehicles to share data and intent with other cars and infrastructure to enable cooperative actions. The CARMA evaluation tools include an open-source simulation environment built on CARMA and SUMO and developing a scaled testbed with hardware for autonomous driving. The CARMA controller model uses the ROS as the main publish and subscribe method to integrate most project components, with the DSRC standard as the communication protocol. With string stability and platoon maneuvres, the Co-VP implementation was planned for October of 2021.

The separation between the longitudinal and lateral control was also used in the testbed control implemented in [307]. The Co-VP controller does not rely on a high-accuracy positioning system or V2I information in this testbed. Nevertheless, the Co-VP controller is based on V2V communication and a low-cost onboard millimeter-wave radar sensor. The preceding vehicle's information, like acceleration, heading, and yaw rate, is sent to the following vehicle through wireless communication, while the radar calculates the inter-vehicle distance and velocity difference. The testbed model used a drive-by-wire electric car, using OBUs with DSRC standards to implement V2V communication.

Just a few testbeds in the literature have already implemented the ITS-G5 standard. For instance, the testbed presented in [59] includes a three vehicles platoon to validate an event-triggered control scheme and communication strategy experimentally to guarantee L2 Co-VP string stability. The testbed vehicle model is the Toyota Prius III Executive, equipped with ETSI ITS-G5 OBUs for V2V communication. However, as this testbed uses real vehicles, the scalability is significantly compromised, restricting the possible tests and reducing the system flexibility. In addition, the authors only investigate the system's time response for vehicles' longitudinal speed and acceleration in this work, avoiding the vehicle lateral controller analysis. To increase the scalability, the work presented in [97] and in chapter 5 introduce a 1:10 testbed called RoboCoplat. This testbed is an extension of CopaDrive, allowing the analysis of the simulated Co-VP algorithms in a realistic platform, using embedded OBUs to communicate though ITS-G5.

Therefore, there is clearly no single solution to support the development, test, and validation of Co-VP systems, as each presents its clear advantages and limitations. Thus, the best approach relies on using several tests and validation tools for each stage of the development process and different technologies. However, the lack of integration between other platforms significantly increases the development time of the Co-CPS system. It is not desirable that the effort involved in integrating the system components with the validation platform is repeatedly discarded due to the significant differences between test environments and the prototype system.

4.5 Open Challenges

Although several prospective work ideas were drawn throughout the chapter, we summarize the open research avenues for safe and reliable Co-VP applications in this section.

Controller Models The control methods of Co-VP systems are the most diverse, although the choices for simplified PIDs, variations of the MPC system, and some robust control applications predominate. The choice of the controller directly impacts the complexity of the system to be implemented and the consequent possibilities of errors, mainly resulting from the response time. Some solutions proposed in theory do not address how to deal with the errors inherent to the processes, whether these arise from network delays or even from the natural uncertainties of mechanical systems, significantly limiting their actual implementation.

Another challenge in the Co-VP controller methods is managing mixed traffic, including HDV, ACC, and Co-VP. Furthermore, the interaction between different agents is crucial in Co-VP implementation since the vehicles should safely interact with the environment. This interaction directly affects the Co-VP controller models that still have to find the best balance between responsiveness and complexity while guaranteeing the system's safety.

Co-VP Network Threats The configuration of network parameters dramatically impacts the ability to execute the planned activities safely. Thus, testing the limits of these communication models, considering the implemented controller systems, is challenging since traditional validation tools are not realistic. In addition, the scalability of these analyses is also a challenge for implementing Co-VP strategies in the real world because the number of critical scenarios and possible failures is large. Therefore, these scenarios must be analyzed, minimizing the agent's risks.

On the other hand, the increase in the computational capacity of embedded systems also opens up new possibilities for implementing techniques and models. For example, with the latest devices, it becomes possible to implement new, more complex control techniques or even artificial intelligence models that reduce errors and can prove more reliable under critical scenarios.

Cybersecurity in Co-VP The amount of work on cybersecurity for Co-VP systems is still lacking, as most of the research in the area refers specifically to cybersecurity problems in vehicular systems. Thus, the analysis of the impacts of security failures on the performance of Co-VP systems is still a field with room to be addressed. As important as analyzing the effects, the challenge is still posed by implementing countermeasures against possible cyberattacks. Since such attacks can generate catastrophic consequences, response systems must be as efficient and quick to act, preserving the system's safety. Again, the difficulty in adopting a single communication network standard has a negative impact, as different protocols have different security implementations, which may or may not be certified. Thus, such performances open the possibility of attacks and the consequent need for quick detection and countermeasures.

Validation Tools To increase the safety of these systems, it is fundamental to validate them in realistic scenarios. Simulation represents an important step in exploring limit scenarios, mainly if the simulation system comprises Hardware in the Loop component test. However, nothing replaces final system validation over current conditions, allowing the evaluation of communication modules and mechanical systems responsiveness. Hence, to reduce validation time and costs, such tools should, as much as possible, consist of complete validation frameworks that cover, in an integrated fashion, steps from development, to simulation, robotic testbeds, up to final system deployment. Relying on open automated development frameworks such as ROS to combine the different validation stages seems a promising solution, increasing the system's modularity and enabling component re-use and continuous integration and validation.

New tools have emerged, enabling the development of Co-VP application simulations with more excellent proximity to reality, which increases their ability to represent real problems. However, the current difficulty of integrating these simulators with different communication standards and their high computational cost reduces their efficiency, invalidating more complex scenarios.

4.6 Conclusions

The interest in implementing Co-VP systems has brought significant investment to this area, both in academia and the automotive industry, since it can enable attractive solutions to reduce traffic, energy consumption, and road accidents. Thus, several aspects of these systems have been studied from different perspectives and communities. In this work, we seek to synthesize the most recent advances, show current research, and point out challenges and fields that are still open for new developments, which can guide further research.

Capítulo 5

CopaDrive: an integrated ROS cooperative driving test and validation framework

In this chapter, we introduce CopaDrive. A ROS-based Co-VP simulation framework integrating vehicle simulation environment, Hardware in The Loop, and Robotic testbed. We will present a detailed description of its features, architecture, and a use case for its functionalities.

5.1 Introduction

This chapter presents CopaDrive, a cooperative driving framework that relies on ROS as an enabler and integrator to support the development and test of Co-VP systems. We leverage ROS's remarkable flexibility and publish/subscribe topics structure to produce a continuous and integrated Co-CPS development system. Containing a vast selection of "off-the-shelf" software packages for hardware, ROS allows the integration of different projects into a powerful, flexible, and modular framework. Those characteristics extended from ROS to CopaDrive grant an adaptable framework that can be used in several stages of the development of CPS, reducing time and increasing integration between the development steps. ROS high stability and reliability [7] also are characteristics that reinforce the choice of ROS as the central component of the framework (Figure 5.1).

Supported by ROS, CopaDrive integrates a physical simulator (Gazebo) with a traffic generator (Sumo) and a network simulator (OMNET++) to analyze a cooperative driving system and evaluate the impact of the network behavior upon the platoon and vice-versa. An OBU can easily replace the network simulator to create a HIL simulation and test the communications platforms and safety components in a virtual environment. ROS integration makes it possible, allowing the

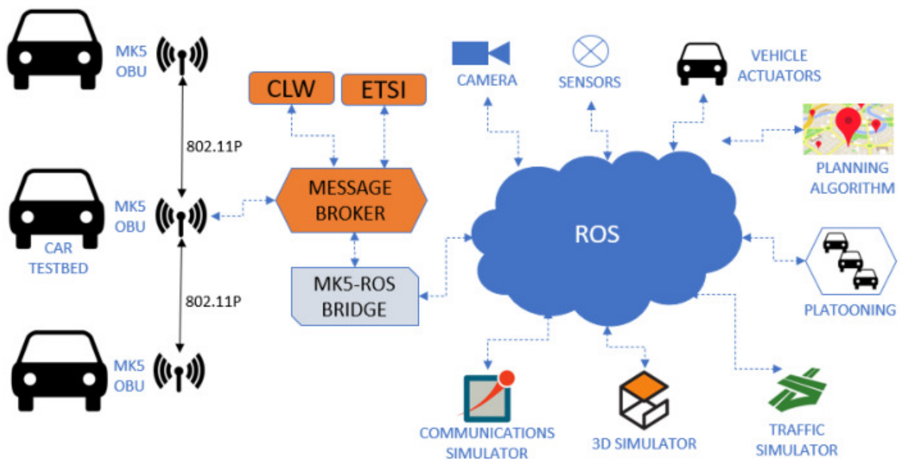


Figura 5.1: CopaDrive Main Architecture View

communication of the simulator with the OBU, keeping the system's controller modularity. Finally, the developed techniques can be integrated into a unified robotic testbed over the computing platforms deployed in the final prototype. The CopaDrive testbed is built over the real onboard computing platforms to demonstrate the Co-CPS system and test the main components used in an actual vehicle.

A development and validation framework that offers straightforward integration with simulation/HIL tools and a robotic testbed allows for developing a more efficient cooperative driving system. It also enables a complete and overarching analysis of its safety limits, particularly the study of different control strategies, network behavior impact, and system scalability. ROS as the central messenger system reduces the integration time and increases modularity and composability. The reuse of the same software components with minor adjustments from each stage of Co-VP development over a series of scenarios increases the system's reliability.

The concrete objectives of the CopaDrive framework are to:

1. *Offer a flexible environment for integrated development, test, and validation, of cooperative driving applications*. Using ROS as an enabler, we aim to reuse the modules in each development step, reducing time and increasing the system's safety. The remaining modules can be sustained if any modules are modified, and the tests can be repeated.

2. *Support the extensive test of cooperative control strategies.* CopaDrive will allow the realistic and repetitive testing of the system in several conditions and scenarios. This way, it will be possible to analyze the control systems' performance limits.

3. *Integrate the ETSI ITS-G5 communication stack.* As previously pointed out, communications play a crucial role in any Co-CPS. Thus, CopaDrive will integrate this communications standard early in the validation stages by supporting it directly in simulations and using real OBUs.

4. *Demonstrate the system in a scalable fashion.* The CopaDrive framework includes a robotic testbed to validate and demonstrate the designed controllers and safety systems in realistic and appealing scenarios. This testbed will allow the validation of real embedded computing platforms that will be used in actual vehicles in the final prototype. In addition, the migration of the different software modules could be done directly as we rely on ROS as the standard underlying middleware.

5. *Support other AVs and vehicular network developments.* CopaDrive's innovative modular product allows the controller and communications model to be modified for various applications involving AVs. Thus, it can be used for different applications and vehicular communication networks.

This chapter aims to demonstrate these objectives by applying the framework to develop a Co-VP system, including the cooperative platooning controller and the safety system, i.e., CLW, as carried out in the SafeCOP project. We will show how we used this framework to (1) carry out an analysis of a Co-VP control system, fully integrating the impact of the communications on its performance; (2) validate the CLW safety system in a virtual scenario via the HIL connection, using real OBUs which will be onboard the prototype vehicles and will integrate the safety system; and (3) to deploy the cooperative platooning system in the final embedded platforms that will be onboard the prototype vehicles, and demonstrate its behavior in a robotic testbed, indoors, in preparation for the final deployment in real-life vehicles.

5.2 Co-VP Safety Tools

Co-VP is a safety-critical system in which a malfunction can have severe consequences. Thus, besides the extensive testing and validation procedures these applications must undergo, additional safety monitoring systems must be in place, ready to trigger emergency action, as an emergency brake. Using different strategies, many safety monitoring systems have been proposed to increase Co-VP safety. For example, in [92], the author proposed a Machine Learning model to validate collision avoidance in Co-VP. In this work, the tool's limit is the number of trials and the system's reduced number of different situations to be learned. Another proposal aims at integrating machine learning with model checking strategies [204]. This work introduces a run-safety monitor that continuously evaluates safety conditions derived from a hazard analysis, previously formalized linear temporal logic.

Security of Co-VP applications is also addressed in some proposals. The authors in [29] proposed a middleware that supports an intrusion detection system to be integrated into the Co-VP nodes. This middleware contains a database of known malicious attacks. Then, it processes several system inputs to detect malicious attacks and issues alarms and notifications to the running application, increasing the system's security.

The author of [12] proposed a Runtime Verification Framework. This work introduces a framework to specify and generate code of runtime monitors based on a formal timed temporal logic.

This framework allows the development of multiple mechanisms to support safety-critical cooperative functions and the safety assurance processing in Co-CPS. Supported by this work, a generic ROS-based runtime monitoring system was built in the SafeCOP project, allowing the runtime monitoring of Co-VP systems by deploying ROS topics that could monitor different processes and hence increase the platoon's safety.

In addition to this work, in SafeCOP, a Control Loss Warning (CLW) safety mechanism was also developed to monitor the Co-VP system. Its most significant advantage is that it is deployed inside the OBUs, thus running separated from the Co-VP controller system implementation and thus guaranteeing complete isolation. Deployed in each platooning vehicle, this system's objective is to detect and alert about control loss situations that can be seen by comparing the movement of the preceding vehicle, as read by the received data with the vehicle's current status. This system behaves as an additional safeguard that can react in case the car platooning controller, for some reason, fails to comply with the platooning actions, for instance, by not reducing speed. In such a case, a CLW emergency action is triggered to prevent a crash by alerting all members of the platoon, as presented in Figure 5.2. This Figure represents the regular platoon course until some vehicles have an accident or a problem. Then, the CLW starts to act, performing a safety analysis and informing the following platoon vehicles, to avoid a collision. This detection and safety action is carried out by analyzing several vehicles' inputs regarding different safety case requirements previously analyzed and programmed in the CLW. As explained, alerts triggered by one node of the system may influence the behavior of other nodes by sending a CLW alert to the other elements involved. Those elements can be the other vehicles of the platoon, police, or emergency services.

Figura 5.2: Control Loss Warning

The system is deployed over Cohda's MK5 On-Board Unit [47] which is the component used to enable vehicular communications in the platoon. This rugged module is small, low-cost, and can be retrofitted to vehicles for aftermarket deployment or field trials in an off-the-shelf deployment manner. It has a Dual IEEE 802.11p radio, a Global Navigation Satellite System (GNSS) that delivers lane-level accuracy, and Supports DSRC (IEEE 802.11p), a critical implementation feature of ITS-G5 standard. Apart from the IEEE 802.11p support, these OBUs have, as well, ETSI ITS-G5 compatibility by providing a proprietary licensed software suite that enables this.

Figure 5.3, presents an overview of how CLW is implemented inside the OBUs. The OBU components are divided into two main groups: Apps Container and Apps Services. The Apps

Container is very simple and simply refers to the group of high-level applications running inside the OBU, namely, the CLW. The Apps Services comprises a set of libraries and applications whose only goal is to provide enhanced features to the upper applications. These services contain the Log Manager, Sensor Data Manager, Communications Manager, and Apps Monitor.

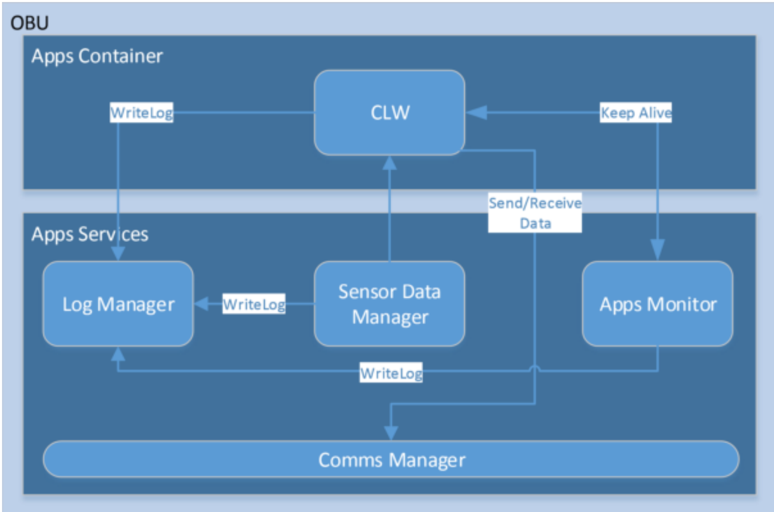


Figura 5.3: OBU Logical Architecture

The system architecture is supported by a three-layer architecture, where each layer provides services for the layer above. The layers are The communication layer, the Services layer, and the Application layer. From top to bottom, the Application Layer consists of a set of applications implementing the business logic from which they are responsible. In particular, the CLW application module is contained there. The services layer has four modules: The Communications manager implements the link between the OBU components, provides the communication channels between the OBU and the vehicle control computing platform, and assures the communications between the OBU in the platoon. The Log Manager consists of a library that each app or service will use to write a log to a file. The Sensor Data Manager module is responsible for listening to and processing data received from internal and external sensors of the vehicle. This module feeds CLW by providing direction, speed, and communications status information. The Apps Monitor is responsible for monitoring the running apps and implements a keep-alive mechanism to monitor the status of the CLW application.

So far, CLW enables the detection and transmission of alerts, which can culminate in triggering emergency actions in three scenarios: (1) failure to reduce speed, (2) failure to increase speed, and (3) failure to turn. The first two concern the failure of one vehicle to reduce or increase its speed at an acceptable rate by the platooning actions. The third is a vehicle's failure to change its orientation to maintain alignment in the platoon. These safety scenarios were implemented and validated in CopaDrive, and the results will be presented later in Section 5.2.

5.3 CopaDrive - Integrated System

Developing a safe cooperative driving system involves several test and validation stages, from the concept until its final implementation. Often, development effort is lost in this process, as the software modules used in the initial system validation, in simulation, for instance, cannot be directly deployed at the final prototype. The ROS framework allows a modular development of the system with a high level of decoupling. As the modules are independent and supported by the ROS middleware, it is possible to migrate them between different platforms and virtual and physical environments. Furthermore, it allows the developer to test the same module within the flexibility of a virtual simulation environment and deploy it directly onto a physical testbed, for instance.

For this reason, CopaDrive relies on ROS to support a continuous process of test and validation of cooperative driving systems.

As we will show regarding the validation of a Co-VP system, which includes a Co-VP controller and a CLW safety system, CopaDrive enables: (1) the validation of the Co-VP controller in a virtual environment, encompassing the impact of communications; (2) the migration of this controller to a HIL setup, enabling the validation of the real communications platform OBU and the CLW implementation in the mixed environment; (3) the migration of the Co-VP components into a platooning robotic testbed for additional validation and demonstration in a physical environment, over the final embedded computing platforms to be included in the final prototype. Thus, CopaDrive allows the test of the same Co-VP system from its initial development in the simulator to its implementation in a vehicle.

Initially focused on implementing a Co-VP system, its flexibility allows adaptation to several new scenarios, such as fixed communications devices (RSUs) or integrating EDGE and Cloud devices for external processing and information storage.

The overall view of the framework can be observed in Figure 5.4. Initially, the Co-VP controller is implemented and tested in simulation, using Gazebo and ROS. However, although the Gazebo simulation analyzes different control aspects, namely the lateral and longitudinal platoon stability in different scenarios, it does not provide information about the communication impact of the designed Co-VP system. So, as it is mandatory to fully understand the constraints of the controller and its safety limits, we analyze the ETSI ITS-G5 impact in our integrated CopaDrive Simulator - CD-S.

Inside the CD-S, ROS topics and middleware support the integration between Gazebo and OMNET++. CD-s allow the observation of how the platoon's stability and safety are affected by the delays and other communication problems realistically. Additional details and results are provided in section 5.4.

To validate the CLW safety system, the flexibility of quickly re-arranging the validation scenarios by changing the distances between vehicles, speed, and track was helpful. Thus, the validation can only be effectively achieved by merging virtual simulation scenarios with physical equipment

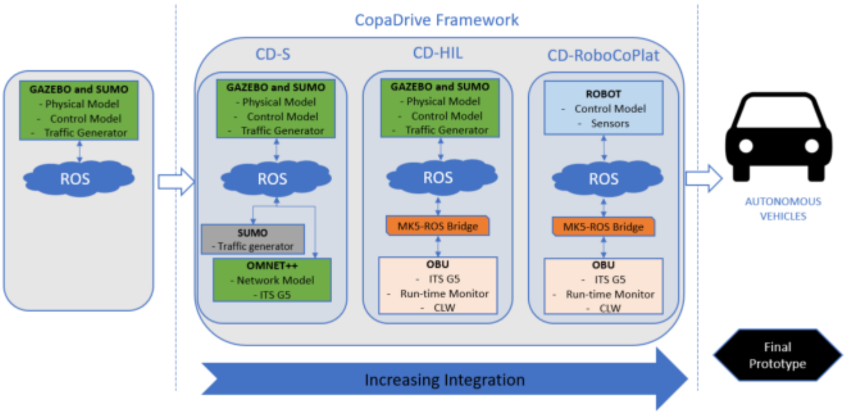


Figura 5.4: CopaDrive toolset and validation stages

and HIL integration. In addition, it was also necessary to test the integration between the Co-VP control system and the OBU communication platforms pre-deployment. Hence, we extended CD-S to integrate the required components in a tight simulation loop by removing the network simulator (OMNET++) from the CD-S and using the physical OBU equipment instead.

The implementation of CopaDrive Hardware in the Loop (CD-HIL) is described in section 5.5. To achieve this integration between ROS and the physical OBUs, we designed a *MK5-ROS Bridge* (section 5.5.1). Once again, maintaining the necessary modularity allowed the deployment of the same bridge component in the robotic testbed (CD-RoboCoPlat). In addition, the control model validated in CD-S is the same used in CD-HIL and the same that is migrated into the robotic testbed.

Therefore, this next stage consists of the migration of the Co-VP components into a cooperative robotic Testbed - CD-RoboCoPlat (Section 5.6). This stage allows the test and validation of all the main Co-VP system components as if to be installed in a final vehicle prototype, in a smaller scale robotic platform. Thus, besides integrating with different elements to be deployed in vehicles, they can be deployed indoors in controlled environments. They can partially replicate a realistic scenario at a fraction of the cost of real cars.

Evaluating the cooperative control algorithms, the communication interfaces and equipment, and the CLW safety system in the previous stage effectively reduces the development time and effort in the robotic testbed. Thus, it directly evaluates the system's performance and limitations as it runs in the final embedded computing platforms.

In the same way, as the underlying ROS system supports the migration from the controller module, validated in the CD-S and CD-HIL to CD-RoboCoPlat, the migration to the final autonomous vehicle prototype is straightforward as it shares the same computing platform. In addition, the validation process enabled by CopaDrive supports the development effort. It dramatically increases the confidence of the Co-VP application reliability before moving into the final testing phase in actual vehicles.

5.4 CopaDrive Simulator (CD-S)

To develop CD-S, we carried out the integration of a well-known ROS-based robotics simulator (Gazebo) with a network simulator (OMNET++) by extending Artery[232], enabling a robust framework to test and validate cooperative autonomous driving applications. On the one hand, we leverage Gazebo's robotic simulation's most prominent features, such as its support for multiple physics engines and its rich library of components and vehicles in integration with ROS, which enables us to build realistic vehicle control scenarios. On the other hand, OMNET++ supports the underlying network simulation relying on an ITS-G5 communications stack, which is currently the *de-facto* standard for C-ITS applications in Europe. This integration supports an accurate analysis of the impact of the communications upon the cooperative application and, on the other hand, the tools to evaluate the network performance using the OMNET++/INET framework.

This section overviews the CD-S, describing the tool, providing a set of relevant simulation scenarios, and carrying out a series of analyses regarding (1) the impact of communications upon the Co-VP controller, (2) the impact of the control technique on the network, and (3) by generating additional traffic with SUMO, we illustrate how the impact of the increase of external network traffic impacts platooning performance.

5.4.1 CD-S Central Components

5.4.1.1 Gazebo

A critical feature of ROS software is its flexibility in facilitating its integration with many open-source projects and tools. One of the ROS-supported and most used robotics simulators is Gazebo [218]. The Gazebo is an open-source 3D robotics simulator with multi-robots support for indoor and outdoor environments. It implements dynamic and kinematic and a pluggable physics engine. It provides a realistic rendering of settings, including high-quality lighting, shadows, and textures. It can model sensors that "see" the simulated environment, such as laser range finders, cameras (including wide-angle), or Kinect style sensors.

Many projects integrate ROS with Gazebo, like the QuadRotor presented in [205], the Humanoid implementation in [90] or the Ground Vehicle in [236]. As a powerful and very visual tool, Gazebo has also been used as the simulation environment for several technical challenges and competitions, like NASA Space Robotics Challenge (SRC) [122], Agile Robotics for Industrial Automation Competition (ARIAC) [209], and Toyota Prius Challenge [130].

5.4.1.2 OMNET++

The ETSI ITS-G5 [72] is considered the enabler, ready-to-go communications technology for such applications, and although there has been an extensive analysis of its performance [150, 149, 336], the understanding of its impact upon the safety of this SoS is relatively immature. Hence, extensive testing and validation must be carried out to understand the safety limits of such SoS by encompassing communications.

Several network simulators are available and capable of carrying out network simulations of vehicular networks. Nonetheless, these tools remain mostly separated from the reality of autonomous driving, offering minimal capabilities in evaluating cooperative autonomous systems.

In this work, OMNET++ is the network simulator. It allows the development of several network stacks using different modules and frameworks. For instance, the VEINS [257] implements the standards of IEEE WAVE and ETSI ITS-G5. The VEINS is responsible for implementing the PHYSICAL and MAC layer in OMNET++ Simulation. And Artery [232] is the VEINS extension to VANET applications. During the research for this work, we identified Artery [232] as the most mature project providing ITS-G5 free implementation.

5.4.2 Framework Architecture

The CD-S architecture is presented in Figure 5.5. The underlying operating system was Linux Ubuntu 18.04.6 Bionic, with Gazebo 9.0, ROS Melodic, and OMNET++ 5.4. The computing platform used for the integration and simulations featured an Intel® Core® i7-975H CPU, with 16 MB RAM memory and a NVIDIA Geforce GTX 1650.

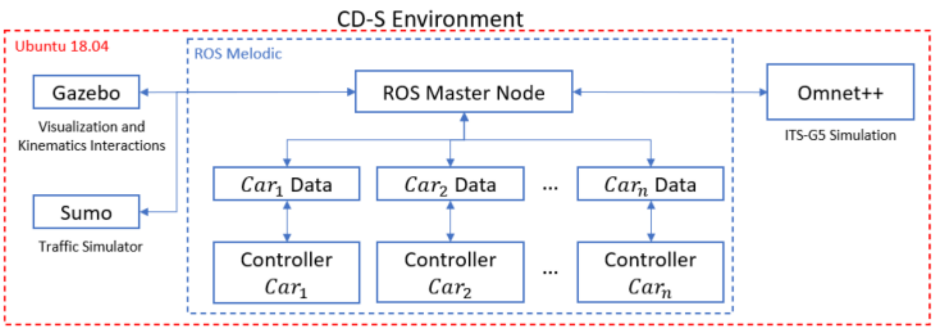


Figura 5.5: CD-S Simulation Environment

5.4.2.1 Synchronization Approach

The OMNET++ is an event-driven simulator, while Gazebo is a time-driven simulator. So, synchronizing both simulators represented a key challenge, and a synchronization module was implemented in OMNET++ to carry out this task, relying upon ROS "/Clock" topic as a clock reference. The OMNET++ synchronization module subscribes to ROS' "/Clock" topic, published at every Gazebo simulation step (i.e., every 1ms). It schedules a custom-made OMNET++ message ("syncMsg") to an exact ROS time. This message allows the OMNET++ simulator engine to generate an event upon reaching that timestamp and to be able to proceed with any other simulation process that should be running at the same time (e.g., CAM generation by CAService).

5.4.2.2 OMNET++'s Modules Overview

We built the simulator framework over the Veins simulator and the Vanetza communications stack implementation, borrowing and extending much of the middleware components from the Artery framework. It relies on ROS publish/subscribe mechanisms to integrate OMNET++ with Gazebo, represented in Figure 5.6. ROS/Gazebo set is responsible for the movement of the vehicles, simulating the physical model of interaction between the cars, track, and obstacles. Using the *MobilityROS.h* library, OMNET++ can subscribe to the ROS threads through the Vehicle Data Provider (VDP), representing the vehicle's network interface. This information populates a single object regarding instantaneous vehicle data, including position, velocity, and acceleration. CaService then uses this object to fill the ETSI ITS-G5 CAM fields when the message triggers are activated. CaService is further responsible for encoding these messages in the ETSI ITS-G5 ASN-1 [82] standard and sending them to the network layer provided by Vanetza. VDP also provides GPS coordinates to define the position of the nodes in the INET mobility module. The sender process is illustrated in detail in Figure 5.7.

The message follows the layers described in the IEEE 802.11p model, reaching the physical layer provided by Veins. Then, OMNET++ simulates the delivery, adding delays and evaluating the probability of collisions and packet loss. Then, OMNET++ proceeds with the reception of this message in the other vehicles. CAMs are messages sent in the broadcast. All vehicles with ETSI ITS-G5 radios in range of the sender's radio should receive the message sent unless there is a collision. The message is received on the receiving vehicle's radio interface by Veins and transmitted to Vanetza at the network layer. The arrival of a message on this channel triggers an evaluation of its content, verifying if it is a CAM or not. In this case, CaSetvice receives the message and, if it is a CAM, proceeds with its treatment or discards it. After checking the fields, a signal is fired to the Robot Middleware (RM). The RM fills in the ROS topic to be published for that specific vehicle and posts it. On the ROS/Gazebo side, the vehicle's radio simulator subscribes to this information and proceeds with the control action to ensure the platoon's continuous movement. The receiver process is illustrated with more details in Figure 5.8.

5.4.2.3 Data Workflow

Figure 5.9 presents a quick overview of how data flows from car_i sensors into car_{i+1} control application, working its way through Gazebo into OMNET++ and then into other Gazebo's car following a CAM transmission between different nodes in OMNET++. To note that nodeX and nodeY represent the network interface of both car_i and car_{i+1}, respectively.

To convey the required information between vehicles, we set up the Protocol Data Unity (PDU) in the messages, as observed in Figure 3.3. This Figure presents the general structure of the data 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.

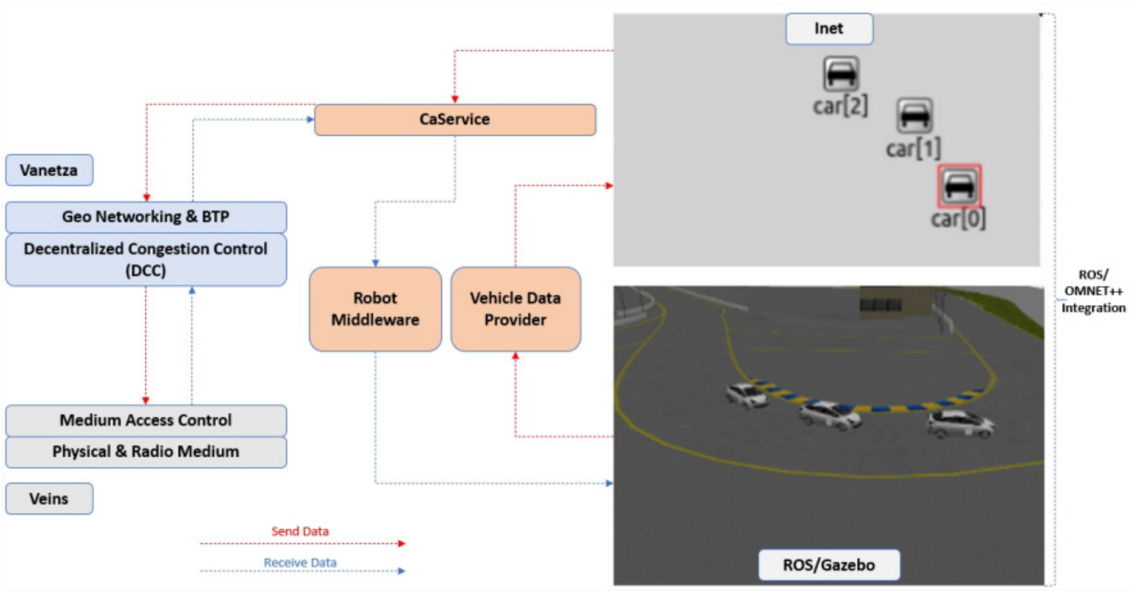


Figura 5.6: Framework Architecture

5.4.3 Experimental Results

The simulation is composed of three vehicles, modeled from a Toyota Prius, running a PID-based platooning control model [149]. This controller solely relies on CAMs to maintain the platooning service, with a safe distance set to 8 meters, using the *CSP* as distance policy. The simulation results were extracted from 45-seconds long runs in four scenarios where the platoon safety was assessed. Scenario A applies fixed CAM frequencies, while the others utilize event-triggered messages. So, scenario B used the ETSI ITS-G5 basic configuration [82], defined here as *CAM Basic Service Profile (CAM-BSP)*, while scenario C used the *CAM-BSP* with platooning-defined specifications [72]. Finally, scenario D applied and evaluated customized settings for *CAM-BSP*, defining a *CAM-CSP*. The simulation environment and simulated platooning trajectory in the 45-second run are depicted in Figure 5.10.

The yellow line represents the initial acceleration path, where the follower is still accelerating to reach the set point distance between itself and its leader. In orange, the road track where the platooning follows a straight line, and in red, a hard turn in which platooning behavior is significantly dependent on the number of CAMs exchanged. The presented experimental results also contain this color reference to help relate them with the track's relevant portion.

In the simulations, as standardized, CAM messages are distributed in a broadcast fashion and triggered according to the details provided in Table 5.1. In this table, Δp means a variation in the position, Δh in the heading of the vehicle, and Δv in the velocity of the local leader, following the description provided in [33]. Although messages are broadcast from each car to all the others, only the messages sent by the previous vehicle are used by the follower.

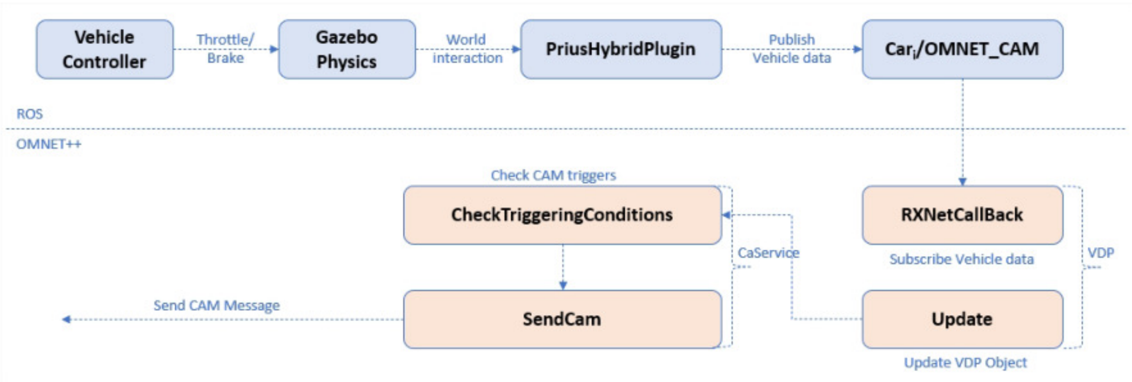


Figura 5.7: CAM Sender Fluxogram

5.4.3.1 Proposed Scenarios

- Scenario A: Fixed CAM frequencies

Four CAM sending frequencies were evaluated (i.e., 10, 5, 3.3, and 2.5 Hz), guaranteeing that CAM messages will always be provided with new information at the highest CAM frequency. We analyzed the impact of these different CAM exchanging frequencies on the follower's behavior regarding the forward distance and steering angles to analyze how different CAM exchanging frequencies affected the Co-VP control. The Co-VP starts from a parked position, and the follower only engages in platooning after the leader starts moving forward, so the follower needs to accelerate to catch up to its leader.

Figures 5.11 and 5.12 shows the vehicle inter distance in each test and Figure 5.13 presents the steering angles.

At higher CAM sending frequencies, the Co-VP PID controller shows better stability, and the inter-distance error decreases. These issues are also evident regarding steering behavior. For the first three CAM inter-arrival times, the steering angles follow the leaders with a slight delay, which increases with frequency. For an inter-arrival time of 0.4 s, the steering angles of the follower are no longer in line with the leader's (Figure 5.13). It is also clearly noticeable that, for a CAM inter-arrival time of 0.4 s, while approaching the left-hand turn (in red), the follower lost track of the leader vehicle, making a complete stop.

CAM sending frequency is too low to keep the follower updated with the leader's steering

Tabela 5.1: Trigger to CAM messages

Scenario A	Scenario B	Scenario C	Scenario D
Fixed Frequency	CAM-BSP	CAM-BSP-P	CAM-CSP
$10Hz$	$1s$	$0.5s$	$0.5s$
$5Hz$	$\Delta p > 4m$	$\Delta p > 4m$	$\Delta p > 2m$
$3.3Hz$	$\Delta h \geq \pm 4°$	$\Delta h \geq \pm 4°$	$\Delta h \geq \pm 4°$
$2.5Hz$	$\Delta v \geq 0.5m/s$	$\Delta v \geq 0.5m/s$	$\Delta v \geq 0.5m/s$

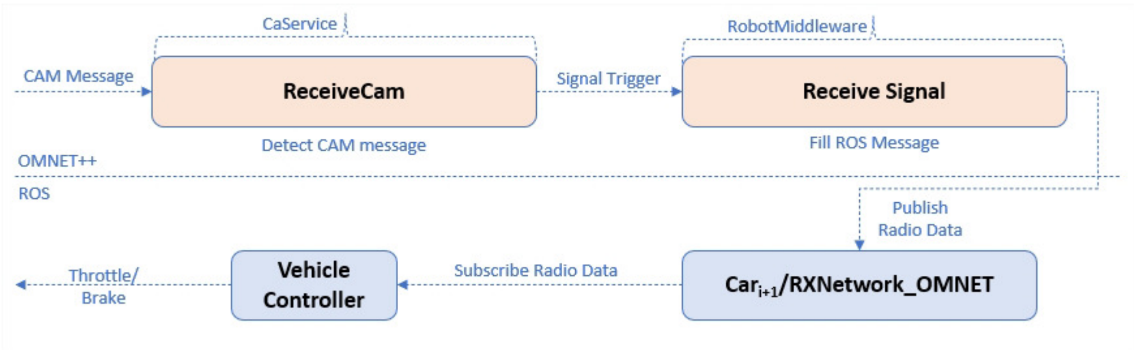


Figura 5.8: CAM Receiver Fluxogram

corrections, resulting in minimal or nearly non-existent steering inputs. As a result, upon entering the left U-turn, the follower's controller struggles to keep up with the leader's heading, failing for inter-arrival times of 0.4 s. The direct relation between higher CAM frequency and PID steering controller stability was demonstrated among several runs at different frequencies. However, fixing a CAM frequency represents a sub-optimal approach for Co-VP, considering that excessive CAM traffic will often be generated, negatively impacting the network's throughput.

To evaluate CD-S stability and limits, we analyzed its inherent latency and computing delays. Figure 5.14 presents the delay between an OMNET++ node reception of a CAM (from a network transmission) and its reception by the Gazebo's vehicle model after receiving it through ROS Pub/Sub mechanisms.

Following Figure 5.9 timeline, the timestamps recorded were taken at CAM reception at the node's middleware and upon Gazebo's car application callback on this referred ROS topic at different CAM sending frequencies (10, 5, 3.3, and 2.5 Hz). It was possible to detect that the delay mostly comes from the ROS underlying Pub/Sub mechanisms. The system's performance doesn't seem to be severely affected by CAM's delay between sending and receiving messages since it remains almost constant during the simulation. Although the maximum obtained delay slightly increased with traffic, the observed latency close to 0.25 milliseconds is insufficient to impact or compromise the application under test.

- *Scenario B: Basic Service Profile*

This scenario allows the analysis of CAM as standardized in ITS-G5 [82] or BSP. BSP defines an interval of 0.1 seconds to 1 second between CAMs, except upon one of the following conditions, at which a CAM message must be immediately triggered:

- the absolute difference between the current heading of the originating vehicle and the heading included in the CAM previously transmitted by the originating vehicle exceeds 4 degrees;
- the distance between the current position of the originating vehicle and the position included in the CAM previously transmitted by the originating vehicle exceeds 4 m;

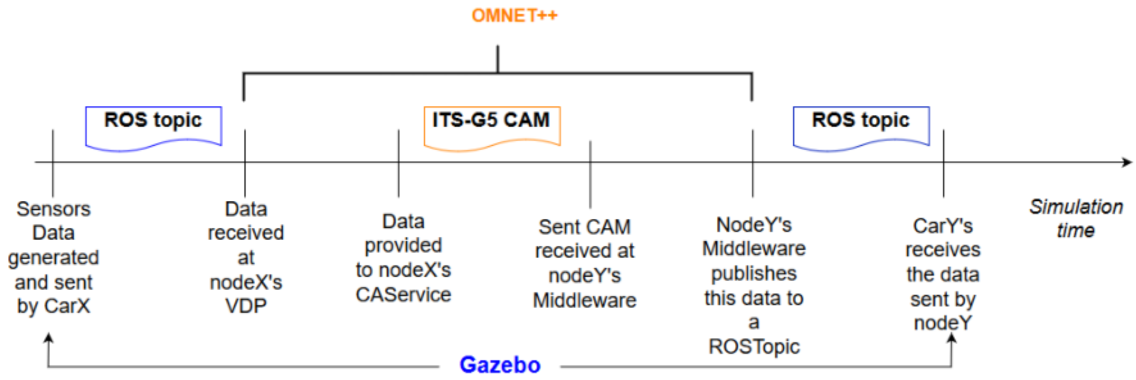


Figura 5.9: Data workflow

- the absolute difference between the current speed of the originating vehicle and the speed included in the CAM previously transmitted by the originating vehicle exceeds 0,5 m/s.

CAM reception intervals are presented in Fig. 5.17. CAM sending frequencies approach 2.0 Hz, mostly due to the second triggering condition, since the CoVP speed during the orange part of the track is constant at around 8 m/s. However, there are some high-frequency triggers in the early iterations, resulting from the quick acceleration at the initial portion of the track while trying to close the distance gap to the leader. The CAM BSP triggers higher frequencies in response to the hard left turn (red portion of the track), which causes a quick shift in the leader's heading. Still, as observed in Figures 5.16 and 5.15, this increase in frequency was insufficient to maintain a stable Co-VP using this control model and failed to follow the leader's steering control. Therefore, we conclude that CAM-BSP is not well-tuned for more demanding Co-VP scenarios, in which the control models exclusively rely upon cooperative support.

- Scenario C: Basic Service Profile for Platooning (CAM BSP-P)

In this scenario, we analyze an extension to the ITS-G5's CAM-BSP specified in [72]. This ETSI report recommends improved CAM-BSP settings for Co-VP. One of its most significant changes was to limit the minimum frequency between CAM transmission to 2 Hz, double the one defined for the original CAM-BSP. This profile will be designated as CAM BSP-P.

Test results were similar to the usage of the original CAM-BSP settings, as triggering conditions remain the same. As depicted in Figure 5.18 and similarly to scenario B, CAM inter-arrival times remain around 2Hz, in this case, as a result of the minimum frequency limit set. Concerning Co-VP behavior, figures 5.16 and 5.15 depict a similar behavior to scenario B, which fails to execute the U-turn. It happens as a consequence of platoon instability. For instance, concerning distance error regarding the setpoint, scenarios B and C present similar and significant errors resulting from low CAM update frequency.

- Scenario D: Custom Service Profile for Platooning (CAM-CSP)

For this scenario, we set up a Custom Service Profile to balance the network load originated by CAM exchanging while guaranteeing stability. Our approach was to adapt the second CAM triggering condition mentioned in scenario B by changing it to 2 meters instead of 4 meters. This

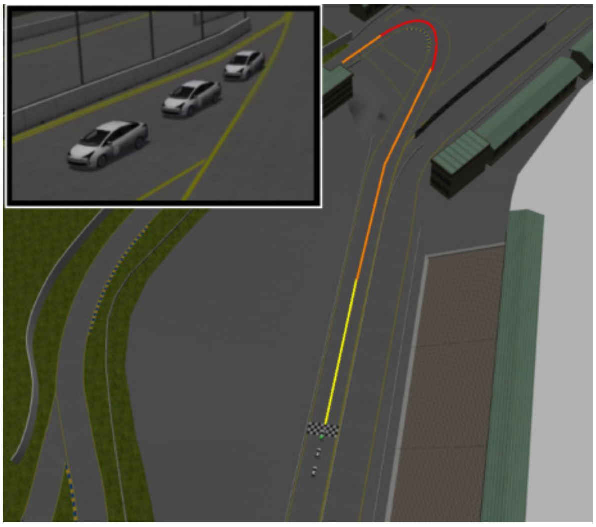


Figura 5.10: Platooning Trajectory

change impacted the Co-VP behavior considerably, both in the number of CAMs sent and its frequency, as it's possible to check at Figure 5.19.

As shown in Figure 5.18, the CAM-CSP conditions caused CAM triggering to happen much more frequently than in previous cases. It result in a more stable Co-VP controller when compared to scenarios B and C (Figures 5.16 and 5.15). It also causes a significant decrease in distance error, leading to a smoother control. In fact, only this change enabled the Co-VP to complete the hard left turn (Figure 5.10).

5.4.3.2 Network Impact upon Co-VP performance

Having learned the shortcoming of the ETSI ITS-G5 standard in supporting such a Co-VP controller, we proposed CSP as a new CAM profile to mitigate the performance issues. This minimal change to the CAM triggering conditions proved quite effective in guaranteeing the platoon's safety. However, it is also important to evaluate how the network is impacted by these CAM triggering setups. CopaDrive enables this analysis by looking into the several metrics provided by OMNET++. This analyses relies upon metrics such as *Network Throughput*, *Application end-to-end delay* and *update delay* in the given tests.

As observed in Figure 5.20, given the reduced size of the packet length and the reduced number of messages, the measured maximum throughput of the channel does not grow beyond 0.13% for a platoon of 3 vehicles. It is possible to observe that the CAM-CSP does not increase the throughput much more than the fixed frequency of $3.33Hz$ (period of $0.3ms$). It also has the advantage of having a kinematics-triggered CAM instead of a purely time-triggered approach to CAM transmission, which is much closer to the objective of the ETSI ITS-G5 standard. Although the throughput of CAM-CSP is higher than the case of CAM-BSP and CAM-BSP-P, it is smaller than in scenarios of $10Hz$ and $5Hz$.

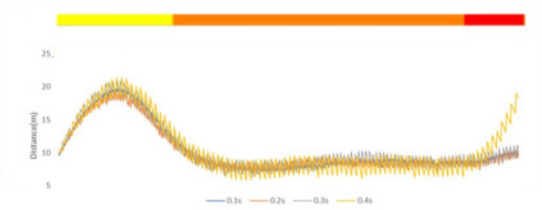


Figura 5.11: Vehicle inter-distances - Scenario A

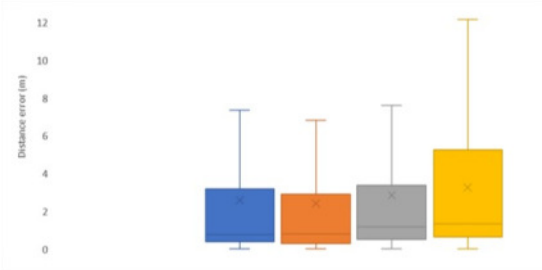


Figura 5.12: Distance Error - Scenario A

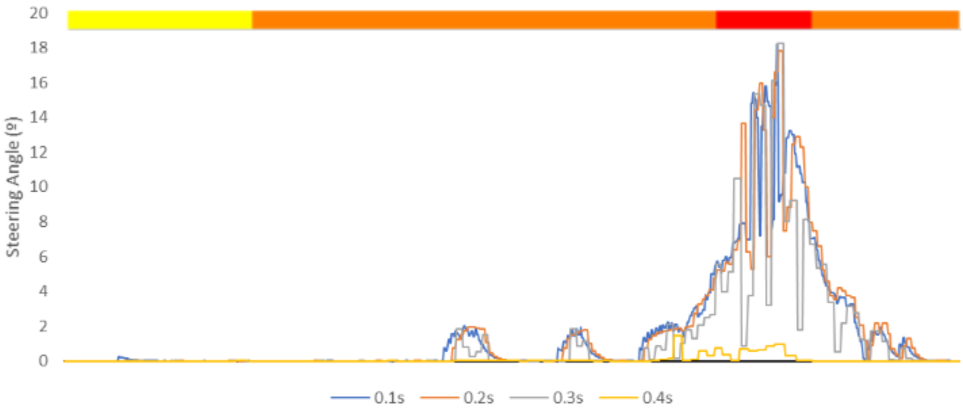


Figura 5.13: Steering Angles - Scenario A

As the throughput in the network is small and there is no packet loss, there's no congestion that can increase the probability of collisions in the network. So, the *end-to-end* delay was the same for all the tests and fixed in $0.544ms$.

Due to the new trigger condition, the number of packages sent in the CAM-CSP is more significant than the number in CAM-BSP and CAM-BSP-P. However, it is smaller than scenarios of $10Hz$ and $5Hz$. The analysis of Figure 5.20 shows that the fixed frequency of $3.33Hz$ is very close to the proposed CAM-CSP. The most significant difference is the flexibility of the network in the CAM-CSP.

This analysis can be confirmed in Figures 5.21a, 5.21b and 5.21c. Those figures shows the *update delay* between the messages received in car_1 that have been sent by car_0. The *update delay* is the measured time between received messages in a node of the network. They also show that most of the exchanged messages between those cars in CAM-CSP are received at least in $0.3ms$. In CAM-BSB and CAM-CSP, as sometimes the messages can be delivered in more than $0.5ms$, the follower does not receive the messages in time to guarantee the platoon's safety.

Table 5.2 summarize the results of the scenarios presented in this section. It presents the number of CAM messages sent during the simulation for each scenario. So, a fixed frequency between 3.3 Hz and 2.5 Hz should be at the threshold borderline balance to maintain Co-VP stability. However, fixing this frequency is not the most reasonable approach since it can cause unnecessary CAM message transmissions or, in some extreme situations, may not suffice. Thus, it is much better to have this CAM triggering approach dependent on vehicle kinematics, as proposed

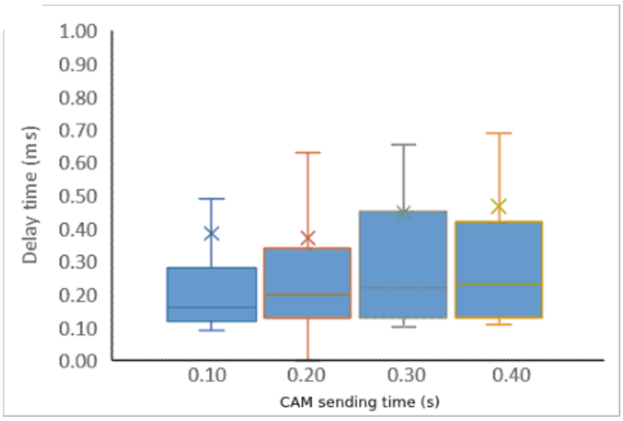


Figura 5.14: CAM Exchanging Delay - Scenario A

in the standard. With this in mind, as defined in ITS-G5, a Service Profile should be the optimal way to handle this. However, this kind of profiling should be adapted to the use case and the control model. For this particular control model under test, CAM information availability is crucial to maintain the platoon's stable and safe behavior. Thus, a new service profile was proposed. This kind of evaluation can be easily carried out using our framework by fully specifying the simulation environment and Co-VP control model over ROS/Gazebo while using OMNET++'s capabilities to analyze the network performance, carrying out an integrated in-depth analysis of the cooperative driving application behavior.

Tabela 5.2: Comparison between Scenarios - Number of messages and Safety Guarantee

Scenarios	Fixed Frequency (Hz)				BSP	BSP-P	CSP
	10	5	3.3	2.5			
Number of Messages	1502	756	504	336	342	359	655
Network Throughput (%)	0.133	0.066	0.044	0.033	0.028	0.031	0.057%
Maximum End-to-end Delay (ms)	0.544	0.544	0.544	0.544	0.544	0.544	0.544
Safety	OK	OK	OK	NOK	NOK	NOK	OK

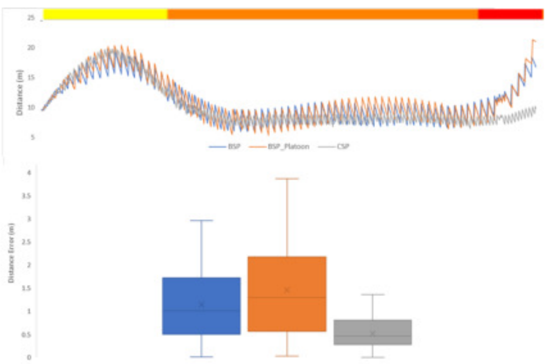


Figura 5.15: Longitudinal distances analysis in scenarios B, C and D

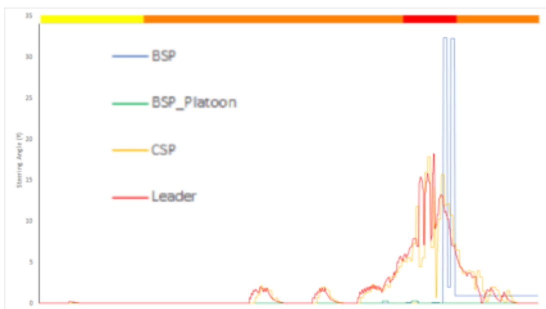


Figura 5.16: Steering Analysis for scenarios B, C and D

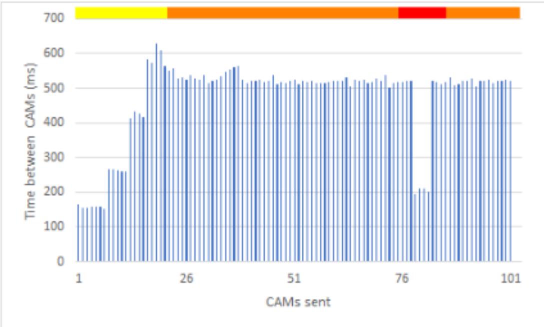


Figura 5.17: Period CAM-BSP

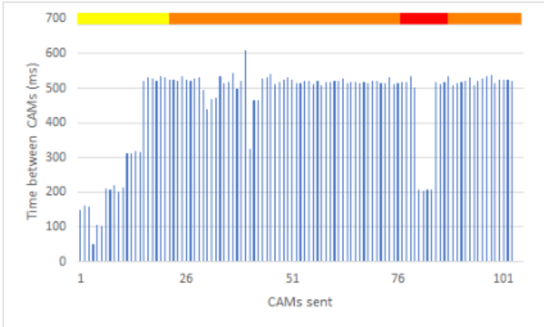


Figura 5.18: Period CAM-BSP Platoon

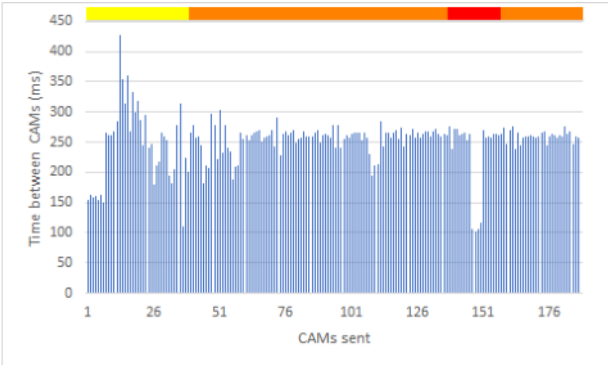


Figura 5.19: Period CAM-CSP

5.4.4 Traffic Analysis

To increase the analysis of the impact of the increasing traffic on communication performance, the CD-S uses the SUMO to create traffic. The integration of the SUMO traffic generator into CD-S was accomplished using the *TraCI* framework [306]. TraCI is an open-source software that enables the expansion of SUMO and the connectivity to OMNET++. It consists of several methods that retrieve all kinds of data from the traffic generator, including the number of cars currently present in the simulation and all the essential information regarding them, including position, speed, heading, etc.

By injecting this data retrieved from the *TraCI* API on the ROS topic, the same vault that stores the information of the vehicles on the simulation also contains the ones generated outside the platoon, and, from then, the OMNET++ simulation will be updated, disclosing the car nodes on the simulation playground. With this connection fully implemented and given that those nodes are present in OMNET++, the integration with the ROS system was similar to the already existing one. However, the messages created were published to a specific topic designed to accommodate only the information related to the cars generated by the SUMO software.

The integration of SUMO to the previous system version allows the insertion of planned traffic into the designed scenario. So, it is possible to analyze how the traffic impacts the ITS-G5 communication. For example, we built a design where a three-vehicle Co-VP runs on a road with an increasing number of other autonomous vehicles. Those vehicles are running on a parallel

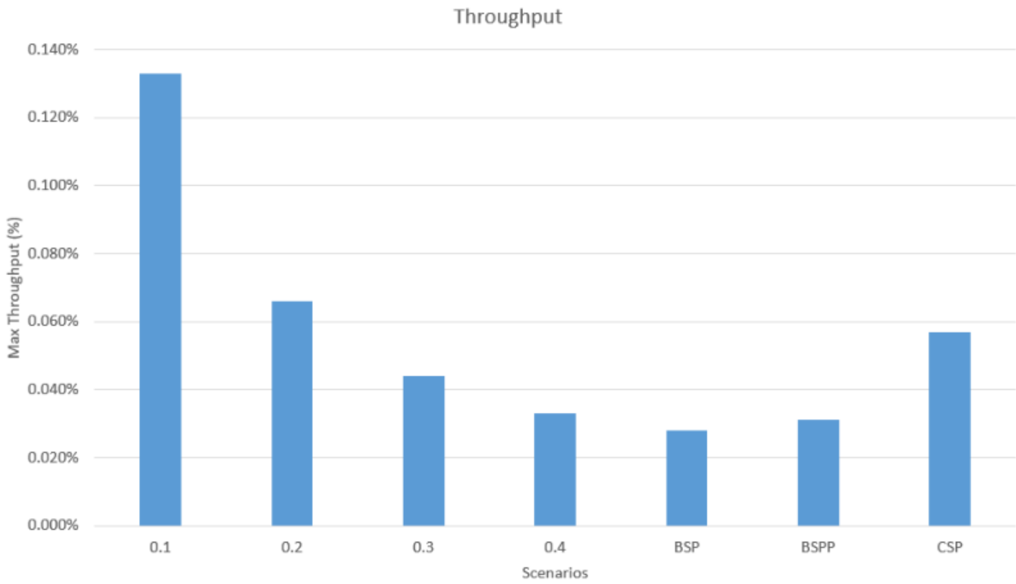


Figura 5.20: Throughput Analysis for different scenarios

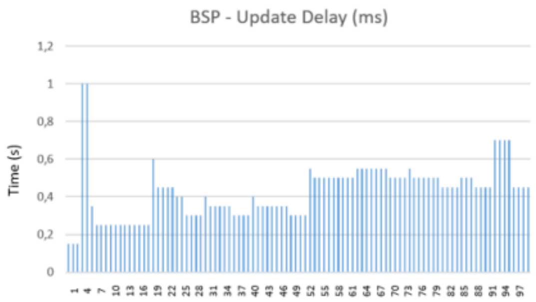
highway at a constant speed, using ITS-G5 communication with the same standards as the vehicles in the platooning. Then, even though those extra vehicles are not a part of the platooning, they also occupy the communication channel and impact the communication between the Cooperative vehicles.

The number of vehicles in the traffic provided by SUMO increases from 1 to 30, being released in two-second intervals. This approach prevents the sudden overflow in the simulation that might generate performance issues. That way, the simulation represents a more realistic scheme. In a real-life traffic scenario, the random influx of vehicles on the road or highway appears in a continuous mode, not all simultaneously, as presented in [321]. The same communication profiles previously studied are analyzed with SUMO to observe the communication impact over the network. The designed scenario is presented in Figure 5.22.

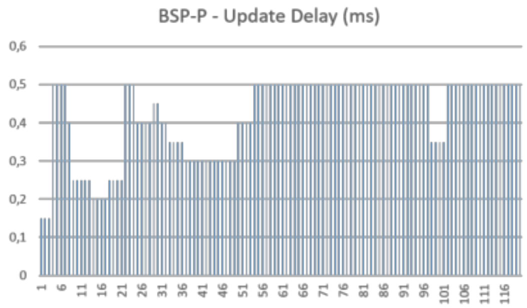
Compared with the previous scenario, with only three vehicles, the increasing number of cars increases the messages delivery time, as can be observed in Figure 5.23, given that all the vehicles transmitted and received the CAM messages. It was also possible to compare the network throughput in the designed scenario, as presented in Figure 5.24. This Figure illustrates that the amount of data traveling in the network channel substantially increases when the number of cars increases. As expected, the throughput in the CSP scenario is bigger than the other communication profiles.

5.5 CopaDrive Hardware-in-the-Loop - CD-HIL

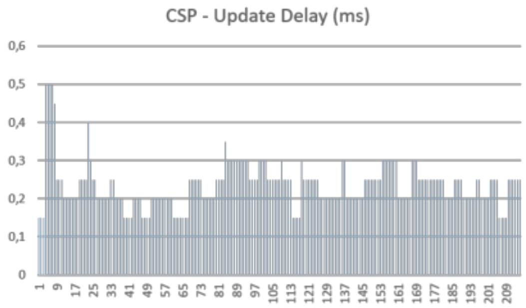
The previously evaluated Co-VP application is logically deemed as safety-critical. Thus, one must have additional mechanisms, such as the already mentioned CLW, to trigger an emergency action upon detecting a Co-VP system failure. This section concerns the evaluation of the CLW mechanism as implemented in a real OBU. Furthermore, it was essential to validate the integration



(a) CAM-BSP Update Delay



(b) CAM-BSP-P Update Delay



(c) CAM-CSP Update Delay

Figura 5.21: Scenarios B, C and D: Update Delay

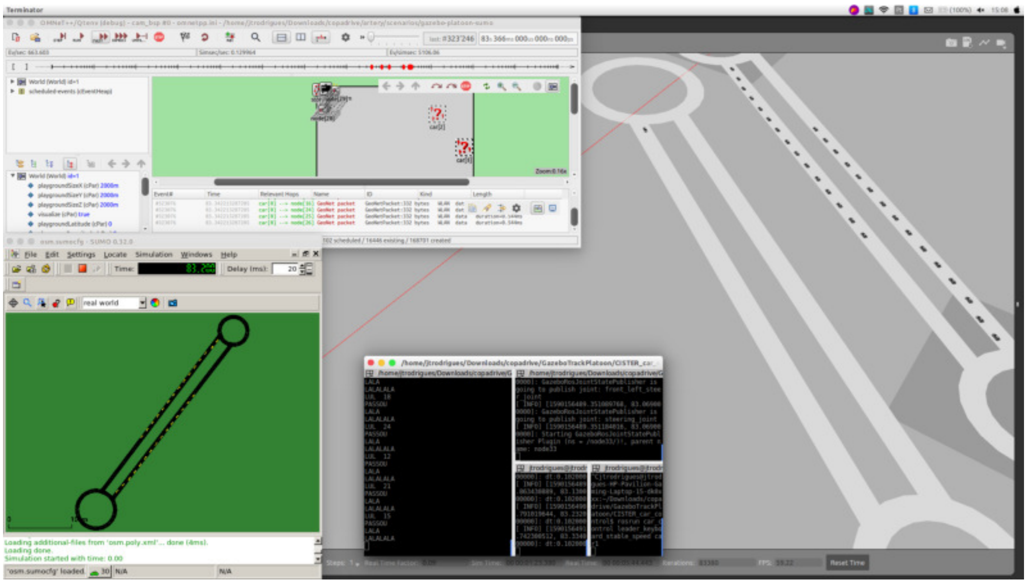


Figura 5.22: CD-S Traffic Simulation Scenario

between the Co-VP control model and the OBUs, which could only be carried out if we relied upon a HIL approach. Usually, such endeavor would require a significant effort to port the already tested control models and systems into a new simulator. Instead, we rely upon the same CopaDrive framework and ROS-enabled flexibility to replace the OMNET++ network simulator component

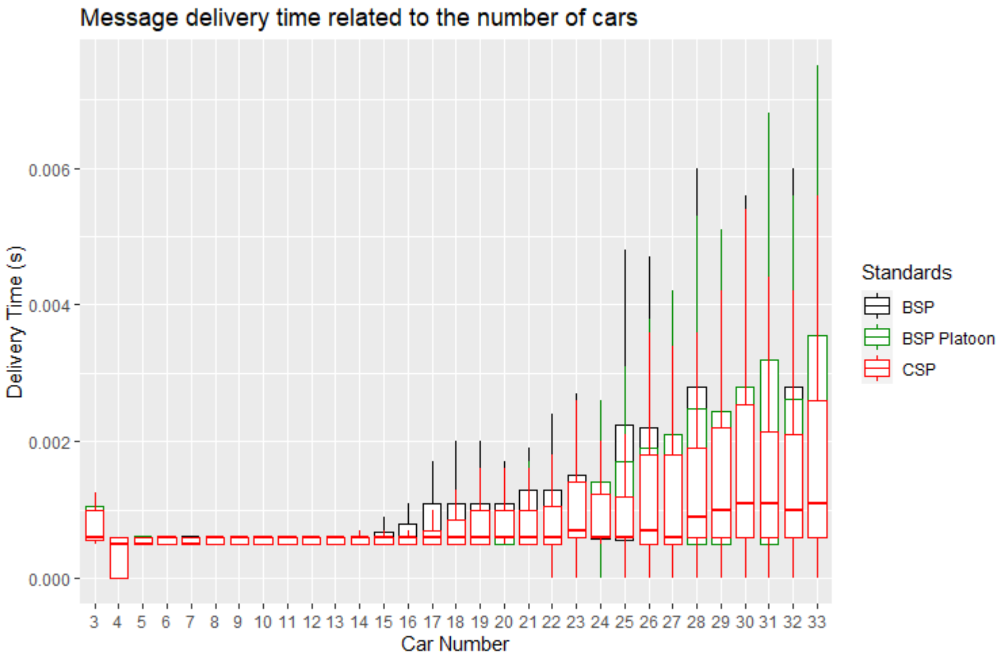


Figura 5.23: Messages Delivery Time in Vehicular Traffic

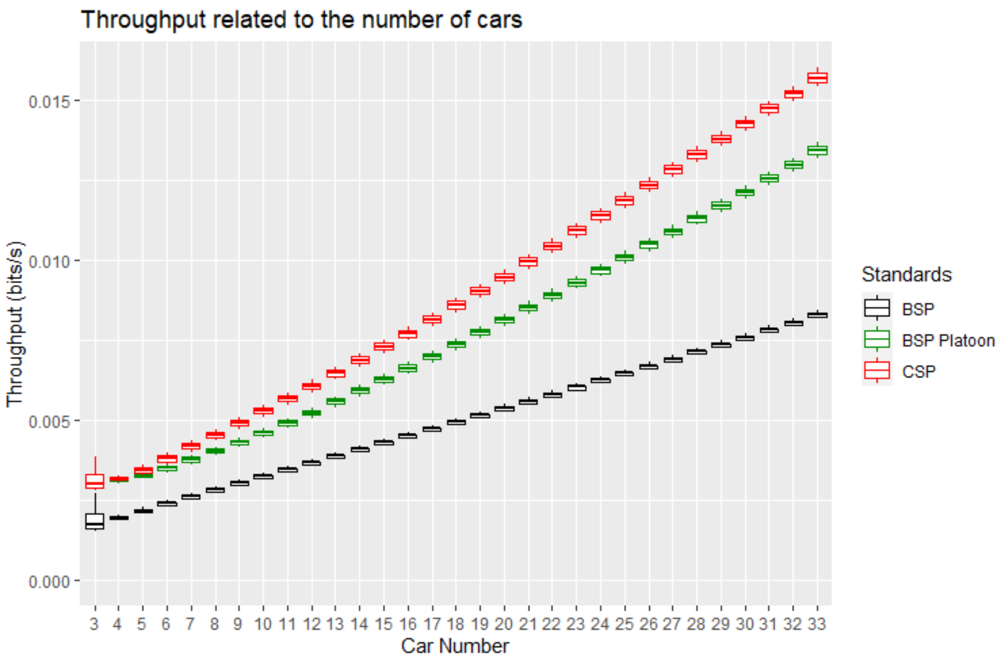


Figura 5.24: Network Throughput in Vehicular Traffic

with real OBUs for the three vehicles to handle communications while keeping the same Co-VP system.

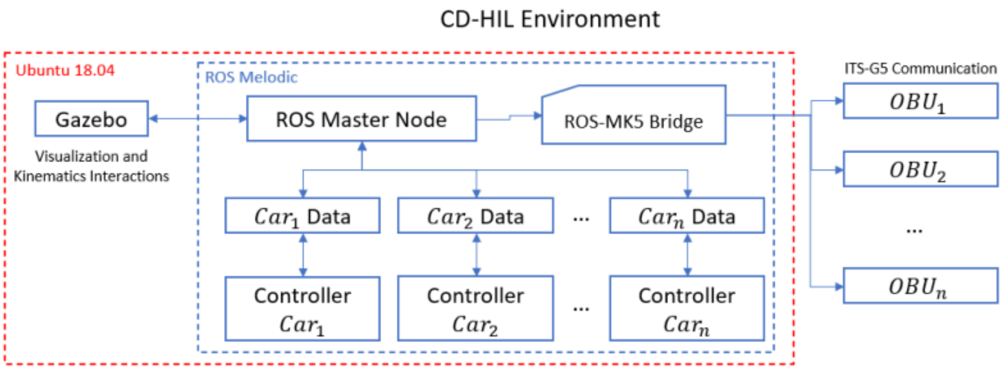


Figura 5.25: CD-HIL Architecture

The CD-HIL implementation aims at supporting the test and validation of the OBUs and the CLW safety mechanisms for the Co-VP application. By replacing the network simulator in CD-S with OBUs for each vehicle, as shown in Figure 5.25, we force all communications to be handled by the real communications platforms that will be on board the vehicles. Its main advantage is keeping the flexibility of the virtual scenarios while the CLW safety mechanism is evaluated.

Figure 5.26 showcases a deployment of the system, in which the three OBUs are visible and connected to the simulator via ethernet. A ROS module had to be developed to serve as a bridge between the ROS sub-system and the OBUs, by conveying the vehicle information from the ROS topics, into the OBUs, for CAM transmission and vice-versa, to map the received data in each OBU into ROS topics, and feed it into the vehicle Co-VP controller.

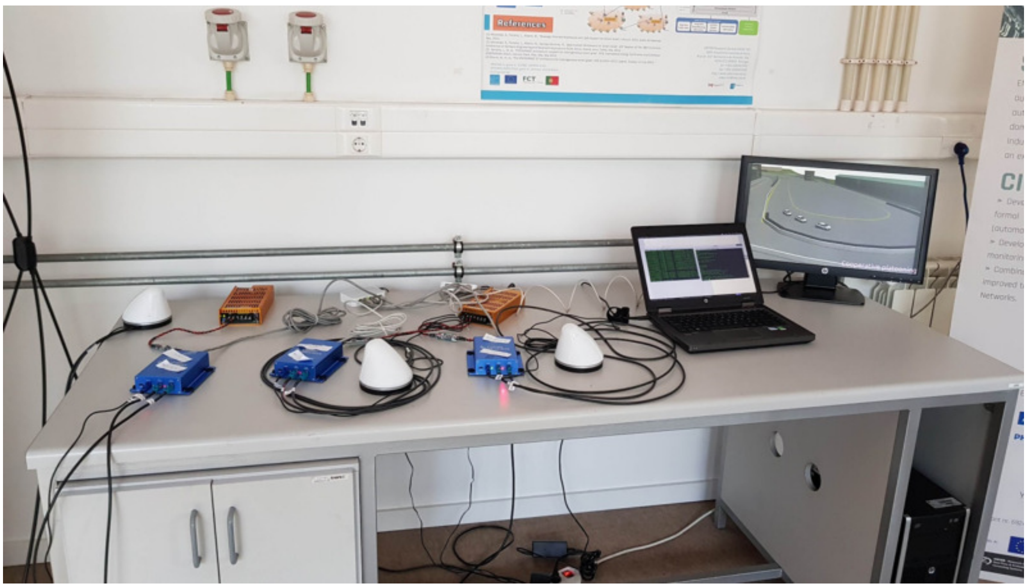


Figura 5.26: CD-HIL Deployment

The HIL architecture allows the evaluation of CLW, which is fundamental in the OBU integration before real deployment. In what follows, we present the details of this module. Next, we will deliver the failure scenarios in which CLW was validated and its assessment.

5.5.1 ROS-MK5 Bridge

The CD-HIL platform integrates the OBUs with the ROS environment and vehicle control systems using the ROS-MK5 Bridge. This communication is performed through TCP sockets, using IPV4 or IPV6. On the OBU side, the messages are processed by the message broker and used by the above modules. On the vehicles' side, the ROS-based control system uses the information received through the bridge to understand better its surrounding environment and the cars involved in the platoon. The ROS-MK5 bridge provides a bi-directional bridge between ROS systems implementing a platoon simulation model running on Gazebo and the MK5 OBUs from Cohda Wireless. In addition, the bridge allows a ROS environment to connect and communicate with other ROS environments through MK5 802.11p OBU platforms. An overview of the ROS-MK5 bridge's primary interfaces is presented in Figure 5.27.

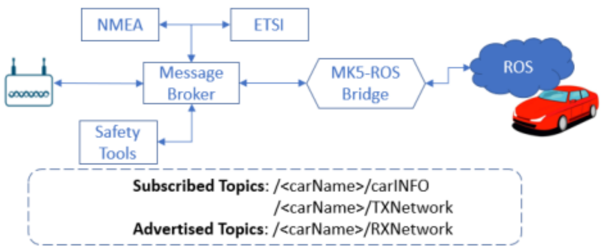


Figura 5.27: Bridge Architecture

The blue cloud named ROS represents the vehicle platooning simulation environment using Gazebo on the right-hand side of the Figure. This Bridge end-side subscribes to topics from the simulator to provide their information through the bridge to feed the OBUs. Nevertheless, it advertises a topic filled with data from the Bridge in the opposite direction, so the simulator can properly acknowledge this.

On the OBU end-side of the Bridge, all the information received from these ROS topics is fed into the Message Broker module, which segregates and processes this data to prepare the respective CAM transmission.

NMEA messages, for positioning, are also filled in the OBU via the information received by the Message Broker to the NMEA server. These are used to provide GPS coordinates, speed, and heading of the correspondent vehicle according to the vehicle position and speed in the simulator. The NMEA Server running in the OBUs computes them, so their information can be used by the remaining MK5 run application. This same module is also reused in the CD-Testbed. The ETSI module is responsible for filling in the necessary information at the standardized message container according to the ETSI ITS-G5 format. Also, when the OBU receives a CAM transmission, the

ETSI module decodes the message in the opposite direction. Finally, it sends its content to the Message Broker to be published in the respective ROS topic.

5.5.2 Experimental Results

The CLW is a safety tool based on a software module running directly on the OBU and receiving data from the ETSI module (e.g., position, speed, heading). This data arrives either from the simulator, if it concerns the ego vehicle, or from other OBUs, according to the format specified in Figure 3.3, acknowledging the position and status of the other platoon members. The CLW uses and compares this data to detect (or even predict) a failure properly.

We consider three failure scenarios, each simulating a different type of failure on the follower vehicle controller.

- *Failure to Increase Speed (FIS)*- In this scenario, the leader, on a straight path, accelerates, and its follower is not able to respond to this because of a failure in its acceleration system. The CLW/RT monitor detects the failure and alerts its control application.
- *Failure to Decrease Speed (FDS)* - In this scenario, the leader, on a straight path, brakes, and its follower is not able to respond to this because of a failure in its braking system. The CLW/RT monitor detects the failure and alerts its control application. This scenario is presented in Figure 5.28. Without the FDS, the follower collides with the leader when the failure happens in this Figure. With FDS, the followers can stop before crashing, given that the CLW detects the failure and actuate over an emergency brake.
- *Failure to Change Direction (FCD)* - In this scenario, the leader takes a turn, and its follower is not able to respond to this because of a failure in its steering system. The CLW/RT monitor detects the failure and alerts its control application.

In all these scenarios, the failures are injected into the second platoon member of a platoon with three vehicles. Then, the vehicle's emergency action upon receiving an alert from the CLW is an emergency brake. The simulations ran in these results context were based on the CD-HIL, with a platoon simulation model running on Gazebo with the vehicle's V2V communications being done through real OBUs connected through the ROS-MK5 bridge into their correspondent vehicle node. To produce a realistic scenario, we performed 30 simulations.

Figure 5.28 showcases the scenario of FDS, with (bottom two figures) and without CLW (top two Figures). Upon the failure injection on the second vehicle, which prevents its controller from reducing speed, without CLW, the second vehicle crashes into the leader. However, the third vehicle can still stop in time at this speed, as no failure occurs in its controller. At the bottom, with CLW, the safety controller triggers the emergency brake safety action upon detection of its controller's inability to reduce speed. As a result, the vehicle does not crash into the leader.

A video showcasing the simulations for the three scenarios can be found at *https://youtu.be/UjFyRQbnGYo*.

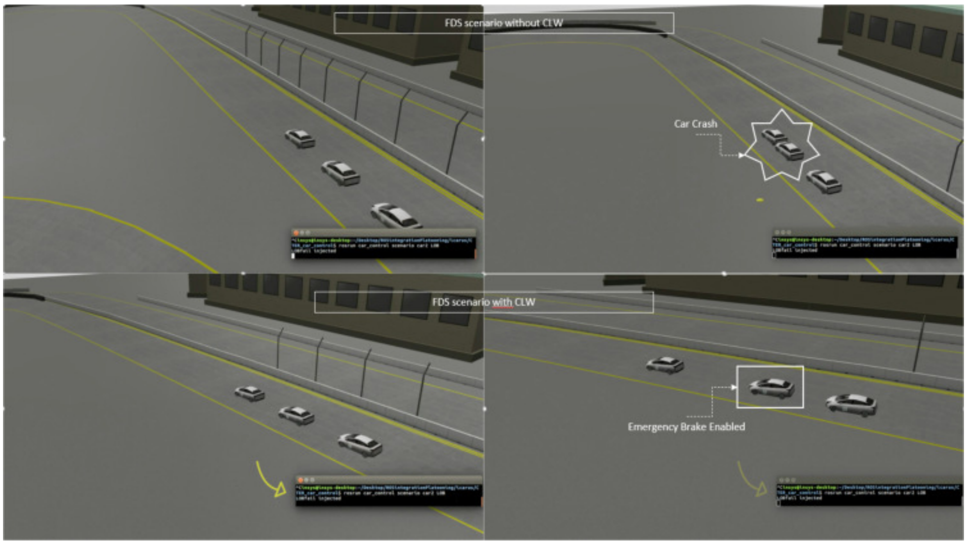


Figura 5.28: FDS scenario with and without CLW

Table 5.3 overviews some results gathered from the carried out tests of the CLW module within the previously stated scenarios. These tests have been running under the SafeCOP project to evaluate and guarantee some safety assurance metrics. Apart from ensuring the well-functioning of these modules, these results also assure the correct functioning of CD-HIL as a testing tool for C-ITS scenarios where the Cohda MK5 OBU is used. In this table, is it possible to observe that CLW avoids the collision between the vehicles in 86.66% of the situations. It means that the CLW increased the system's safety in case of catastrophic failures, as it intends to. In the remains, some crashes may occur if the loss happens when the vehicles are too close or when some communications problems happen, as pointed out in the table.

With the Co-VP controller, the CLW safety system, and the OBU integration validated over virtual scenarios, the next stage of CopaDrive aimed for increased system integration. So, the CopaDrive framework relies on CD-RoboCoPlat, a robotic testbed in which systems can be integrated similarly to the final prototype vehicles while keeping risks and costs at bay by deploying these platforms in controlled and smaller environments. The following section overviews this CopaDrive component and elaborates on the validation and demonstration of the Co-VP system we report on in this chapter.

5.6 CD-RoboCoPlat

There are several works on vehicle platooning. However, few instantiate their proposals over real hardware deployments. Therefore, this section focuses on practical implementation, particularly on robotic testbeds.

In contrast to previous works, as presented in section 4.4.2, our testbed provides clear advantages. For instance, it relies on ROS to enable new sensors and platforms integration, increasing its flexibility and reconfiguration options and integration with simulation software. Furthermore,

Tabela 5.3: HIL simulations - Analysis of CLW alert systems

Metrics	Results
Percentage of test runs where a crash between two or more platooning nodes occurred	13,3333%
Average number of communication failures detected per test run	0,2353
Average number of corrupted messages detected per test run	47
Average number of delivery delays detected per test run	0,2674
Percentage of test time where communications were performed under an acceptable level of latency	98,8889 %
Percentage of successful full stops after a catastrophic failure	86,6667 %
Downtime percentage of any SW component	1,2069 %
Average response time for warnings	30,004 ms
Average response time for non warnings	35,302 ms

this architecture allows the initial development of a control model in a simulator over a ROS environment and brings it to life in the robotic testbed in a comprehensive and continuous integration effort. Thus, it integrates a true communications OBU (ETSI ITS-G5), enabling the field trial of different communication scenarios in parallel with several cooperative control algorithms to evaluate its safety inter-dependencies better. Therefore, it is cheaper than any other deployment with real-size autonomous vehicles, as the number of cars can easily be increased. Finally, it is highly portable and can be easily deployed in a new indoor or outdoor environment in different track configurations.

The CD-RoboCoPlat is a 1/10 scale cooperative driving robotic testbed platform to support the deployment of different cooperative driving test scenarios in an indoor or outdoor environment. The objective is to showcase a higher system integration while supporting validation in platforms closer to an actual vehicle.

This platform enables the deduction of 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 current version of CD-RoboCoPlat with three cars (a leader and two followers) is presented in Figure 5.29, and the main components of each one of the vehicles are shown in table 5.4.

5.6.1 Testbed's Architecture

Each vehicle of our testbed is based on the F1tenth vehicle architecture [212], an open-source autonomous cyber-physical platform with some additional sensors. This high-performance autonomous vehicle architecture was designed to short-circuit the access to autonomous driving deployment and validation via an affordable vehicle solution with realistic dynamics, i.e., Ackermann steering and the ability to travel at high speeds, i.e., above 60 km/h. The car model used is a

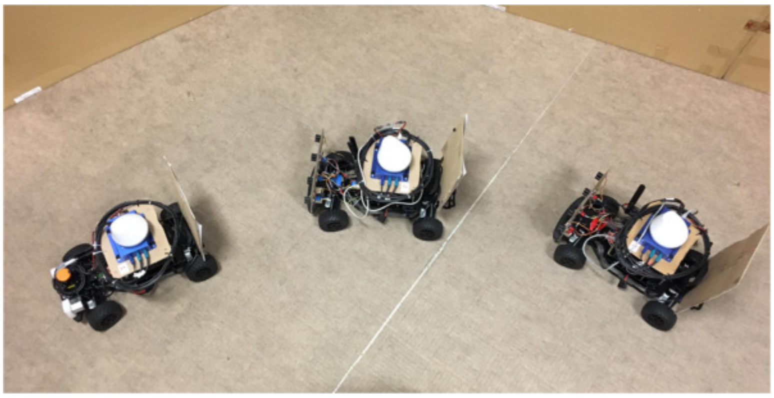


Figura 5.29: CD-RoboCoPlat

Tabela 5.4: CD-RoboCoplat Components

Components	Manufacturer	Model	Reference
Traxxas RC Car	Traxxas	Traxxas Fiesta ST Rally	[64]
Jetson	Nvidia	TX2	[140]
Teensy	Teensy	Teensy 3.2	[133]
Camera	Stereolabs	Zed Stereo camera	[264]
IMU	Sparkfun	9 DOF Razor	[261]
Sonar	Devantech	SRF08	[238]
Range Finder	Sharp	GP2Y0A02YK0F	[260]
OBU	Cohda	Cohda MK5	[48]

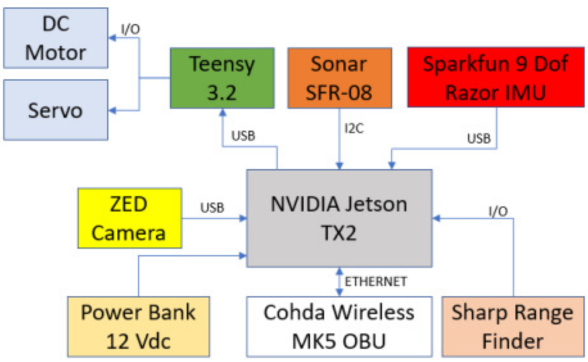


Figura 5.30: Hardware Architecture

Traxxas Fiesta ST Rally, a 1/10 scale of a real car. The versatility of the RC model allows it to be adjusted at will, creating a well-structured platform to test different scenarios.

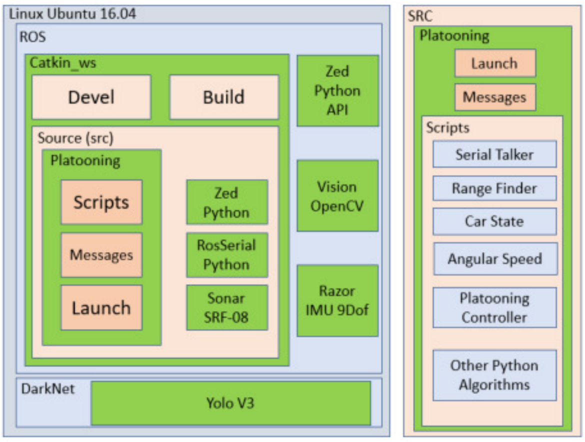


Figura 5.31: Software Architecture

The CD-RoboCoPlat architecture is presented in Figure 5.30. This architecture is replicated among all the vehicles, except for the first one, the leader. The leader also features a Lidar for enabling improved SLAM capabilities. In this architecture, the central component is the Nvidia Jetson TX2 [140], which 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 an 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 does not provide a direct interface to the vehicle's motor and servo, we set up 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. The communication between the vehicles is by the Cohda Wireless MK5 OBU via an ethernet connection to the Jetson TX2.

The operating system running on the Jetson TX2 is Linux Ubuntu 16.04.6 Xenial. The ROS-based system implements the processing pipeline to execute 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 later used to enable visual odometry. This architecture is presented in Figure 5.31, where the SRC container has the key nodes designed to control the movement of the vehicles. *Serial Talker* and *Range Finder* nodes provide the communication with the peripherals. In contrast, the *Car State* node collects the data from the sensors and computes the position of the vehicle. There is a specific node that calculates the car's angular speed, 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.

The controller is one of the components of the software architecture presented in Figure 5.31. As a multi-use testbed, the controller of CD-RoboCoPlat can be replaced by any other control model in future works. The OBUs perform the communication between the vehicles. Each MK5 OBU in vehicles transmits data to other OBUs in a defined range through broadcast messages.

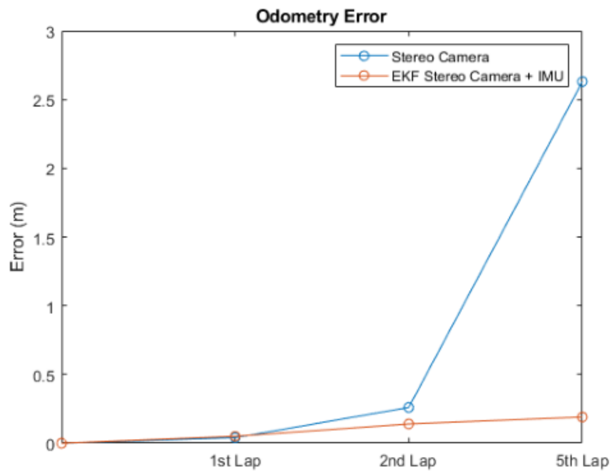


Figura 5.32: Odometry Error

5.6.2 Localization

Even that the integrated OBUs feature an embedded GPS module, the main objective was to deploy the testbed in a controlled indoor environment without requiring external localization sources for the vehicle. Thus, 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 can repeat the leader's trajectory. Although the leader is fitted with a Hokuyo10LX Lidar for improved performance, to keep the cost of the remaining vehicles low, we decided not to implement Lidar on the follower vehicles. Instead, the current version of RoboCoPlat uses the ZED stereo camera [264] visual odometry and an IMU [135] for detecting vehicle position and speed. 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 solution's effectiveness. At the end of each lap, the vehicle's current position was compared with the position indicated by the 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 5.32 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.

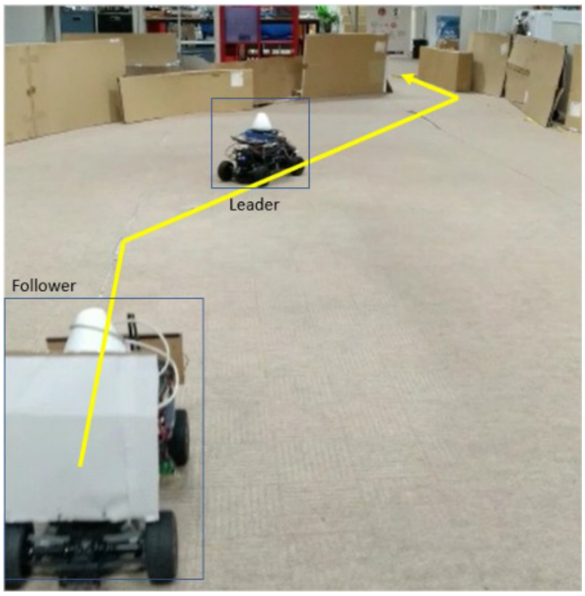


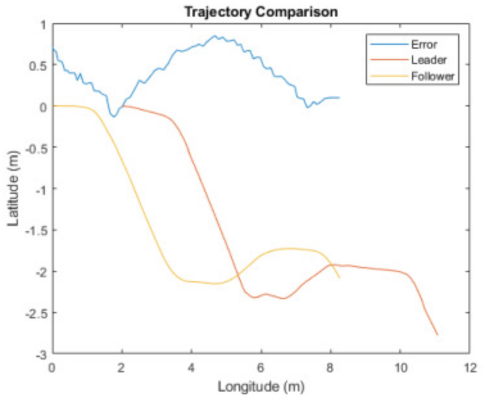
Figura 5.33: *Platoon's path*

5.6.3 System Validation

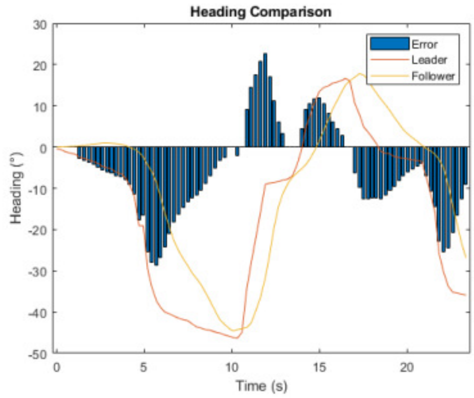
The overall cooperative platooning system is outlined and demonstrated in Figure 5.33. The local leader travels in a designed way at a constant speed of 1.0 m/s, while continuously providing to the followers, via the OBUs, relevant information such as linear and angular position, speed, and steering angle as presented in Figure 3.3. The follower receives this information and uses it to adjust its longitudinal and lateral motion, keeping a safe distance from the leader. The target distance between the vehicles is set as 3.0m, with the initial space defined as 2.0m. The global position of the cars is defined in terms of Cartesian coordinates that represent their latitude and longitude.

The control system was set up to carry out synchronized distance and orientation adjustments to analyze the system's response and precision. It means that the follower vehicle follows its leader by maintaining the target distance and applying the necessary lateral corrections to mimic the behavior of the preceding vehicle in terms of orientation and speed. Figure 5.34a depicts the path traveled by the leader and the follower (car_2). It is possible to observe that the follower keeps a distance from the leader, adjusting its position with an average error close to 0.5m. The distance between them is defined as the Euclidian distance between the global position of the vehicles in time t. The measured error is the difference between the desired and measured distances. 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. A video with the full demonstration can be seen at https://youtu.be/I6xWyMSyKwM.

The leader also performs an "S" movement, and the follower can repeat the action with minor differences. For example, figure 5.34b presents the comparison between the leader and the follower's heading values. Here, it is possible to observe that the follower performed the orien-



(a) Platoon's Trajectory Comparison



(b) Comparison of the Leader and Follower's Heading Values over time.

Figura 5.34: Trajectory and Heading comparison

tation adjustments roughly 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 data originated is reached.

5.7 Conclusions

This chapter presented CopaDrive, an integrated framework for developing safety-critical cooperative driving applications. This framework uses ROS as an enabler and integrator of three tools, CD-S, CD-HIL, and the CD-RoboCoPlat. Starting from a pure simulation environment, encompassing both communications and control perspectives of the application, CopaDrive moves on into test and validation stages with increased integration of system components, culminating on a robotic testbed. Most of all, the systems can be integrated, validated, and demonstrated.

By relying on ROS to integrate the different stages of CopaDrive, from the initial simulation tools until the testbed implementations, we significantly increase the system's modularity and reuse the development effort from the previous stages towards an increasingly validated and integrated approach.

Shortly, regarding CopaDrive, we hope to improve user friendliness by developing integrated user interfaces that can ease the simulation setup and result retrieval, which is still not an automated process. We also intend to apply this framework to the drone development and validation process to validate similar cooperative systems, such as handover communication systems between drones and control stations.

Meanwhile, we hope to mature the developed solutions further and ultimately test them in actual vehicles, thus targeting the automotive sector OEMs as potential customers for the safety solutions. To increase the maturity level, several steps will be taken. First of all, additional communication channels will be considered for the cooperative functions. In addition, we will carry out the inclusion of different monitoring variables in the CLW. Safety information can be improved with access to infrastructure information and by allowing other vehicles, apart from the platooning vehicles, to receive the warnings and the infrastructure. Thus future work will collaborate with road authorities and municipalities to develop and deploy pilots related to safety information in urban areas, for example, by incorporating the CLW warnings into the Other Hazardous Location Notification (OHLN) day 1 C-ITS service.

Certainly, CopaDrive will play a fundamental role in supporting such work by harnessing the ROS middleware integration's flexibility, modularity, and power.

Capítulo 6

An Integrated Look Ahead Controller for Co-VP

This chapter presents the Look-Ahead PID controller for Co-VP systems, integrating the longitudinal and lateral controls of the platoon. Here we show how the controller was developed, including the safety metrics adopted and how the platoon performance was improved based on these metrics. We also present an adjustment of the vehicle's lateral control, responsible for minimizing the "cutting corner"effect in platooning implementations.

6.1 Introduction

The Co-VP controller integrates several control areas. For instance, the error amplification and disturbance propagation in a platoon is studied in [21], where the authors analyze the problem of controlling a string of vehicles moving in a straight line. This study shows that the disturbance is propagated through the platoon, causing instability in the spacing error even with a constant speed. Another critical challenge is managing the *cutting corner problem* [25], where the vehicles have the same orientation but do not follow the leader's trajectory. Thus, the Co-VP curve's performance is critical.

Interestingly, several Co-VP control models do not address lateral control, and those that do ignore the advantage of relying upon V2X communications. The lateral controller is also known as the heading controller. A compromise is found in work presented in [307] by proposing an integrated longitudinal and lateral controller which integrates an on-board radar sensor with V2V communication. In this case, the heading controller is performed by a path estimation algorithm based on a linear time-varying model predictive control (LTV-MPC).

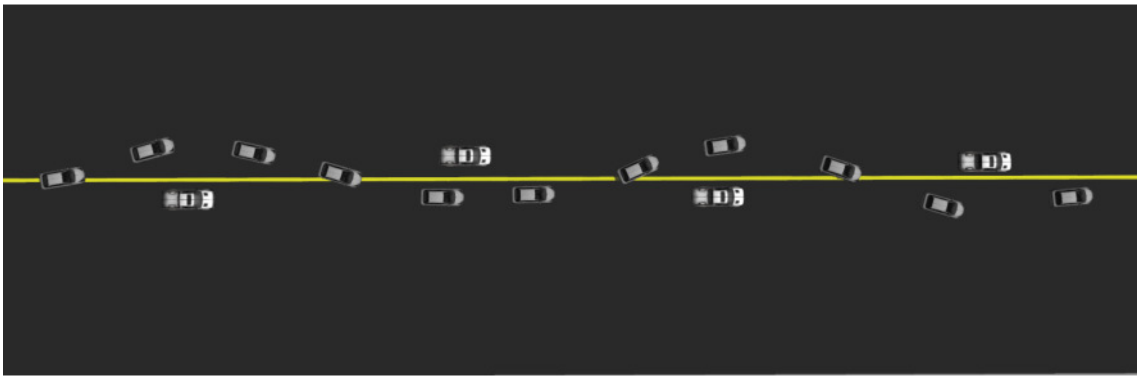


Figura 6.1: Platooning View

In [156], the authors propose an integrated lateral and longitudinal controller using the preceding vehicle acceleration, keeping the platooning safety with three main controllers. These controllers are a feed-forward controller for the string stable longitudinal control, a Corrective constraint controller, and an MPC controller for the lateral problem. However, the error propagation through the platoon and the cutting corner problem is not addressed. A solution to it was proposed in [26] with a Look Ahead Controller (LAC). The controller estimates a trajectory between each leader trajectory point in this work. However, there is a lack of research focusing solely on V2V communications to accomplish Co-VP control. This possibility is becoming increasingly scalable and viable with the advent of 5G integrated communications. It can be helpful, particularly in scenarios where vehicle sensors can become impaired and provide incorrect readings, providing an extra layer of safety. In addition, relying upon V2V makes these applications more flexible and cheap, as they are not so dependent on expensive vehicle sensors. However, to enable such an approach, more research is needed to fully understand its potential and limitations, such as the impact of the network Quality of Service (QoS) on the Co-VP safety and performance. Nevertheless, to support such research, one needs to rely upon functional cooperative control models enabled by V2V communications in the first place.

In this chapter, we propose a V2V-enabled Co-VP Look Ahead Controller (LAC) with low complexity that can keep the platoon's distance, alignment, and safety, reducing the impacts of the errors through the platoon solving the cutting corner problem. Using a well-known base controller as a PID reduces the system complexity, increasing the system implementation capability in real-life scenarios. The main simulator view can be observed in Fig. 6.1. The main contributions proposed in this chapter are: (1) The development of a longitudinal and lateral Co-VP controller that relies only upon V2V communications; (2) Improvement of the lateral controller to solve the cutting corner problem; (3) The development of a LAC strategy to increase the platoon's stability even with a large number of vehicles, reducing the disturbance propagation problem, presented in [246]; and (4) a safety analysis of the Co-VP controller in a realistic scenario, with trajectory changes of the leader and obstacle avoidance. All the scenarios rely upon a robotics simulator, demonstrating that this controller and proposed mechanisms can be implemented in reality.

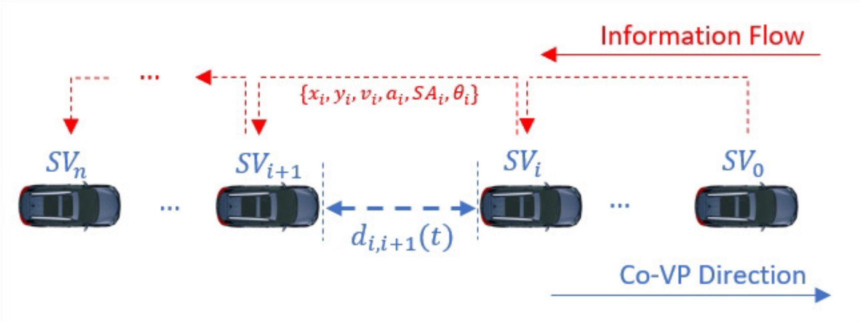


Figura 6.2: Co-VP Model with PF-IFT

6.2 Problem Formulation

6.2.1 Control Model

We assume a platoon of $n+1$ vehicles under a ETSI ITS-G5 communication environment [84] using a Predecessor-Follower IFT (PF-IFT) [150, 179], as presented in Fig. 6.2. To facilitate the understanding of the formulation proposed here, the table 6.1 presents the main terms used and their nomenclatures. Each vehicle is modeled as unicycle in a Cartesian coordinate system. The vehicles in the platoon are referred to as *subject vehicles* and identified by SV_i (where $i \in \{0 \leq i \leq n, i \in \mathbb{N}\}$), with SV_0 being the platoon leader. Each SV_i can be both a local leader of SV_{i+1} and a follower of SV_{i-1}. The platoon's model is exemplified in Fig. 6.3.

In this Co-VP model, each follower (SV_{i+1}) decides their behavior based solely on the CAM messages ($m_{i,i+1}(t)$) received from SV_i, transmitted upon activation of the kinematic triggers. Each SV_{i+1} receive data from SV_i, containing: the global position of SV_i - $(x_i(t), y_i(t))$, speed ($v_i(t)$), acceleration ($a_i(t)$), steering angle ($\alpha_i(t)$) and Heading ($\theta_i(t)$). The SV_{i+1} should gather it's own orientation, position, speed, acceleration and steering angle from internal sensors.

The inter vehicles spacing methodology is the *constant time-headway policy* (CTHP) [57], that uses the current speed of the vehicle to define the safety distance SD. In CTHP, the objective

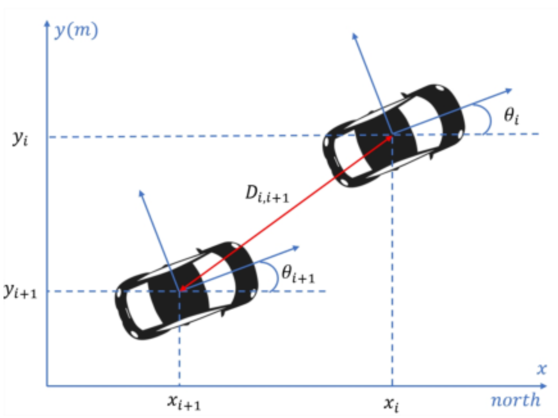


Figura 6.3: Platoon Model

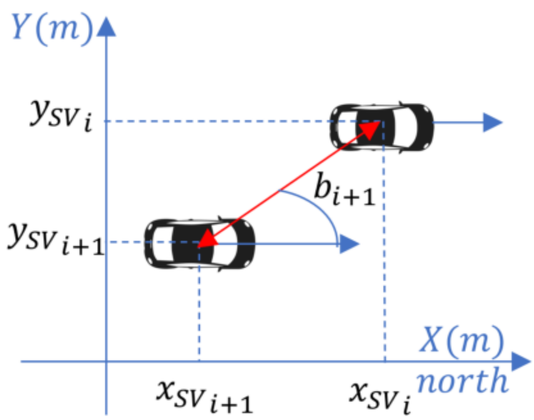


Figura 6.4: Bearing Error

range (d_{ref}) is $d_{ref}(t) = SD + T_h v_i(t)$, where is T_h is the defined time headway (typically 0.5 and 2 seconds), and $v_{i+1}(t)$ is the followers speed.

Tabela 6.1: Definition Terms

Abreviation	**Meaning**
i	Vehicle Identification
SV_i	Subject Vehicle i
$m_{i,i+1}$	Message from SV_i to SV_{i+1}
$D_{SV_i,SV_{i+1}}$	Inter-Vehicles Distance
d_{ref}	Objective Distance
SD	Safety Distance
e_i^{δ}	Distance Error
e_i^{θ}	Heading Error
b_i	Bearing Error
α_i	Steering Angle
be_i^{θ}	Lateral Error (with Bearing)

6.2.2 Metrics of Platoon Safety

Distance error (or Longitudinal Error) e_i^{δ}: The platoon stability is defined as the spacing error between the real and the desired inter-vehicle spacing [246]. The spacing error between SV_i and SV_{i+1} is defined simply as

$$e_i^{\delta}(t) = d_{i,i+1}(t) - d_{ref}, \tag{6.1}$$

where $d_{i,i+1}(t)$ is the Euclidean distance between $SV_i(t)$ and $SV_{i+1}(t)$.

Stability: We consider the **local platoon stability**, described by the following transfer function of the steady-state error:

$$H(s) = e_{i+1}^{\delta}/e_i^{\delta},, \tag{6.2}$$

based on the $\mathscr{L}_2$ norms. The platoon stability is guaranteed if $\|H(s))\|_{\infty} \leq 1$ and $h(t) > 0$, where $h(t)$ is the impulse response corresponding to $H(s)$ [341]. Alternatively, the string stability can be defined as $\mathscr{L}_{\infty}$, in order to to guarantee the absence of overshoot for a signal while it propagates throughout the platoon. This performance metric is the same as characterized in [333], which defines the worst case performance in the sense of measuring the peak magnitude of the spacing distance between the vehicles, defining a *global platoon stability*.

Heading Error (or Lateral Error) e_i^{θ}**:**

In the platooning application, the SV_{i+1} should perform the same path as the SV_i, based only on the received information (in the form of a message $m_{i,i+1}$). In a longitudinal-only Co-VP application, the controller of vehicle SV_{i+1} uses $m_{i,i+1}(t)$ to define and correct $e_{i+1}^{\delta}(t)$.

However, if longitudinal and lateral control is considered, vehicle SV_{i+1} compares $\theta_{i+1}(t)$ with $\theta_i(t - T_0)$, where T_0 is the time that SV_{i+1} takes to reach a similar position to SV_i when $m_{i,i+1}(t - T_0)$

was sent. This means that, as $d_{ref}(t) \geq SD$, when SV_i is in position $(x_i(t), y_i(t))$, SV_{i+1} is in position $(x_{i+1}(t), y_{i+1}(t))$, with a speed of $(v_{i+1}(t)\cos(\theta_{i+1}(t)), v_{i+1}(t)\sin(\theta_{i+1}(t)))$, the $m_{i,i+1}(t)$, sent by SV_i, will be received by SV_{i+1} in $t+\zeta$, where ζ is the message delay between the sent and receiving time. The information contained in $m_{i,i+1}(t)$ will be immediately used to calculate $e^{\delta}_{i+1}(t)$, while $m_{i,i+1}(t-T_0)$ will be used to calculate the **heading error** $(e^{\theta}_{i+1}(t))$, defined as:

$$e^{\theta}_{i+1}(t) = \theta_i(t-T_0) - \theta_{i+1}(t) \tag{6.3}$$

The actuation time over the $\alpha_{i+1}(t)$, caused by $e^{\theta}_i(t)$, is responsible for the cutting corner error. This effect is a consequence of the difference between $(x_i(t-T_0), y_i(t-T_0))$ and $(x_{i+1}(t), y_{i+1}(t))$. So, this difference is responsible for a bad alignment between SV_i and SV_{i+1}, even with a $e^{\theta}_i(t) = 0$, given that the follower can start to perform a curve at a different position, compared with the leader. This bad alignment is called *bearing error*, $B_i(t)$, and is illustrated in Fig. 6.4. The bearing error rises from accumulated heading errors of SV_{i+1} while following SV_i, particularly in curves, and should be calculated when $e^{\theta}_i(t) \approx 0$. In our work, we defined this threshold as $0.15rad$. This limit indicates that the desired SV position is ahead of the current SV_i position, at a maximum angle of up to 16°. This value prevents Bearing performance from causing a correction beyond the vehicle's limits, causing instability in sharp turns. The Bearing error is defined as:

$$b_i(t) = \arctan\left(\frac{x_i - x_{i+1}}{y_i - y_{i+1}}\right). \tag{6.4}$$

6.3 Controller Model

We divide the implemented control methods for the SVs into Longitudinal and Lateral controllers. Both were defined with a low complexity PID controller model. This chapter also proposes a LAC that modifies both longitudinal and lateral controllers, increasing the platoon safety.

6.3.1 Longitudinal and Lateral Controllers

The longitudinal controller ensures the platoon' safety, maintaining the inter distance between SV_i and SV_{i+1}, adjusting v_{i+1}. The main PID controller equation for $SV_i(t)$ is:

$$v_{i+1}(t) = K_P * e^{\delta}_{i+1}(t) + K_I * \int e^{\delta}_{i+1}(t)dt + K_D * \frac{\Delta e^{\delta}_{i+1}(t)}{dt}, \tag{6.5}$$

where K_P, K_I and K_D denote respectively the Proportional, Integrative and Derivative gain constants. The full controller is presented in Fig. 6.5, where we assume that the time constant of the actuator is much bigger than the time constant of the motor.

The lateral controller actuate over the vehicle's heading. The main equation of the heading PID controller is presented in eq. 6.6.

$$\alpha_{i+1}(t) = K^{\theta}_P * be^{\theta}_{i+1}(t) + K^{\theta}_I * \int be^{\theta}_{i+1}(t)dt + K^{\theta}_D * \frac{\Delta be^{\theta}_{i+1}(t)}{dt}, \tag{6.6}$$

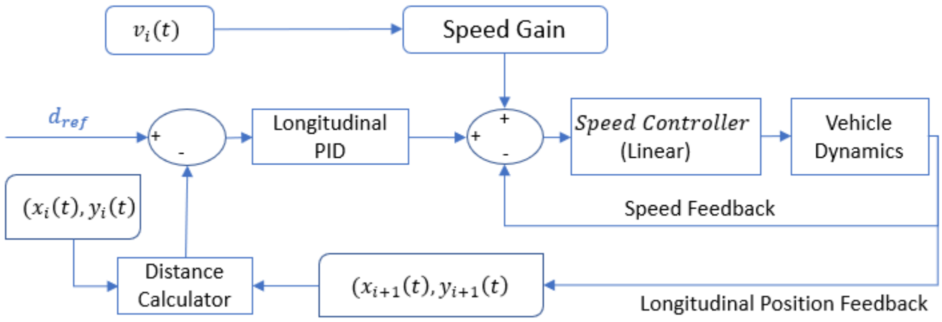


Figura 6.5: Longitudinal Controller Model

where K_P^θ, K_I^θ and K_D^θ denote the Proportional, Integrator, and Derivative gain constants of the lateral controller, and α_{i+1} is the *Steering Wheel Angle* to be applied to the vehicle. Finally, $be_{i+1}^\theta(t)$ is defined in (6.7) and depends of the bearing adjust actuation.

$$be_{i+1}^\theta(t) = \begin{cases} e_{i+1}^\theta(t) + B_{i+1}(t), & e_{i+1}^\theta(t) \leq 0.15 \\ e_{i+1}^\theta(t), & e_{i+1}^\theta(t) > 0.15 \end{cases} \tag{6.7}$$

6.3.2 Look Ahead Controller - LAC

The PID controller is typically reactive, and thus, when facing an abrupt setpoint change, the adjustment can saturate the actuator and cause oscillations or instability. In the Co-VP, this effect is observed particularly after closed curves and in quick re-adjustments of speed with a cumulative effect throughout the platoon. In many situations, this effect is reduced given the nature of the test track, mainly when using only long straight roads with few curves. However, in a more realistic scenario, the oscillations of the platooning can cause instability and decrease the system's safety.

The proposed LAC adds an error information about $SV_i, i > 0$ in the controller of SV_{i+1}. This information is transmitted to SV_{i+1} to reduce the disturbance propagation, allowing SV_{i+1} to compare its position with SV_{i-1} position, keeping the main reference in the SV_i. This approach also avoids the need for the leader to send messages to all platoon cars, which allows the platoon to have an increased size.

As demonstrated in [26], analyzing the platooning, the disadvantage of the common LAC is that the SV_{i+1} lateral position is correct only in a straight line, compared with SV_i. This disadvantage leads the system to the cutting corner problem since there is no information about the leader's trajectory. There is also the increasing error provided by the difference between the current position of SV_{i+1} and the desired position, provided by the path performed by SV_i. Assuming that this error exist, are greater then 0 and are denoted by $e_{i+1}^\delta(t)$ and $e_{i+1}^\theta(t)$, the error between the SV_i and SV_{i+2} increases in each curve. So, the proposed look ahead incorporates the difference between the current position of SV_{i+1} and its desired position, increasing the correction to be performed by

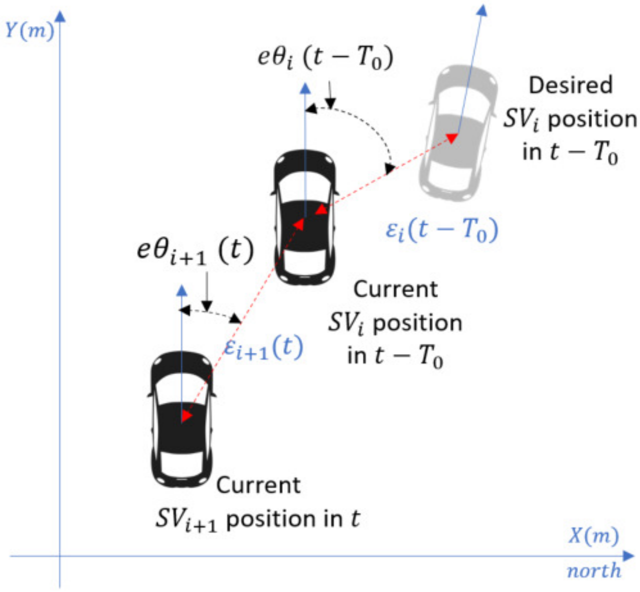


Figura 6.6: The LAC consider the difference between the current position of SV_{i-1} and the desired position

SV_{i+2}. The LAC reduces the difference between the path provided by the platoon's leader and the rest of the followers, as depicted in fig. 6.6. The new errors can be defined, $\forall SV_i, i > 1$, as:

$$e^{\delta}_{i+1}(t) = e^{\delta}_{i+1}(t) + e^{\delta}_{i}(t - T_0) \tag{6.8}$$

$$e^{\theta}_{i+1}(t) = e^{\theta}_{i+1}(t) + e^{\theta}_{i}(t - T_0) \tag{6.9}$$

6.4 Simulation Environment

In this work, V2V communication is simulated using the ROS topics, as presented in Fig. 6.8, in a Linux Ubuntu 18.04.6 Bionic, with Gazebo 9.0 and ROS Melodic. The running PC has a Intel® Core® i7-975H CPU, with 16 MB RAM memory and a NVIDIA Geforce GTX 1650. Every vehicle in the platoon publishes its own information in the *car$_i$/TXNetwork* in a frequency of $33Hz$ - the maximum frequency proposed in [84]. The simulator full architecture was defined in section 5.4.

Those topics are all republished by a ROS topic *Network_Simulation* in another topic called *car$_i$/RXNetwork*. So, the SV_i subscribes to the respectively *car$_i$/RXNetwork* topic and performs the defined control actions. As the proposed V2V communication uses a broadcast model, every vehicle receives all the data from other vehicles in the network but only uses the corresponding SV_{i-1}.

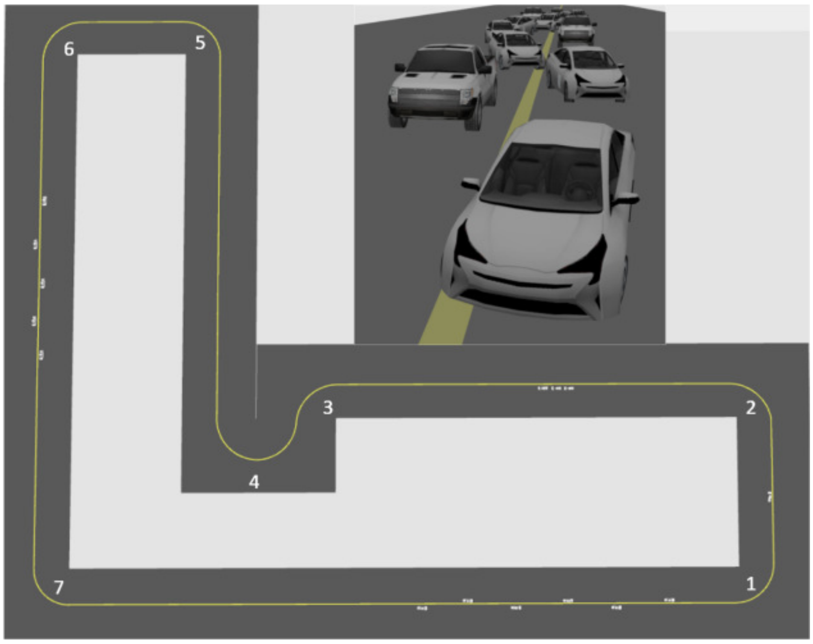


Figura 6.7: City Circuit

6.5 Simulation Scenarios

To evaluate the proposed controllers, four scenarios were designed, with two circuits - an Oval and a City circuit, presented in Fig. 6.7. Scenarios 01 and 02 present each controller feature, namely the bearing controller and the LAC, comparing the platoon safety performance with and without these controllers. Scenarios 03 and 04 present a more realistic situation in a city circuit with and without obstacles. All the scenarios represent a full lap in the designed road, finishing with a braking action by the platoon leader, without any crash, as visible in the video presented in *https://youtu.be/Gjpg-yV0tDc*. The principal scenarios parameters are shown in table 6.2:

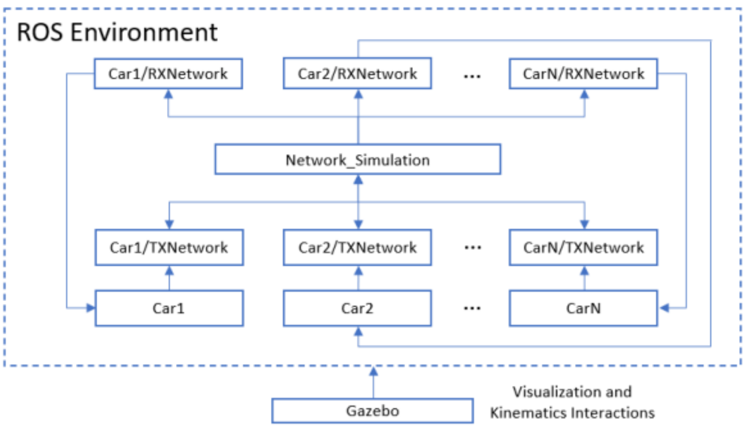


Figura 6.8: Simulation Architecture

Tabela 6.2: Model Parameters

Parameters	Definition
Vehicles	4 to 11
Max Steering	0.52 rad
Safety Distance (SD)	5.5 m
Time Headway (T_H)	0.5 s
Leader Speed	50 Km/h
Longitudinal: K_P, K_I, K_D	2.0 , 0.005 , 2.0
Lateral: K_P, K_I, K_D	2.5 , 0.001 , 1.0
Time between Messages	0.03 s

6.5.1 Scenario 01 - Bearing test

The first scenario was designed to test the bearing adjustment of the lateral controller in an oval circuit. We performed a full lap with 4 SVs without the bearing controller. The vehicles' paths in each test are presented in Fig. 6.9. This figure illustrates that even with a similar trajectory, the cars without the bearing controller do not follow the same path in some circuit parts. This error occurs when e_i^θ is near zero, even with the cars in the wrong alignment. The error is reduced in the curvature sections but increases afterward.

With the Bearing controller correction, however, results are much better, and the path of SVs is very close to the one performed by the leader. We can compare the average distance error for SV_3 in each test to evaluate this performance. Without the bearing controller, $e_3^\delta = 0.9863m$, while using the bearing controller, this error was reduced to $e_3^\delta = 0.4931m$, that indicates 50.01% improvement. These values are obtained from Fig. 6.10, where we compare the vehicle's position at the lap's end. This figure shows that the heading of the first three vehicles have the same longitude and differs only by the latitude, while $e_i^\delta \approx 0$, in both tests. In this case, the bearing controller is responsible for adjusting the platooning alignment, reducing the distance error at that point.

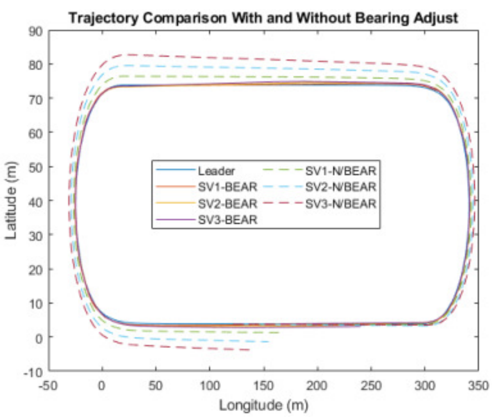


Figura 6.9: Scenario 01 - Bearing Test

Control Model	Car	Long. (m)	Lat. (m)	Head. (rad)	Dist. Error (m)	Head. Error (rad)
Without Bearing Control	Leader	150.90	73.81	-1.57	0.00	0.00
	SV1	150.86	76.20	-1.57	2.39	0.00
	SV2	150.89	78.78	-1.58	2.58	0.01
	SV3	150.94	81.34	-1.58	2.56	0.00
With Bearing Control	Leader	150.97	73.81	-1.57	0.00	0.00
	SV1	150.86	73.94	-1.57	0.17	0.00
	SV2	150.98	74.23	-1.56	0.32	0.01
	SV3	150.95	74.66	-1.56	0.43	0.00

Figura 6.10: SVs Position with/without bearing Controller

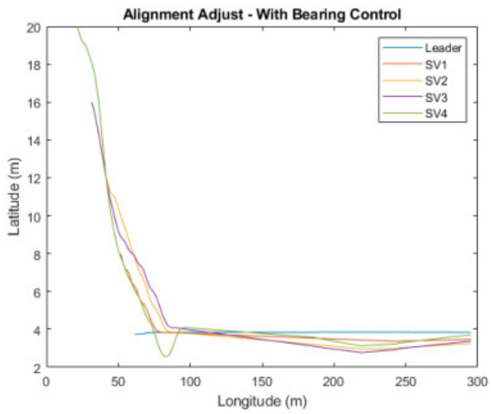


Figura 6.11: Scenario 01 - Longitudinal alignment

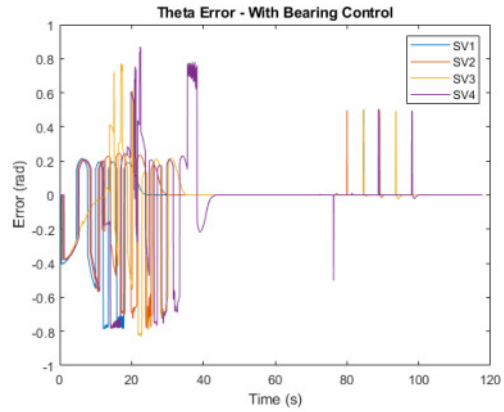


Figura 6.12: Scenario 01 - Heading Alignment

To test the bearing adjust in a more complex condition, we started a five-vehicle platoon with the same heading ($e_i^{\theta}(t)$=0) but with a wrong alignment. Figs. 6.11 and 6.12 illustrate the path performed by the vehicles and the variation of e_i^{θ}. This simulation shows the performance of the bearing adjusted to align the vehicles in complex situations and demonstrates that this model may be used in a joining's platoon condition.

6.5.2 Scenario 02 - LAC

We carried out several laps with a nine-vehicle Co-VP without the LAC to evaluate the LAC performance. Then, we rebuilt the test using the LAC with an eleven-vehicle platoon. Figure 6.13 compares the vehicle's trajectory in both situations during one lap. Without the LAC, the SVs could follow the leader, but with oscillations in the tail vehicles, namely in SV_6, SV_7, and SV_8. This oscillation is caused by the backpropagation of the leader's trajectories adjustments, which increases considerably with the number of vehicles in the platoon. The LAC, as depicted in fig.

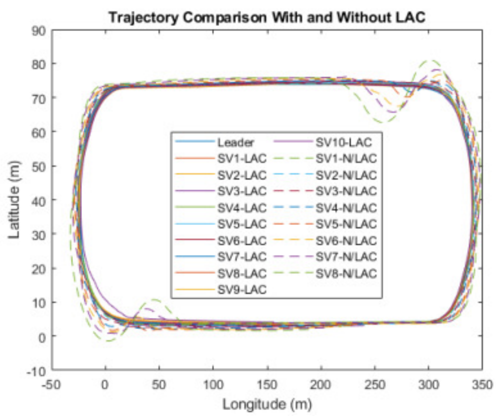


Figura 6.13: Scenario 02 - Look Ahead Controller (LAC)

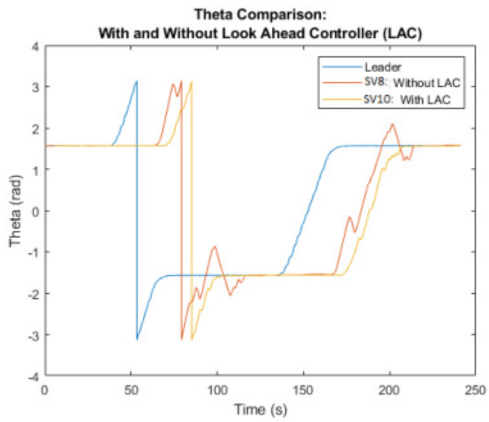


Figura 6.14: Scenario 02 - Platooning curve performance

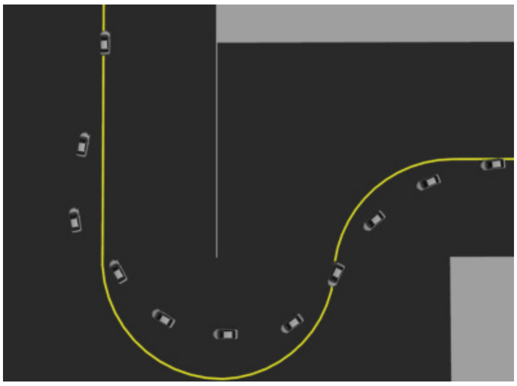


Figura 6.15: Scenario 03 - Platooning curve performance

6.13, deals with this problem, reducing the error propagation throughout the platoon and reducing the oscillation in the vehicles' trajectory. The LAC performance can be demonstrated considering the average distance error ($AVG(e_i^\delta)$) in SV_8. Without the LAC, $AVG(e_8^\delta) = 1.523m$, while using LAC, this error is reduced to $AVG(e_8^\delta) = 0.7079m$. Another improvement provided by the LAC implementation is the possibility to increase the platoon size, given that the $AVG(e_{10}^\delta) = 1.207m$, also reducing the error from the last vehicle in the test without the LAC in 16%.

Compared to the leader, the last vehicles' heading adjust test also demonstrates the improvement provided by the LAC, as presented in Fig. 6.14, with a comparison between the SV_8 in the test without LAC and SV_{10} using the LAC. It shows that SV_{10} heading adjusts are more smooth and have less oscillation in comparison with SV_8.

6.5.3 Scenario 03 - Complex Circuit

The Co-VP is well defined and largely analysed in long straight roads, with easy or no curvature. However, with harder curvature, more complex scenarios can cause oscillation and even

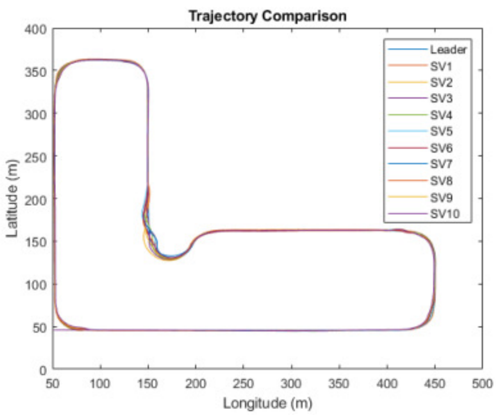


Figura 6.16: Scenario 03 - Vehicles Path

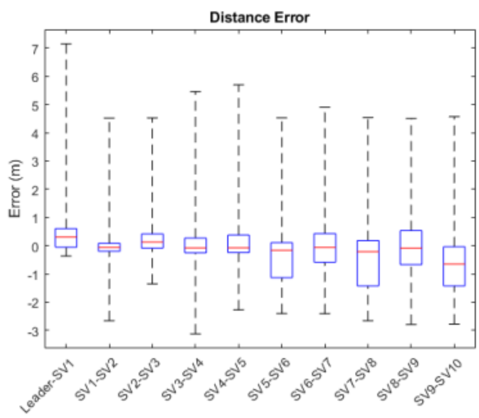


Figura 6.17: Scenario 03 - Vehicles inter Distances

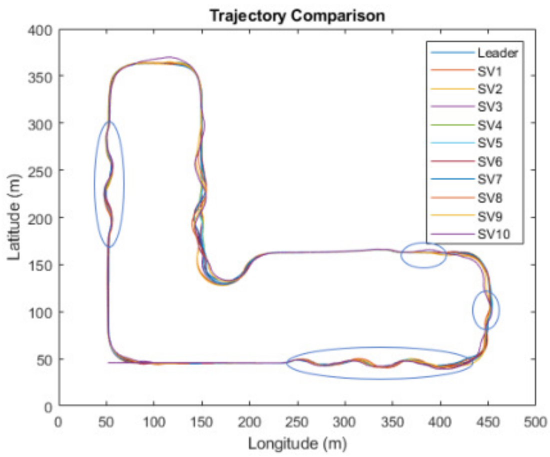


Figura 6.18: Scenario 04 - Vehicles Path

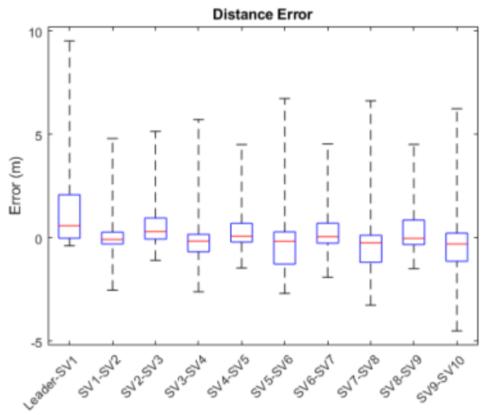


Figura 6.19: Scenario 04 - Vehicles Inter Distances

instability in many controllers, decreasing the platoon's safety. Then, using the longitudinal, lateral, bearing, and LAC to analyze our controller, we performed several laps with a platoon with eleven-vehicle Co-VP in the circuit of fig. 6.7, without obstacles. This circuit presents some interesting challenges, namely the different direction curves, straight sections and a quite hard bend, presented in Fig. 6.15.

The vehicle's trajectory is presented in Fig. 6.16 and demonstrated in the video in *https://youtu.be/Gjpg-yV0tDc*. All the SVs were able to closely copy the same path as of leader, with just a small oscillation in curve 4. Figure 6.17 shows the average error between the desired distance of SV_i and SV_{i+1} during the lap. As this distance never gets close to the defined SD, the platoon's safety is guaranteed, thus avoiding any collisions between the vehicles. Fig. 6.17 also demonstrates that the average error of the vehicle's distances is close to zero, although it varies in different situations, like curves. However, even with those changes, the errors are reduced after the curves. In the selected scenario, the local stability of the platoon cannot be guaranteed by the strict criteria proposed in 6.2, since $e_4^\delta / e_3^\delta > 1$, for instance. However, the *global* stability of the platoon can be guaranteed, since $\forall e_{i+1}^\delta / e_1^\delta < 1$, for $i > 1$.

6.5.4 Scenario 04 - Obstacle Avoidance

To further push the limits of the Co-VP controller, in this last scenario we included a slalom section. The leader uses sonars to avoid 14 vehicles distributed in the circuit, as presented in Fig. 6.7. The path carried out by the vehicles is depicted in Fig. 6.18, where the blue circles indicates the track obstacles. Then, all the SVs closely followed the leader's path, even with the many quick shifts in orientation. A presentation of this scenario is given in *https://youtu.be/4ysgAFnvWpI*. In Fig. 6.19, it is possible to observe that even with these imposed oscillations of the leader, the mean of the distance errors of the platoon vehicles is close to zero. However, it is also possible to observe that SV_{10} gets closer to SV_9, since the distance error increases in the negative way. This means that as the vehicles perform closed curves in sequence, the SV_{i+1} are getting closer tho the

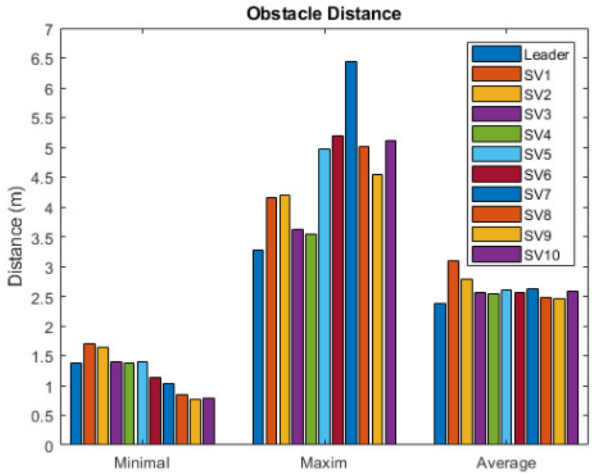


Figura 6.20: Scenario 04 - Obstacles Distances

SV_i, decreasing the platoon's safety. This occurs because the desired speed is constant. So, in a sequence of turns in different directions, while the SV_i linear speed decreases, SV_{i+1} linear speed is bigger, reducing the distance between the vehicles.

Since the leader reduces its linear speed in curves 3 and 4, the platoon propagates this effect. Even with this approximation, none of the vehicles have come near to the SD. It is also possible to observe that the global platoon stability was guaranteed. This scenario also demonstrates the importance of having the lateral controller working together with the bearing adjust mechanism, in order to avoid collisions.

In this scenario, the platooning performance can be evaluated by the distance that the vehicles pass from the obstacles. In this case, the safety distance to the obstacles was defined as $0.5m$. The minimal, maximum and average distance between the vehicles and the obstacles are show in Fig. 6.20. This one also demonstrates that the average distance between the vehicles in the platoon to the obstacles is almost the same, which shows the algorithm's efficiency and the vehicle's ability to follow the leader's path.

6.6 Conclusions

In this chapter, we presented a V2V-enabled Look Ahead PID controller, together with a method to reduce the disturbance propagation in the platoon. The proposed platooning controller also implements a solution to solve the cutting corner problem, keeping the platooning alignment. We evaluated the performance of these mechanisms over a robotics simulator, showing that this low complexity V2V-enabled Co-VP controller can be effectively implemented and its maintains its stability under several different and challenging scenarios.

Capítulo 7

Improving the Performance of Cooperative Platooning With Restricted Message Trigger Thresholds

This chapter presents a study on the performance of the cooperative platoon from further restriction of the triggers responsible for firing the CAM messages proposed in ETSI ITS-G5. By increasing the safety of the platoon movement, through the reduction of lateral and longitudinal errors, we prove the effectiveness of these triggers. We also show the impact of these changes on the network, indicating that the number of triggered messages does not increase appreciably, and thus can be used in real scenarios.

7.1 Introduction

In the absence of a dedicated communication protocol for sharing relevant platooning control information, vehicles can rely solely on received CAMs for setting up a platoon. In such cases, the CAM trigger thresholds play a fundamental role in keeping the platoon's lateral and longitudinal coherence and stability. In this chapter, we investigate the behavior of a platoon as we vary the threshold values of the CAM triggers over a selected range (organized into five candidate service profiles) in terms of efficiency and safety of the vehicles. Ultimately, from the five candidate service profiles, we select the one that performs best for the particular application of platooning and present it as the Platoon Service Profile (PSP) to be considered for integration into the standard. PSP improves the performance of the Co-VP lateral and longitudinal controllers in scenarios with complex trajectories, such as a sequence of curves or while overtaking obstacles, reducing the distance and heading errors and increasing the platoon's stability. Finally, we evaluate the network overhead incurred by this new profile in terms of throughput and inter-message delays. In addition, the Packet Delivery Rate (PDR) under PSP is also evaluated, specifically in congested network conditions. These results extend previous work [192, 294] by carrying out an integrated analysis of the control and networking perspective in four Co-VP scenarios.

The contributions of our work are the following:

1. We assess the impact of the standard event-based Service Profiles (SPs), in scenarios of abrupt maneuvers (acceleration, deceleration) and sequential curves involving obstacle avoidance, on the performance of the platoon's stability (distance and heading errors). This was done by relying on a high-fidelity analysis of Co-VP lateral and longitudinal dynamics over a six-vehicle Co-VP over a realistic simulation framework called CopaDrive [101].

2. We identified a set of scenarios where standard settings provided decreased performance and could compromise the platoon's safety. Hence, we explore new Service Profiles that can mitigate this problem with negligible implications to the standard. We assess these new SPs and show that they increase the Co-VP overall safety, reducing lateral and longitudinal errors in multiple scenarios.

3. The network performance was analyzed to evaluate the impact of a new profile in the occupation of the channel. We demonstrate that the proposed PSP does not significantly increase network usage even under a heavy traffic environment than the ETSI ITS-G5 profiles.

In this chapter, we will use the same platoon model and safety metrics presented in chapter 6.

7.2 Related Works

Communication Impact on Cooperative Platoon Performance: The impact of vehicular communications on the performance of autonomous vehicle applications has been extensively studied [65, 200, 148, 102], but not explicitly related to platooning applications. The works presented in [331, 330] analyze the performance of a cooperative platoon in a scenario of constant-time headway, with a multiple predecessor-follower IFT and a multiple preceding and following IFT including random packet losses. The authors determine the upper bound for communication delay for longitudinal control to guarantee platoon stability. However, this work does not address communication standards and the trigger conditions presented in realistic scenarios.

In [58], the authors investigate the impact of the packet loss ratio and time message delay in the Co-VP controllers, considering DSRC and LTE C-V2X networks. This work assumes a fixed inter-message delay and a packet loss model based on the Bernoulli distribution. The authors observed that longitudinal and lateral errors increase with message delay and packet loss and proposed a limit to both variables. However, the work does not investigate any scenario with both network conditions. A similar strategy was applied in [273] to evaluate the impact of a deliberate communication failure in one of the vehicles and the consequences on platooning stability. This work used a simulated 14-vehicle platooning with WAVE communication with fixed time delays.

In contrast with the usual steady-state communication analysis, the time-varying performance of IEEE 802.11p Co-VP communication is discussed in [312]. The authors consider the impact of a disturbance in the leader's behavior and derive the time-dependent states of the followers. Finally, the authors used the packet loss and the message delay as metrics, concluding that the IEEE 802.11p can keep the platooning stability under a disturbance. However, this work considers a leader-followers IFT, which reduces the number of sent messages. A similar evaluation was performed in [192], using the ETSI ITS-G5 standard and the leader predecessor-followers IFT. This work identifies the phenomena that decrease communication performance based on message synchronization after sequential disturbances in Co-VP speed.

Trigger Thresholds for Message Transmission: The performance of the cooperative platoon also depends on the message trigger strategy: time-triggered and event-triggered. Although the ETSI ITS-G5 defines the event-triggered strategy as a standard, many implementations have been performed with time-triggered strategies [244, 276, 196]. Time-triggered messages can increase platoon safety if a high message frequency ratio is used, at the cost of increased packet collision probability due to a crowded medium [59]. In turn, the event-trigger solution reduces the network channel busy rate (CBR), enhancing vehicular network dissemination performances [148].

The event-triggered communication studied can be divided into two groups. The first group, such as presented in [181], assumes that the V2V communication is fully reliable. The authors of [181] developed a framework for event-triggered coordination of nonlinear vehicle dynamics with general controllers and a lower limit inter-event time. The second group investigates the impact of the network instabilities in event-triggered platooning [178]. In [20], an event-triggered message control is defined for a Co-VP application with time-varying delay and sensor faults. The event-triggering mechanism is a function of the present value of the sensor faults (and not the parameters defined in the ITS-G5 standard). In [308], it is proposed a flexible event-triggering strategy based on tunable parameters for each platooning vehicle, reducing the communication burden.

An external observer is proposed in [324] creating a distributed and adaptive event-triggered control mechanism based on the estimation of the leader state matrix. However, none of the above studies leverages an active ITS communication standard, which distances their conclusions from real deployment scenarios. An evaluation of the delay between messages in a cooperative platoon is conducted with ETSI ITS-G5 in [296]. The authors compare message delay using the ETSI event-triggered specifications against a fixed frequency of $10Hz$. In both modes, the authors consider a random transmission delay. The authors conclude that the platoon performance at fixed frequency outperforms the one with the ITS-G5 standard, especially at higher speeds. However, they do not address the CBR and its effects on the platoon.

We claim that event-triggered solutions can offer satisfactory platoon safety and efficient performance while improving the medium capacity and reducing network congestion. The latter aspect is relevant on account of co-existence: the platoon's internal communications should not become so intense as to degrade the communications of the other road users around the platoon. While several works have studied event-triggered solutions, as far as we know, none have evaluated their performance under the ETSI ITS-G5 standard, particularly in a realistic approach, which encompasses both control and kinematic properties of vehicles alongside the communica-

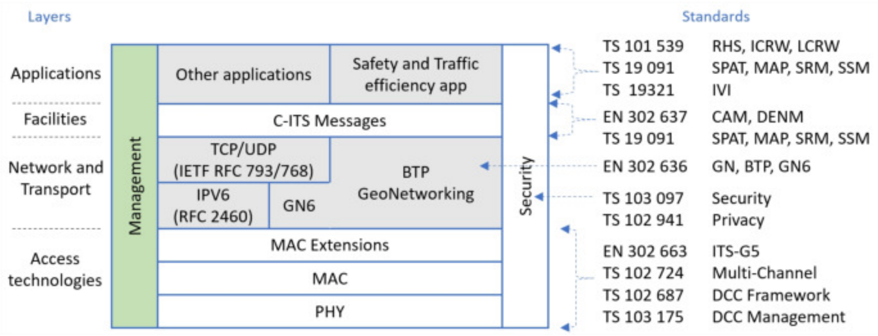


Figura 7.1: ETSI ITS-G5 Stack

tion aspects. This work fulfills this gap by presenting an in-depth analysis of the ETSI ITS-G5 event-triggered message transmission and a microscopic evaluation of the platoon's longitudinal and lateral error under representative road scenarios.

7.3 Exploring Trigger Thresholds for CAM Messages

7.3.1 Current ETSI ITS Triggers

The ETSI ITS stack is described as a family of ETSI standards, with the key one being ETSI EN 302.665 [73], as it describes the communication architecture. Fig. 7.1 (adapted from [95]) presents the protocol stack and reference architecture for ETSI ITS-S and lists the key standards of the European ITS standard. The Cooperative Awareness Messages (CAMs), defined in ETSI EN 302 637-2 [82], can be event-triggered or periodic. The CAM event-triggered mechanism is kinematic-dependent: each vehicle generates new CAMs depending on updates of its current position, speed, and heading [192]. In other words, an OBU adjusts the CAM message periodicity accordingly to the vehicle's dynamics. For example, it increases the frequency as it accelerates or decelerates, at high speeds, or when performing abrupt maneuvers.

Then, when detecting a variation in any of the parameters, the sensors must inform the controller so that it triggers a new CAM message. This action is critical in hard braking circumstances, where acceleration decreases quickly, and the platoon must be informed to avoid a collision. The same can be inferred in a quick change of direction, where the heading varies rapidly, and new messages are fired. So, the followers can correct their trajectories smoothly, allowing the route to be restored.

The CAM trigger threshold values are defined in [82]; for convenience, we refer to this set of values as the **Basic Service [*Transmission Trigger*] Profile** (BSP), following the nomenclature presented in chapter 5. The threshold values triggers are defined within an upper (T_{max}) and lower (T_{min}) messages bound times and kinematics triggers to check e^{δ}_{i+1} and e^{θ}_{i+1} comparing SV_{i+1} data with the received one from SV_i. These rules are checked latest every $100ms$, which is defined as $\Delta = T\_CheckCamGen$, and are stated as follows:

- Maximum time (T_{max}) interval between CAM generations: $1s$;

- Minimum time (T_{min}) interval between CAM generations: $0.1s$;
- Heading difference (τ_{Head}): absolute difference between current and last heading provided in a CAM; a CAM is triggered if $\tau_{\text{Head}} > 4°$;
- Position difference (τ_{Pos}): a CAM is triggered if $\tau_{\text{Pos}} > 4m$;
- Speed difference (τ_{Speed}): a CAM is triggered if $\tau_{\text{Speed}} > 0.5m/s$;

The timing existent during CAMs generation and processing, follow a timeline with a set of a couple of requirements defined at figure 7.2. In this figure, t_A represents the *acquisition time* from sensors, and t_D is the processing time of the information. Finally, t_C is the time between two consecutive CAM messages. According to the ETSI specification, t_A and t_D should be smaller than $50ms$.

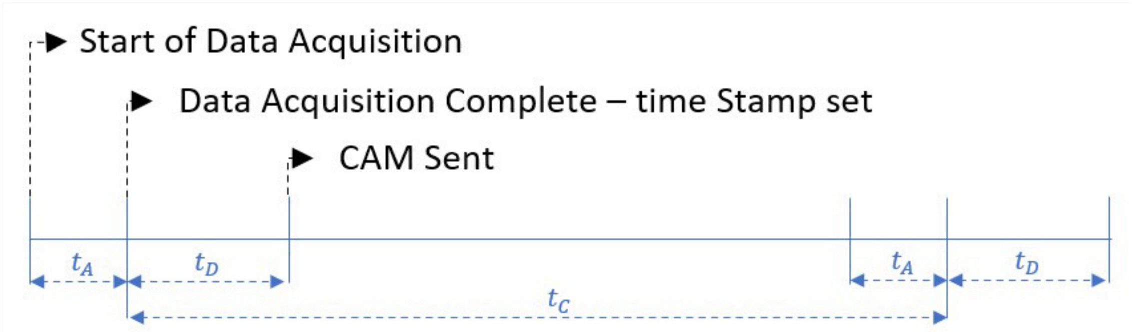


Figura 7.2: Time requirements for CAM generation and CAM processing

Thus, in BSP, assuming a straight-line trajectory, with a vehicle traveling at 12 m/s, the τ_{Pos} is activated at 3 Hz. If this speed is increased up to 15 m/s, the CAM trigger frequency becomes 3.75 Hz, and if increased up to 20 m/s, it becomes 5 Hz. This adaptation of the messaging system to vehicle conditions increases responsiveness to variations, although it also increases network traffic as the number of messages increases. The reverse also happens, with the messaging frequency decreasing with speed reduction and movement stabilization.

In [72], ETSI defines some ITS use cases, including a Co-VP situation, where it reduces T_{max} to $0.5ms$. This use case is defined as BSP for platooning (BSP-P) in [294]. This change causes that, in a straight line, a platoon member whose constant velocity is greater than or equal to $6.67m/s$ will transmit a CAM by trigger τ_{Speed} instead of T_{max}, unless another trigger is detected. The ETSI standard [82] also specifies that T_{max} of a kinematic trigger assumes the value elapsed between the last two CAMs ($T_{max} = t_{CAM}$), until one new trigger is fired or until three messages limited by the new T_{max} are sent. If all three messages are sent, and no trigger is activated, T_{max} returns to the original value defined in the communication profile.

7.3.2 Tentative Service Profiles

In related works, the performance of communication profiles specified in the ETSI ITS architecture has been evaluated in cooperative platooning conditions, typically following straight lines

Tabela 7.1: Service Profiles

Profile	*BSP*	BSP-P	SP_1	SP_2	SP_3	SP_4	SP_5
T_{max} (s)	1.0	0.5	1.0	1.0	1.0	1.0	1.0
T_{min} (s)	0.1	0.1	0.1	0.1	0.1	0.1	0.1
τ_{Head} $(^\circ)$	>4	>4	>2	>1	>4	>2	>1
τ_{Pos} (m)	<4	>4	>4	>4	>2	>2	>2
τ_{Speed} (m/s)				<0.5			

without obstacles. In such scenarios, platoon safety is evaluated solely on its ability to keep the distance between vehicles as the leader changes its speed.

However, a natural condition to analyze the performance of the Co-VP application suggests the presence of curves and obstacles on the track. Keeping speed constant, the trigger values τ_{Head} and τ_{Pos} must assume a more significant role in the controller response. Thus, we propose the analysis of different trigger values than the ones proposed by ITS-G5, defining five Service Profiles (SPs), as shown in Table 7.1. Considering that the value of τ_{Speed} in the ETSI ITS is already quite restrictive, we chose to analyze the impact of reducing the values of τ_{Head} and τ_{Pos} in these SP. So, initially, in SP_1 and SP_2, we reduced the τ_{Head}, respectively, to 2° and 1° to check their influence over the heading error. Nevertheless, in SP_3, SP_4, and SP_5, we changed τ_{Pos} to $2m$, reducing the maximum speed for triggering by t_{min} from $40m/s$ to $20m/s$. Thus, it is possible to analyze the network congestion caused by this condition by the increased messages concerning the expected increase in performance. For complete analysis, SP_3, SP_4, and SP_5 mirror the values of τ_{Head} from *BSP*, SP_1, and SP_2, respectively.

7.4 Evaluation of the Service Profiles

We use the CopaDrive framework [101], which integrates a 3D robotic simulator (Gazebo) with an ETSI ITS stack provided by the Artery project [234], running on the network simulator OMNET++. The communication between Gazebo and Artery is made through messages exchanged under Robotic Operating System (ROS). Within a hybrid architecture between events and real-time, OMNET++ waits for events generated in the Gazebo to perform the communication between the vehicles. Thus, the kinematic events that trigger messages are performed in ROS/-Gazebo and interpreted in OMNET++, which evaluates if the triggers should be activated and fire new CAM messages. CopaDrive allows a realistic evaluation of the platoon behavior at a microscopic scale and accurate simulation of network events, thus enabling us to study the impact of different trigger conditions in more complex settings.

The architecture of CopaDrive is presented in Fig. 5.5. Furthermore, the possibility of integration with a robotic testbed, as described in section 5.6 for future validation of the obtained data was seen as an advantage in choosing the tool. As in previous chapters, CopaDrive operates in a Linux

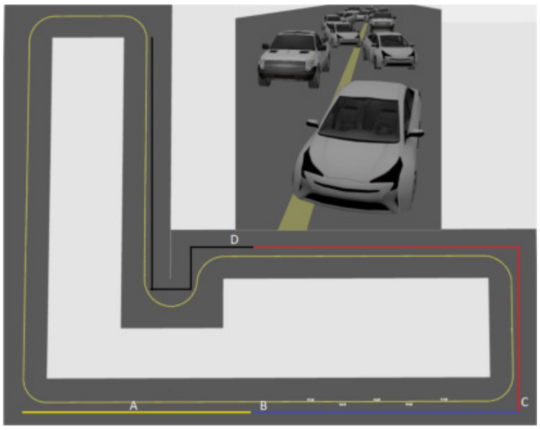


Figura 7.3: Track for Simulations

Ubuntu 18.04.6 Bionic, with Gazebo 9.0 and ROS Melodic. The PC running the simulations is an Intel® Core® i7-975H CPU, with 16 GB RAM memory and a NVIDIA Geforce GTX 1650.

7.4.1 Evaluated Scenarios

We defined four scenarios involving different challenges to the platoon's safety. In all simulations, the vehicles carry out a trajectory on the track presented in Fig. 7.3. This is divided into four sections to facilitate the discussion of the platoon performance: A (*yellow*), B (*blue*), C (*red*) and D (*black*). In scenarios 1, 2, and 3, we consider a platoon composed of six autonomous vehicles was used, the first being SV_0, the platoon leader, and the last SV_5. The leader performs its trajectory in all scenarios by following a line drawn in the track and applying a line following the algorithm. Each profile was evaluated using five complete tests, and the values presented are the average of these measurements. These tests have statistically validated the results of the experiments since the simulator shows only slight variations between each simulation performed.

Straight Line Scenario (Scenario 1) was developed to analyze the Co-VP performance while the vehicles travel on a straight road, without obstacles. This scenario encompasses sections A (*yellow*) and B (*blue*) of the track. In this scenario, the leader starts moving, increasing speed from 0 m/s to 16 m/s, after which it maintains a constant speed for approximately 500m until a complete break occurs. A video of the SC1 simulation of BSP and SP3 can be found at https://youtu.be/TEiSW1XFLJg.

Multi-curve Scenario (Scenario 2) extends SC1 with four curves, as presented in https://youtu.be/MCDcIEtaF8Y. It encompasses sections A, B, C, and D *black* of the track, with the two closed curves. The leader accelerates in the same way as in SC1, running through a straight line and performing three left curves and a sharp right curve. Then, it will run for 400m and then stop altogether. This scenario evaluates the Co-VP capacity to follow the leader's trajectory, including sharp curves in 90° and one of 180°. As long as the maximum vehicle steering angle is $|0.52|rad$, the vehicles will perform the curves through several small movements, affecting the system's stability. In this case, the lateral error increases its importance since it will demonstrate

Tabela 7.2: Model Parameters

Parameters	Definition
Vehicles (Scen. 1 to 3)	6
Max steering angle	0.52 rad
Safety distance (DS)	5.5 m
Time headway (TH)	0.5 s
Leader speed	16 m/s
Longitudinal: K_P, K_I, K_D	2.0 , 0.005 , 2.0
Lateral: K_P, K_I, K_D	2.5 , 0.001 , 1.0

the followers' ability to perform the same path as the leader. We can assume that the Co-VP speed error performance is similar to the one presented in SC1 since the acceleration and deceleration time is the same in both scenarios. However, as the trajectory diverges and the path is different, the e_5^{δ} tends to increase.

Obstacle Scenario (Scenario 3) adds five static obstacles on the track, represented by *pickups*. This scenario encompasses sections A, B, and C of the track. Such obstacles are placed on different sides of the track, as shown in Fig. 7.3, and their function is to *force* the platoon leader and his followers to perform a *slalom* maneuver. Thus, the vehicles must perform maneuvers with minimum error to avoid collisions with each other and with obstacles, receiving as information only the data from the SV_{i-1}. Thus, the leader follows the same acceleration profile presented in SC1 and SC2, reaching the desired speed, maintaining a constant speed to avoid obstacles, and aligning himself again to perform the curves indicated in the red color path, performing complete braking at the end of this excerpt. This scenario is presented in https://youtu.be/F3zGpP2XBBU.

High Medium Occupation (Scenario 4) encompasses sections A, B, and C of the track. Scenario 4 repeated the SC3 trajectory but increased the vehicle's number from 10 to 100 to evaluate the network congestion due to the proposed profiles. In a congested network, it is common for messages not to be delivered or to be delivered late. Therefore, it negatively impacts the performance of the platoon, leading to unsafe situations. Furthermore, the followers may collide and fail to correctly follow the leader's trajectory by not receiving the messages in time. Thus, even with restricted triggers, their information does not reach the destination in time to cause the correct safety action.

The main scenario, kinematic and control parameters enforced are presented in Table 7.2:

7.4.2 Metrics

As Co-VP errors propagate and accumulate from the first to the last follower, we focus our discussion on the performance of the subject vehicle SV_5. It is important to note that the triggers are analyzed individually car-by-car. Thus, there is a difference in the subsequent processing of the CAMs triggering in each platoon's vehicle. Due to this option, the effects of synchronization analyzed in [192] are practically mitigated due to the decoupling between vehicles. On the other hand, this effect makes the reaction times of each vehicle slightly different and consequently more

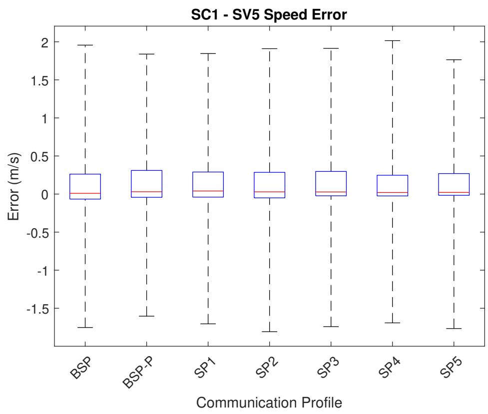


Figura 7.4: SC1 - SV_5 Speed Error Comparison

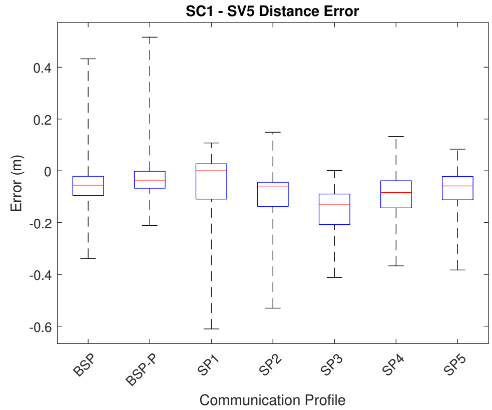


Figura 7.5: SC1 - SV_5 Distance Error Comparison (e_5^δ)

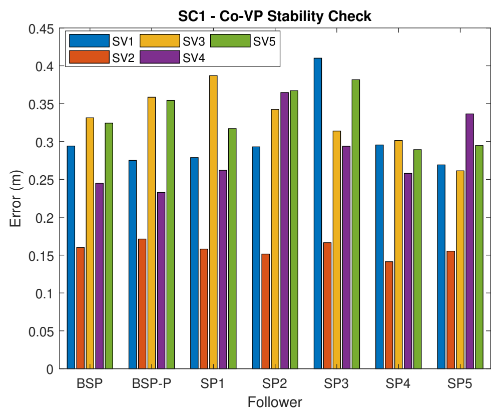


Figura 7.6: SC1 - Co-VP Stability Check

Profile	τ_{Head}	τ_{Pos}	τ_{Speed}	T_{max}	Total
BSP	435	79	292	508	1314
BSP-P	652	47	239	538	1476
SP1	621	64	285	400	1370
SP2	775	60	270	382	1487
SP3	16	639	253	390	1298
SP4	22	644	253	389	1308
SP5	19	650	245	391	1305

Figura 7.7: SC1 - Total Triggers per Profile

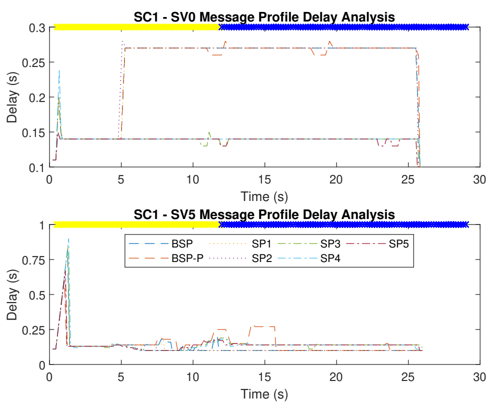


Figura 7.8: SC1 - Message Profile Delay Analysis

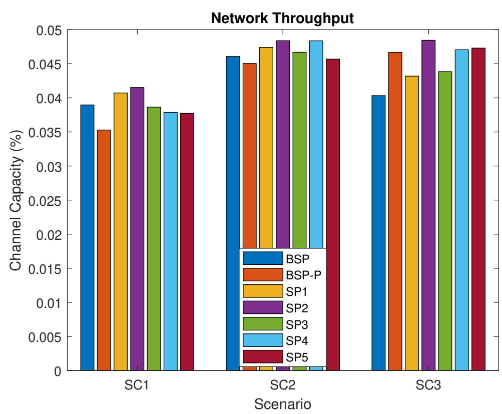


Figura 7.9: Scenarios Throughput

complex to be analyzed. Finally, we analyze the error between SV_{i+1} and SV_i or between SV_{i+1} and SV_0. The different trigger conditions happen because, although slight, there are variations in the trajectory of SV_0 in each simulation, which affect the movements and activation of their followers' triggers. In each scenario, the behavior of the network was analyzed in terms of **throughput**, **number of sent messages**, and **inter-message delay (IMD)**. The throughput is defined as presented in eq. 7.1.

$$Throughput = \frac{ReceivedMessages * MessageSize}{time * BitRate} \tag{7.1}$$

7.4.3 Straight Line Scenario - SC1

Platoon Safety: Fig. 7.4 presents the quartiles of the speed error for SV_5 throughout Scenario 1. The median speed error is close to zero in all profiles, while maximum velocity errors approach $2.0m/s$. The distance error of SV_5, e_5^{δ}, is depicted in Fig. 7.5. We observe that the median of the distance errors tends to be under-estimate (the vehicle is farther than the estimate). 50% of BSP errors fall in a limited range (-0.1 to $0m$), but the profile also overestimates the most. This justifies the introduction of BSP-P, in which the behavior is inverted: distances tend to be underestimated. SP1 to SP5 also tend to underestimate distances and have an inferior range of occurrence of error w.r.t. BSP-P, with SP3 having the smallest range (-0.4 to $0m$). Finally Fig. 7.6 presents the local stability condition for all SPs, demonstrating that the platoon stability is guaranteed since $\mathscr{L}_\infty(e_5^{\delta}) < \max_1^4(\mathscr{L}_\infty(e_i^{\delta}))$. These results show that the new profiles do not degrade performance regarding the established profiles (BSP/-P) but even improve it as SP3, SP4, and SP5 reduce e_5^{δ} in 28% (Fig. 7.5), as a consequence of the reduction of τ_{Pos} in these specific profiles.

Network Performance: The inter-message delay (IMD) is shown in Fig. 7.8 for SV_0 (top) and SV_5 (bottom). The IMD of SV_0 at the start of the trajectory is small for all profiles since it is the speed difference threshold τ_{Speed} that triggers the CAMs. After $5s$, when the speed stabilizes, the profiles with higher values of τ_{Pos} trigger messages with low frequencies. This behavior indicates that the leader occupies less transmission medium in the BSP, BSP-P, SP1, and SP2. However, the bottom of the figure shows that the IMD of SV_5 is nearly the same for all the profiles. This similarity confirms that all the profiles for this vehicle's quantities produce a similar network load. Fig. 7.9 shows the vehicle's throughput, calculated with eq. 7.1, with a bit rate of $6Mbps$ [82]. This figure demonstrates that the throughput variation between all profiles does not reach 0.01% of the channel capacity, showing that the better performance of CSP3, CSP4, and CSP5 does not imply considerable overhead in network usage.

Discussion: As followers diverge from the leader, they transmit messages at higher rates while correcting their trajectories in profiles with higher errors (BSP, BSP-P, SP1, and SP2). In profiles with a more conservative τ_{Pos}, more messages are fired due to the values of e_i^{δ}. The cost of sending more messages caused by the tighter trigger values on SP3, SP4, and SP5 is offset by the number of sent messages by other profiles. The last column of Fig. 7.7 illustrates this cost, in which the best performing profile (SP3) had 1% fewer messages than the worst performing one (BSP).

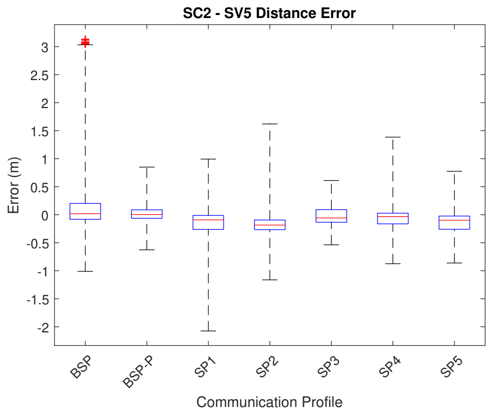


Figura 7.10: SC2 - SV_5 Distance Error (e_5^δ)

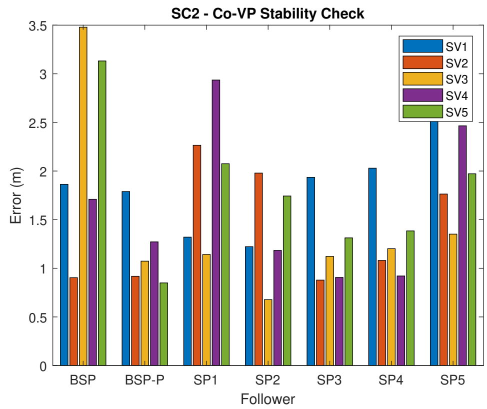


Figura 7.11: SC2 - Co-VP Stability Check

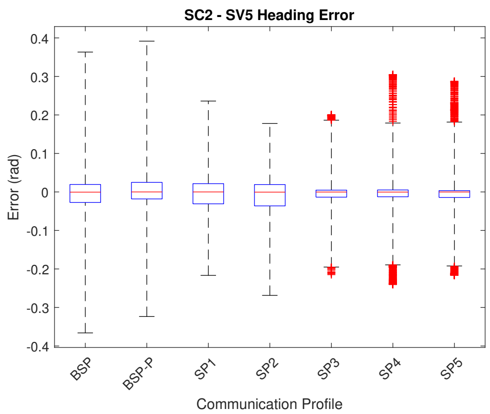


Figura 7.12: SC2 - SV_5 Heading Error (e_5^θ)

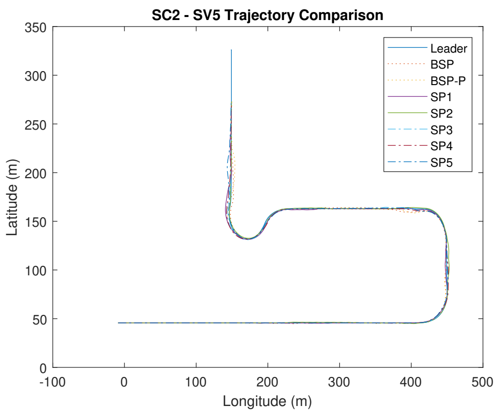


Figura 7.13: SC2 - SV_5 Trajectory Comparison

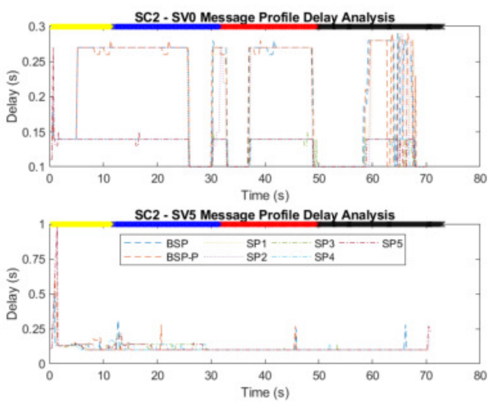


Figura 7.14: SC2 - Message Profile Delay Analysis

Profile	τ_{Head}	τ_{Pos}	τ_{Speed}	T_{max}	Total
BSP	2372	109	254	1011	3746
BSP-P	2435	95	139	1151	3820
SP1	2506	161	267	677	3611
SP2	2914	98	280	505	3797
SP3	1527	957	241	917	3642
SP4	1773	848	232	846	3699
SP5	2055	788	232	659	3734

Figura 7.15: SC2 - Total Triggers per Profile

Furthermore, the minor distance error profiles had the slightest trajectory corrections caused by τ_{Head}. The full set of triggers in each scenario is presented in Table 7.3.

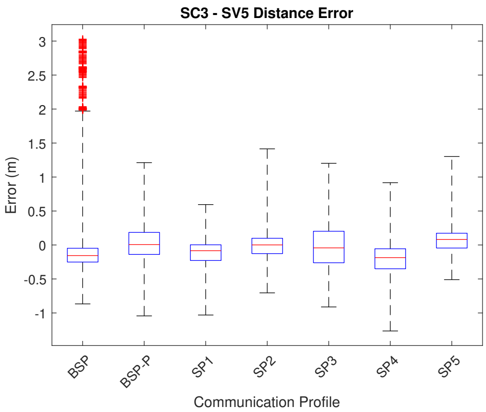


Figura 7.16: SC3 - SV_5 Distance Error (e_5^δ)

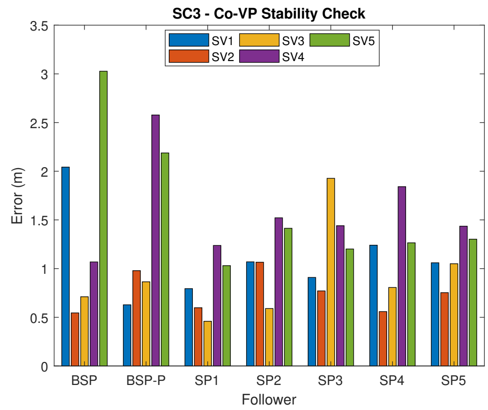


Figura 7.17: SC3 - Co-VP Stability Check

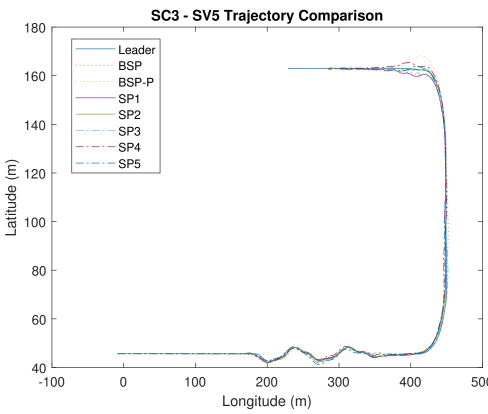


Figura 7.18: SC3 - SV_5 Trajectory Comparison

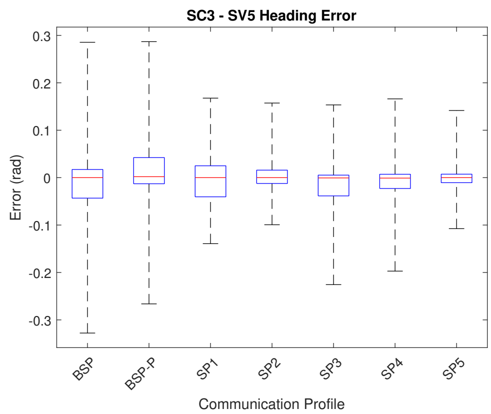


Figura 7.19: SC3 - SV_5 Heading Error (e_5^θ)

7.4.4 Multi-Curve Scenario - SC2

Platoon Safety: Fig. 7.10 shows distance error e_5^δ; as this scenario is more complex than scenario 1, the error tends to increases along the trajectory. This error is a consequence of the Euclidean distance used to calculate $d_{i,i+1}(t)$ since it is affected by the lateral deviation of the vehicles. Due to this, the distance error e_i^δ tends to increase on curves in order to reduce the e_i^θ, avoiding the *cutting-corner effect*. It demonstrates that the maximum variation of the distance error e_5^δ has a 25% reduction with SP1 in comparison with BSP and about 70% with SP3. The SP3 also outperforms the BSP-P in $\approx$ 15%. It means that the reducing τ_{Pos} has a larger influence on the distance adjustment of the platoon members than τ_{Head}, given that SP3 outperforms SP4 and SP5. In this scenario, BSP end up overestimating e_5^δ, while SP1 underestimates it. Thus, it is possible to observe that the platoon's performance is improved with SP3 since this profile presents the slightest variation (-0.5 to $0.6m$), still having the average value very close to zero. It is also interesting to notice that in the SP1 and SP2 profiles, the more restrictive value of τ_{Head}, without the reduction of τ_{Pos}, causes a slight downward shift in the average of e_5^δ, due to the corrections triggered not by position, but by the heading variation.

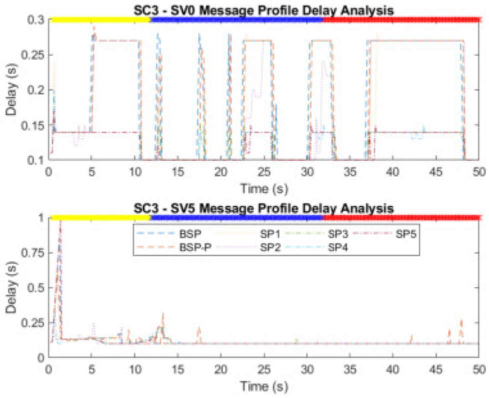


Figura 7.20: SC3 - Message Profile Delay Analysis

Profile	τ_{Head}	τ_{Pos}	τ_{Speed}	T_{max}	Total
BSP	1623	79	294	676	2672
BSP-P	1745	64	271	758	2838
SP1	1876	74	258	509	2717
SP2	2085	60	240	414	2799
SP3	1419	445	253	653	2770
SP4	1633	395	248	532	2808
SP5	1805	409	249	488	2951

Figura 7.21: SC3 - Total Triggers per Profile

The Co-VP performance regarding the distance error is confirmed in the stability check, illustrated in Fig. 7.11. All the profiles satisfy the *local* stability criteria, and the overall distance error of SP3 for all vehicles is smaller than the other profiles.

In scenario 2, a well-performed trajectory implies a small heading error e_i^θ, which indicates how well followers perform curves concerning the leader. Fig. 7.13 shows that the leader and vehicle SV_5 perform a similar path in all profiles, apart from minor oscillations after the first and the last curves. These oscillations can be better visualized in Fig. 7.12; the worst heading error of SV_5, e_5^θ, is near to $0.4rad$ with BSP/-P. Again, the best performance is obtained with SP3 since its application reduced the maximum value of e_5^θ in $\approx 50\%$, to $0.2rad$, even with complex curves, also decreasing the total variation between extreme error values.

Fig. 7.12 also demonstrates that exclusively reducing τ_{Head} leads to a better heading error e_5^θ performance, as SP1 and SP2 lowered it in $\approx 38\%$ in comparison with BSP/-P. The analysis of the maximum of e_5^θ in SP4 and SP5 indicates that the impact of reducing τ_{Pos} is greater than reducing τ_{Head} in the proposed profiles.

The impact on the number of transmitted messages relating to the threshold values of the heading difference τ_{Head} and position difference τ_{Pos} in SC2 can be observed in Fig. 7.15. The threshold of parameter τ_{Head} is the most activated due to the numerous trajectory adjustments of followers to execute the scenario's curves. Under SP3, the combination of a reduced τ_{Pos} with the standard τ_{Head} produces the best general Co-VP performance, reducing distance and lateral errors (e_5^δ and e_5^θ). These results confirm that reducing τ_{Pos} reduces the longitudinal and lateral errors, as observed in SC1 and SC2. Furthermore, under these conditions, the SP3 reduction improves the BSP-P performance both in e_5^δ and e_i^θ, also reducing the number of sent messages. Reducing τ_{Head} also does so, although to an inferior extent.

Network Performance: The inter-message delay (IMD) is shown in Fig. 7.14 for SV_0 (top) and SV_5 (bottom). As SC2 is an extension of SC1, the first interval between $0-25s$ presents a similar behavior. However, after the first curve, at the end of the blue section and the beginning of the red section, we observe in vehicle SV_0 a few periods of high-frequency/slight inter-message

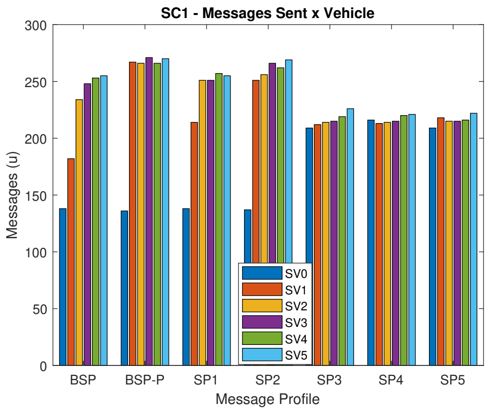


Figura 7.22: SC1 - Sent Messages per Vehicle

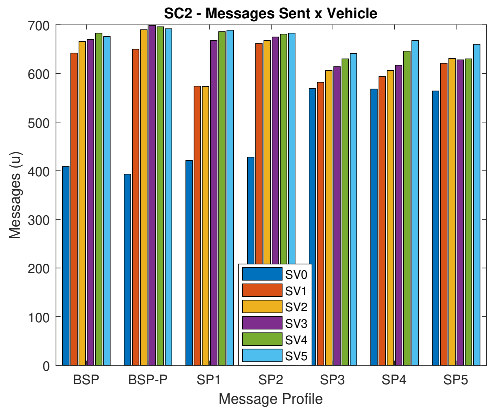


Figura 7.23: SC2 - Sent Messages per Vehicle

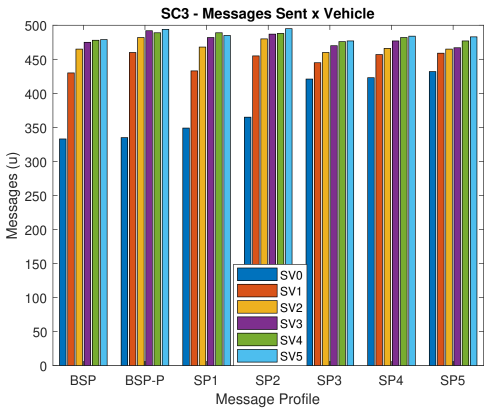


Figura 7.24: SC3 - Sent Messages per Vehicle

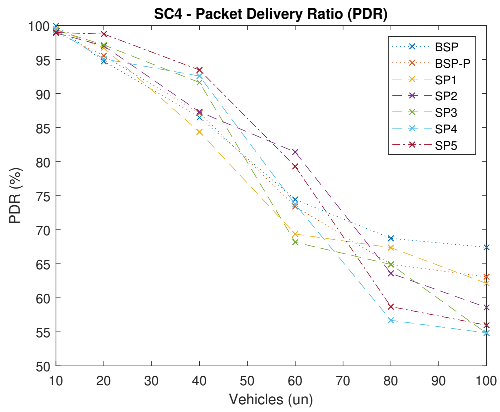


Figura 7.25: SC4 - Packet Deliver Ratio

delay. These high-frequency periods are caused by the path's curves and occur in all profiles. After this occurrence, the less restrictive profiles return to a lower message transmission frequency, as expected.

Nevertheless, the message delay of SV_5 presents a low IMD in all profiles, from the beginning to the end of SC2, due to the vehicle's trajectories adjustments. As in scenario 1, the propagation of control information hop-by-hop, from SV_1 to SV_5, leads to SV_5 constantly requiring more abrupt adjustments, hence the increased transmission frequency. Fig. 7.15 illustrates that all profiles have transmitted a similar number of packets at the end of SC2 as in SC1.

Discussion: Overall, we observe that SP3 has the best performance in the control metrics, with a saving of transmitted messages, with a reduced τ_{Pos}. This performance supports our claim that, by altering the message trigger thresholds (τ_{Pos} and τ_{Head})to become more sensitive, the platoon safety performance is improved without burdening the network, reducing the errors e_i^{δ} and e_i^{θ}.

7.4.5 Obstacle Scenario - SC3

Platoon Safety:

The distance error e_5^δ, presented in Fig. 7.16, shows that BSP has a similar performance regarding the maximum error variation as in SC2, $\approx 3.5m$, indicating that the obstacles cause little or no difference in both scenarios for this profile. However, as in SC1 and SC2, the proposed profiles and the BSP-P present a better e_5^δ performance, reducing the error variation to $\approx 2.0m$, albeit at a more significant error than in SC2. This result is expected since obstacles negatively impact the longitudinal platoon performance.

Even under these conditions, the platoon *local* stability is granted in all the proposed profiles, as depicted in Fig. 7.17, albeit with larger errors than the presented in SC1 and SC2. However, the BSP fails to provide Co-VP stability in SC3, since $\|H(s))\|_\infty > 1$.

As in SC2, the Co-VP performance in SC3 regards the follower's capacity to perform the same trajectory as the leader, with smaller e_i^θ as possible. So, the SV_5 trajectory in all profiles is presented in Fig. 7.18 and confirms that all followers avoid the obstacles. However, the accumulated error of the SV_5 trajectory limits the ability of the platoon to perform the second curve (in the red road area) in the same trajectory as the platoon Leader. In this scenario, the BSP-P has more significant oscillation, while the SP1 suffers from the *cutting-corner* effect. As the e_i^δ tends to increase over the curves to reduce the e_i^θ, this effect explains why the SP1 has a slightly better result than the other profiles, as presented in Fig. 7.16.

The *slalom* maneuver performed in this scenario relies on several curves produced by the leader's trajectory. So, the τ_{Head} have a direct impact over the in Co-VP heading error (e_i^θ), as illustrated in Fig. 7.19, while the influence of τ_{Pos} to e_i^θ is reduced, due to the required adjustments, given curves proximity. This figure demonstrates that the SP2 and SP5, with more restricted τ_{Head} values, have a better e_5^θ performance, presenting an improvement of 45% in comparison with the BSP/-P. The SP3 also outperforms the ETSI triggers performance in 26%.

Network Performance: The IMD is directly affected by the *slalom* maneuver, which triggers message transmissions at high frequency in all profiles, as visible in Fig. 7.20. As in SC1 and SC2, the period between $0-10s$ has similar behaviors, with two well-defined levels, for the BSP, BSP-P, SP1, and SP2, and others for SP3 and SP4, and SP5. However, when the SV_0 starts the object avoidance algorithm, the IMD is directly affected, being reduced for the profile T_{min}. This occurrence indicates where SV_0 avoids the obstacles and when it returns to a straight trajectory before performing the two curves and finally stops at the end of the scenario. As expected, the load on the network caused by SV_0 is slightly smaller in BSP, SP1, and SP2, while the BSP-P has the biggest number of triggered messages. However, as in previous scenarios, this behavior does not correspond to the other platoon vehicles. The bottom figure highlight that the network load caused by SV_5 is the same for all the profiles due to the constant adjustments. However, in this scenario, the restricted τ_{Head} produces a network overhead, as presented in Fig. 7.9 and confirmed in Fig. 7.21. Thus, SP5 implies a 10% overhead over sent messages and about 0.005% over the network capacity.

Discussion In this scenario, the impact of the τ_{Head} restriction becomes more evident since the leader's *slalom* to avoid obstacles forces the other vehicles to make quick turns to correct their route. While the standard profiles in ETSI ITS G5 present a similar performance in scenarios 1 and

2, the proposed profiles still produce better performance, with lower distance and heading errors. Even the reduction of T_{max} in BSP-P does not reduce the e_i^θ as the proposed profiles. However, this scenario presents an overhead of profiles with a more restricted τ_{Head}, affecting a more crowded network scenario.

The comparison between the SPs standardized by ETSI and those presented here takes into account not only the reduction of e_i^δ and e_i^θ but also their impact on the network. The analysis of the SC1, SC2, and SC3 results shows that SP1-5 reduces errors compared to BSP in all scenarios and also outperforms BSP-P to a lesser extent in reducing e_i^δ and with great advantage in reducing e_5^θ. The comparison of the cost of the higher constraint triggers can be seen in the Figures 7.7, 7.15, and 7.21, and also reinforced with the Figs. 7.22, 7.23, and 7.24, which illustrate the amount of messages sent by each vehicle in each scenario in each profile tested. These figures show that the evaluated profiles do not increase the number of messages sent by each car and show that BSP, even in simpler scenarios, tends to send more messages to obtain a performance inferior to SP3, for example.

7.4.6 Network Performance - SC4

The results on network performance presented for Scenarios 1, 2, and 3 showed that proposed SPs have a residual impact on the number of transmitted messages compared to ETSI profiles. In Scenario 4, we extend SC3, the best performing SP, by increasing the number of communicating vehicles to simulate high-density vehicle occupation conditions and, accordingly, high occupation of the wireless medium. For simplicity, additional 'virtual' vehicles were simulated as if they were in the same position as the leader. This setup, while very pessimistic, allows us to study performance in almost worst-case conditions: when the leader fires a message, the 'virtual' vehicles also send it, inducing an extensive medium usage at that instant. Tests were run with 10, 20, 40, 60, 80, and 100 cars. In SC4, our analysis is limited to Packet Delivery Rate (PDR) related to the increasing number of vehicles.

Fig. 7.25 shows PDR for each proposed profile as the number of vehicles increases. We assume that a better PDR leads to better Co-VP performance since more messages are delivered to the followers. However, in this paper, we will not address the PDR's decreasing impact on the platooning performance, but just the analysis of how the PDR falls with the rising number of vehicles in each profile.

Beginning with 10 vehicles, PDR is close to 100% in all profiles, from BSP to SP5, meaning that we can believe that the performance obtained in SC1, SC2, and SC3 should be maintained. Meanwhile, for 20 cars, the variation between the best and the worst PDR is 4%. Thus, under these conditions, the SP5 PDR is ≈ 99% while the BSP PDR is nearly 95%. On the other hand, SP1, SP2, and SP3 PDRs remain around 97%, and SP4 and BSP-P deal with 96%. Therefore, the decreasing PDR should not have a high impact on the Co-VP performance in these conditions. This analysis also shows that the more significant restriction of τ_{Head} and τ_{Pos} does not severely impact the number of packets since the drop in the PDR is directly related to the increase in collisions.

The best PDR response decays to 94% in SP5 while increasing the number of vehicles to 40. On the other hand, the worst PDR response is obtained from the SP1, with 94%. SP3 and SP4 have a similar PDR response as SP5, while BSP, BSP-P, and SP2 have a PDR near 87%.

As expected, when the number of vehicles increases up to 60, the PDR response decreases in all the profiles. However, the best PDR response is obtained in SP2, with 82%, while SP5 decays to 80%. BSP, BSP-P, and SP4 have a PDR of 75%. The SP3 and SP1 have the worst PDR result, close to 69%. Finally, raising the car quantity to 80 and 100 leads the PDR response to under 70% in all the profiles, which indicates a communication link that may cause safety concerns [17].

Discussion The analysis of the PDR chart in Fig. 7.25 confirms that the profiles proposed with the restriction of τ_{Pos} and τ_{Head} do not cause more network congestion than the ETSI profiles. Furthermore, although their PDR also declines with the increase in network congestion, their performance in SC1, SC2, and SC3 scenarios proves that this is an option that tends to increase the safety of the platoon by reducing errors e_i^δ and e_i^θ.

Considering the proposed scenarios and the results presented, we reinforce the thesis that a greater restriction of the ITS-G5 τ_{Pos} and τ_{Head} triggers positively impacts platooning performance without causing overhead on the communications network. Thus, in adverse conditions, such as obstacles and curves, the SP1-5 profiles present better performance than those established by ETSI. Moreover, among the profiles presented, it was possible to observe that the SP3 profile represents the best cost-benefit, as it reduces the errors e_5^δ and e_5^θ in all scenarios, without providing overhead concerning the profiles proposed by ETSI. The PDR response of SP3 also indicates this profile's applicability regarding the proposed conditions, with reduced errors and similar network performance compared with other profiles. So, for an up-limited scenario of 40 vehicles, the 94% PDR suits as an acceptable compromise for a Co-VP application, but further investigation is necessary to evaluate the platoon safety condition under more congested scenarios.

Thus, based on the results obtained, we propose the implementation of the Platoon Service Profile (PSP), based on SP3, with a more restricted τ_{Pos}, aiming to increase the platoon's performance, reducing longitudinal and lateral errors. This profile showed an improvement in e_5^δ of 74% on SC2 and 44% on SC3 compared to BSP/-P. Also, improved e_5^θ by 42% on SC2 and 50% on SC3 over the same profiles, without increasing overhead. As the scenario proposed in SC3 can be considered extreme, due to the presence of the "slalom", this profile would successfully meet the needs of Co-VP systems in a diversity of scenarios not covered by the ETSI BSP/-P.

7.5 Conclusions

This work presented five new Service Profiles, using different message threshold values over triggers τ_{Head} and τ_{Pos}, compared to the BSP and BSP-P profiles proposed in the ETSI specifications. The performance observed in the studied scenarios shows that the proposed adjustments in τ_{Head} and τ_{Pos} positively impact the Co-VP safety performance, reducing both longitudinal and lateral errors.

In SC1 and SC2, SP3 presented the best results, with τ_{Pos} adjusted, while in SC3, the best lateral error was achieved with the τ_{Head} adjustment. Thus, the integrated analysis of all scenarios showed that the joint reduction of the triggers τ_{Head} and τ_{Pos} did not necessarily show the best result in all scenarios in lateral and longitudinal terms. The different results are affected by triggering conditions that cause different adjustments and corrections in the platoon, often increasing longitudinal errors to correct lateral errors.

Furthermore, the need for corrections arising from errors accumulated throughout the platoon and the activated triggers also implied an increase in network load. Thus, it was observed that the use of more restricted triggers did not significantly increase the load on the network since these triggers implied fewer corrections and, consequently, fewer messages sent. Such observation is extended to a scenario of an increase in the number of vehicles, where less restricted triggers do not represent a significant gain in terms of network performance, burdening it in a very similar way in all profiles.

We conclude that reducing the value of τ_{Pos} proposed by the ETSI ITS standard can increase the safety conditions of the platoon in complex scenarios involving curves and even with the presence of obstacles. So, we propose a Platoon Sevice Profile (PSP) based on the τ_{Pos} reduction from $4m$ to $2m$. Furthermore, it should also be noted that the reduction of τ_{Head} in very close obstacle scenarios also improves the lateral error, ensuring system safety.

In future work, we will investigate the impact of the drop in PDR with the increase of vehicles on the safety conditions of the platoon. This scenario also suggests the possibility of validating a flexible trigger profile based on road conditions.

Tabela 7.3: Triggers Profile per Vehicle

-	-	SV0				SV1				SV2				SV3				SV4				SV5			
Scen.	SP	τ_{Head}	τ_{Pos}	τ_{Speed}	T_{max}	τ_{Head}	τ_{Pos}	τ_{Speed}	T_{max}	τ_{Head}	τ_{Pos}	τ_{Speed}	T_{max}	τ_{Head}	τ_{Pos}	τ_{Speed}	T_{max}	τ_{Head}	τ_{Pos}	τ_{Speed}	T_{max}	τ_{Head}	τ_{Pos}	τ_{Speed}	T_{max}
	BSP	1	42	42	53	37	24	44	77	91	6	49	89	108	2	46	93	96	2	55	101	102	3	56	95
	BSP-P	1	37	33	65	122	2	35	108	130	2	41	93	128	0	42	101	136	4	44	82	135	2	44	89
	SP1	1	41	42	54	82	15	43	74	134	5	45	68	132	2	53	65	145	1	48	64	127	0	54	75
SC1	SP2	1	43	40	55	124	9	42	89	147	4	43	68	162	3	44	68	163	0	51	51	178	1	50	51
	SP3	1	107	39	62	1	111	40	60	1	108	40	66	0	109	41	66	4	109	43	64	9	95	50	72
	SP4	5	108	41	65	1	111	40	62	1	112	40	62	0	104	42	70	7	102	40	72	1	107	50	64
	SP5	1	105	39	65	6	112	40	64	1	108	41	66	0	109	40	67	1	112	41	63	10	104	44	66
	BSP	168	82	38	121	400	13	40	189	419	7	41	199	449	5	42	174	467	0	45	171	469	2	48	157
	BSP-P	154	76	16	147	414	17	14	205	455	2	23	210	472	0	23	204	470	0	28	198	470	0	35	187
	SP1	184	81	40	116	371	39	39	125	390	38	43	102	501	3	50	114	517	0	50	119	543	0	45	101
SC2	SP2	196	79	38	115	524	9	40	89	538	5	47	78	545	3	51	76	552	2	51	76	559	0	53	71
	SP3	171	225	39	134	212	203	39	128	246	159	39	162	263	145	39	167	306	125	41	158	329	100	44	168
	SP4	174	225	38	131	249	183	39	123	265	159	39	143	304	139	39	135	362	98	37	149	419	44	40	165
	SP5	181	221	37	125	336	133	38	114	357	125	37	112	371	115	38	104	372	122	35	101	438	72	47	103
	BSP	159	50	41	85	254	17	43	118	287	7	45	128	310	1	56	110	307	3	55	115	306	1	54	120
	BSP-P	167	41	38	102	290	12	35	136	312	5	46	134	324	1	53	129	321	3	52	128	331	2	47	129
	SP1	184	46	42	79	284	19	42	90	329	7	43	90	362	2	32	87	351	0	49	91	366	0	50	72
SC3	SP2	208	38	39	87	340	13	39	67	377	5	41	66	391	1	36	60	382	1	45	64	387	2	40	70
	SP3	161	129	40	95	210	95	37	107	230	75	41	118	256	53	42	122	280	46	42	111	282	47	51	100
	SP4	171	122	40	92	256	76	39	90	270	58	38	102	301	44	41	93	321	49	45	72	314	46	45	83
	SP5	210	115	40	88	296	75	39	82	301	70	38	95	310	61	41	78	339	46	43	71	349	42	48	74

Capítulo 8

CopaDrive Extensions

This chapter presents two CopaDrive extensions. First, using the described software framework, we use minor variations of the same simulator to develop an interactive tool for the study of control, to be used in classrooms, promoting active study. And, having the integration of ROS with OMNET++ as a base, CopaDrive was extended to validate the DSME MAC of IEEE 802.15.4 as an intra-vehicle communication network to minimize accidents.

8.1 Introduction

Scientific research should not be seen as an end in itself but rather as a bridge and a basis for discoveries to be made and built upon. In this way, a series of tools serve as an intellectual framework for other research to emerge, allowing knowledge to expand to other frontiers. The very construction of CopaDrive, presented here, emerges from the efforts of other researchers who created the bases and models that were later used here.

Thus, given the flexibility of CopaDrive provided by ROS, it can be used in different contexts, expanding its capacity for integration and applicability in different contexts of systems engineering and validation. Thus, although it was initially created in the context of the European SafeCOP project [242], it has been used in other projects, aiming to validate algorithms and models to increase Co-CPS applications' safety. Some examples of these projects where CopaDrive has been used are ADACORSA [6], FLOYD [46], and INSECTT [134].

Seeking to show this flexibility, we present two variations of CopaDrive, applied to different contexts. Initially, we redesigned the platoon leader vehicle application for its application as a tool to teach control techniques to engineering students. Based on this variation, it is possible to study control models and apply them to bring the theory learned in the classroom closer to the practice of moving the vehicle over a circuit. Next, we present how the ROS/OMNET++ integration could be modified to validate an intra-vehicle network based on the IEEE 802.15.4 standard to increase the safety of parking and obstacle detection applications and emergency action-taking. In all cases, the control models can be interchanged between applications, minimizing the tool's learning curve and allowing the user to focus on the technology being validated.

8.2 RosDrive: An Open-source ROS-Based Vehicular Simulator for STEM Control Systems Classes Tutorial

Control system techniques are one of the most significant challenges in several engineering courses. Since it requires extensive mathematical background, a theoretical load is quite extensive, requiring effort to learn by students and teachers. Moreover, there is still an inherent difficulty in transporting the studied theory to practice, making it challenging to retain learning [30]. Thus, alternative teaching techniques [198] can facilitate knowledge production and construction of the skills expected by the agents involved. In addition, [138] concludes in their work that the student's perception of applicability and the ability to construct different solutions is a motivator for the search for more knowledge.

This line of education development puts the student as a producer of dynamic and practical knowledge. It should be encouraged to take an active and autonomous attitude and not necessarily follow pre-established models [201]. So, the student can go further and propose new solutions to existing problems and even create different issues. Active student engagement in the learning process also helps to keep the motivation to research and learn [250], using Active Learning techniques. Thus, integrating different areas of knowledge, experimentation, and implementation allows the student to retain more excellent expertise and develop new skills. This integration of knowledge is called STEM - Science, Technology, Engineering, and Mathematics [315].

A standard solution in many universities is using pre-defined laboratory sessions, using commercial kits such as [305, 189, 88]. Although such solutions are attractive, efficient, and robust, they are often expensive and not flexible for experimenting and developing different solutions. Nevertheless, using Arduino development kits has shown promising results as a learning tool [108]. This study suggests the development of kits that can be used throughout the semesters, gradually increasing the project's difficulty [38] and even in specific dynamics and control systems projects [214].

In the same line of knowledge integration, other low-cost projects have been developed and implemented, giving students greater flexibility in experimenting with techniques and knowledge. For example, in [114], the authors proposed an educational line-following robot based on Arduino, allowing the implementation of low-level control techniques. A similar application is presented in [247], suggesting an even lower cost robot with less flexibility. The increased complexity of possible control algorithms is achieved on other platforms, such as those seen in [51, 172]. However, such applications imply a significant increase in project costs. Thus, although Arduino-based solutions integrate the theoretical model and practice regarding control aspects, they present a limitation regarding the complexity of the algorithms, given the restriction of processing capacity and design flexibility, due to the need to purchase different sensors.

A solution that combines a low-cost implementation with flexibility and allows knowledge retention through experimentation and active learning is based on realistic open-source simulators. In STEM, a simulator represents a crucial stage of development and education, reducing the time to produce prototypes. Thus, emulating a real scenario with physical interactions allows the development of safety tests in different environments and situations. As a result, it is possible to experiment with techniques, analyze results and propose solutions flexibly, with great speed and less cost.

The work done in [263] presents some of these tools, comparing simulators such as Webots [206], Gazebo, and ROS [216], using criteria such as supported operating systems, programming languages, documentation, tutorials, among others. Others have been developed over time, such as presented in [268], where a virtual laboratory is designed so that students can experiment with models of line-following robots for competitions. However, such a simulator does not allow the 3D visualization of the models, allowing only the testing of the proposed algorithms. Another interesting simulator is proposed in [60], which presents CARLA, an open-source simulator aimed at autonomous-driving research in this work. It is a very realistic simulator with many items, with several physical interactions between the components. However, despite being an extensible platform for new developments, its vehicle control methods are limited to artificial intelligence learning models without control models.

The authors of [286] present a simulator that uses a competition model to teach robotics based on ROS. An autonomous robot capable of traveling a path is used in this simulator, following directions on the track. Such a simulator showed promising results when crossing the designated paths but presented the limitation of not using a realistic vehicle model or even different control models. It has also been used in competition simulation, which increases students' comprehension and stimulates self-learning [117, 229].

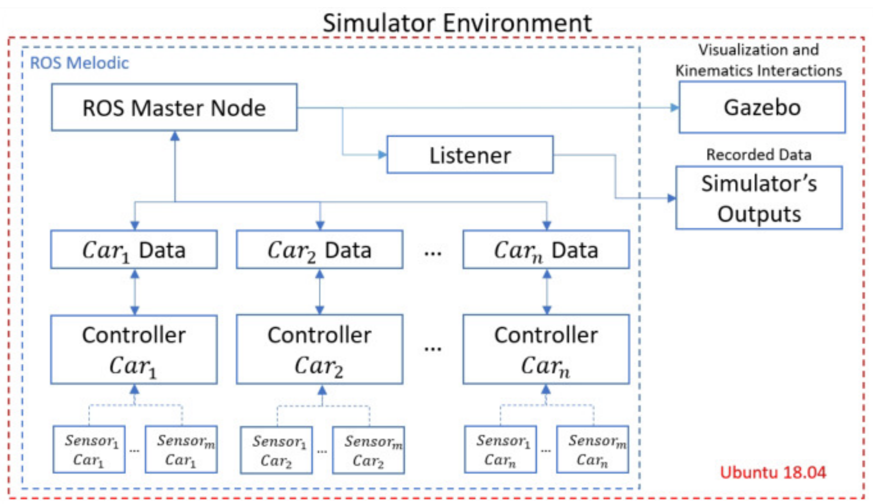


Figura 8.1: General Simulator Architecture

Seeking to use the advantages of a simulator capable of emulating realistic vehicles, RosDrive is presented. A flexible platform based on ROS and the 3D simulator, Gazebo, for studying different models of vehicle control. RosDrive uses an electric vehicle model [280], with several sensors capable of covering different routes and avoiding obstacles. Thus, the student will be able to implement additional control strategies, analyze the system's responses, and visualize the impacts of theoretical models and their variables on the simulated scenarios. In addition, the tool allows the use of different strategies in different vehicles, allowing the comparison between the adopted models. For instance, a line follower controller mode with obstacle avoidance will be implemented to exhibit the simulator results. The tool's flexibility allowed its extension for the study of communication models [98], the development of hardware in the loop (HIL) simulation [101], and the implementation of the same control model on a testbed platform [97]. As it is an open-source tool, the full code access is provided at `https://github.com/enioprates/rosdrive`, for general use, with all the necessary steps for its installation. The general simulator environment is illustrated in Fig. 8.1.

8.2.1 Simulator Architecture

This section will introduce the simulator tools and their general architecture, providing details about the vehicle model and data analysis.

8.2.1.1 Robot Operating System (ROS)

ROS is an open software developed by Open Source Robotics Foundation. It is a robotic middleware with many software frameworks for robot software development. It provides hardware abstraction, enabling users to avoid low-level problems with profound device control, communication between nodes and processes, and packet management. ROS-based functions are realized in nodes that may post, receive and reproduce control features, sensor data, state of the node, or general messages. ROS is not a real-time framework or a Real-time Operating System (RTOS). This project will be used in ROS Melodic distribution.

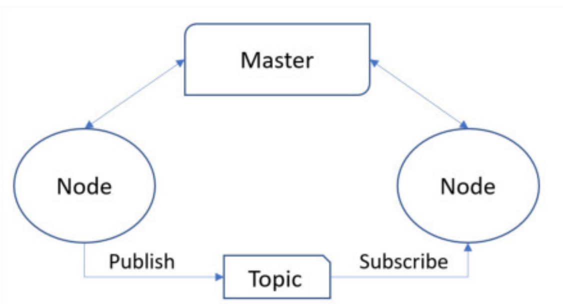


Figura 8.2: Publish/Subscribe Model

The basic concepts of ROS are nodes, Master, messages, and topics. The Primary node works as a central node of the system, storing data and information regarding the ROS Nodes. Nodes inform their registration information to the Master and then can receive data from other nodes.

The Master is also responsible for reporting the nodes, using Callbacks, if new information or connections are made. The nodes exchanges messages using the publish/subscribe method, as described in Fig. 8.2.

Due to its flexibility, ROS has been used in several vehicular applications, such as ground [236], aerial [107], and water [291] and many other robotic platforms. As a consolidated open-source community, several new libraries are available and supported, at the same time that it is highly portable between platforms, including embedded platforms [157]. The extensive material allows a quick learning curve for the student, enabling a simple familiarization with the commands and interfaces and quickly creating new modules.

8.2.1.2 Gazebo

One of the critical aspects of a learning-oriented system is its ability to present the results of user interactions intuitively. Thus, the high capacity of ROS to integrate with other platforms shows itself to be a competitive advantage since its functionalities can be extended, expanding the experimentation horizon. For example, integration with an automated simulation tool helps to visualize the iterations between objects simply, aiding in learning [163]. One of the most used tools for robotic simulation in ROS is Gazebo. The Gazebo is an open-source 3D robotics simulator, for indoor and outdoor environments, with multi-robot support that allows a complete implementation of dynamic and kinematic physics and a pluggable physics engine. Furthermore, it provides a realistic rendering of backgrounds, including high-quality lighting, shadows, and textures. In addition, it can model sensors that "see"the simulated environment, such as laser range finders, cameras (including wide-angle), and Kinect style sensors, among others.

The Gazebo presents the same message interface as the rest of the ROS ecosystem. So, the development of ROS nodes is compatible with simulation, logged data, and hardware. Many projects integrate ROS with Gazebo, such as the QuadRotor presented in [205], the Humanoid implementation in [90], and the Ground Vehicle in [236]. As a powerful and very visual tool, Gazebo has also been used as the simulation environment for several technical challenges and competitions, such as NASA Space Robotics Challenge (SRC) [122], Agile Robotics for Industrial Automation Competition (ARIAC) [209], and Toyota Prius Challenge [217].

The Gazebo is responsible for realistically mimicking the system's fundamental dynamics, representing physical issues such as mass, inertia moment, friction, and even collisions. To ensure better representation, Gazebo supports four engines: Simbody [249], Bullet Physics [228], ODE [255], and DART [116]. Such engines guarantee a wide range of representations, bringing simulations closer to reality, and offering the student a greater possibility of representing theoretical concepts practically.

Although some Gazebo components show some lag with new technologies, its overview still has more advantages than the alternatives presented. For example, Unity [129] has similarities in the implemented physics, but its integration with ROS is still complex. Furthermore, the Webbots recently developed a ROS integration but still do not have the same flexibility in implementing different physical models. Finally, the Coppelia [50] does not have the same rendering quality [237] as Gazebo, although it has similar flexibility and quality in physical representation.

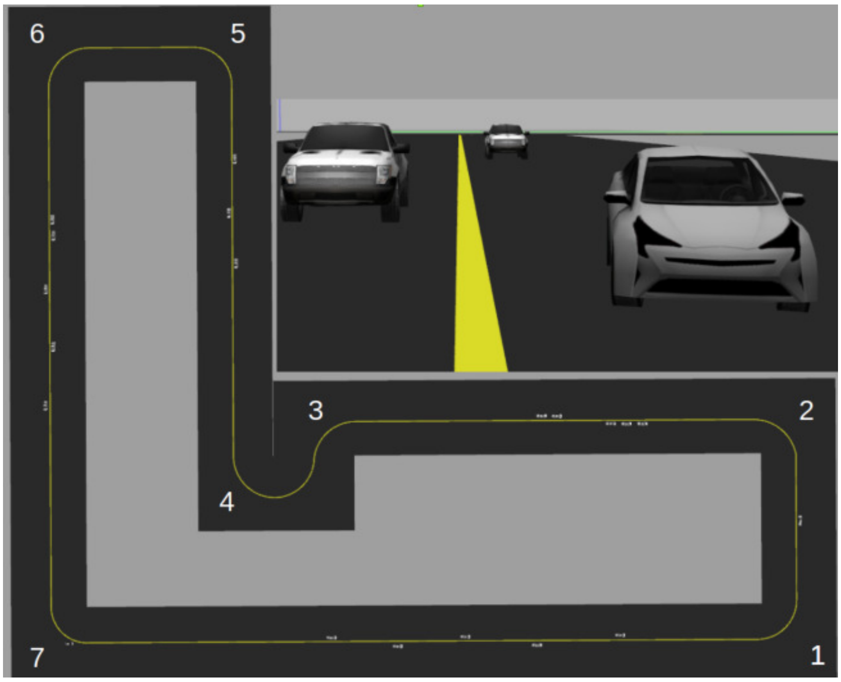


Figura 8.3: Track Model

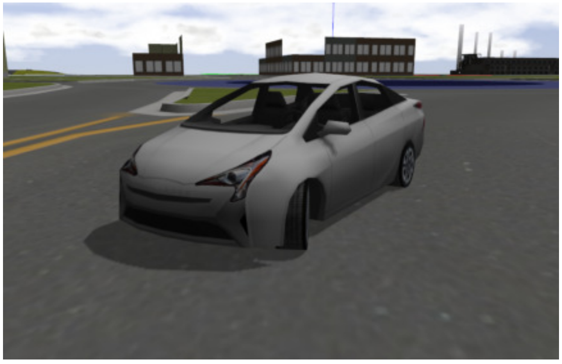
8.2.1.3 Scenario and 3D Vehicle Model

The Gazebo allows the construction of several different scenarios, including as many objects as desired. Those objects can be static or dynamic and controlled in the simulation. For illustration, this work introduces the track presented in Fig. 8.3 with and without obstacles. Those obstacles can be removed or added by the user.

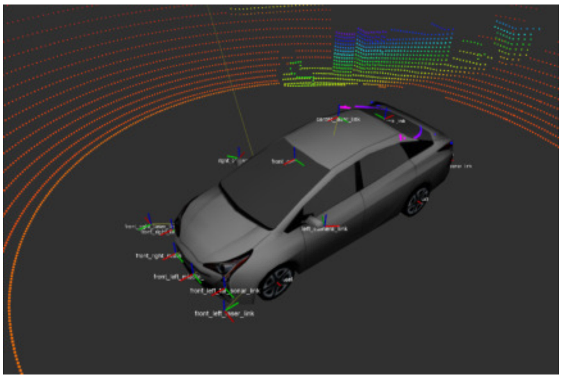
ROS applications have a launch file that allows the easy start of several applications with previously saved scenarios and desired configurations. In this project, the file *car_demo.launch* is responsible for starting the track, and the *cars.launch* defines the vehicle's initial coordinates and model. The simulator flexibility allows different car models to modify their positions and configurations, including or removing sensors. The sensors data can be real-time observed through the *RViz* software.

The 3D car model used in this work was presented in [280]. Figure 8.4 illustrates the Hybrid Prius 3D model's main details. Its fundamental dynamics are contained in the node *PriusHybrid-Plugin.cpp*, and the model's characteristics can be edited in *prius.urdf*.

The primary vehicle controllers, such as throttle, brake, steering, and gear, can be actuated by publishing to a ROS topic. Thus, the vehicle Powertrain will control the gear control in this simulation. The simulated vehicle also has multiple sensors: 16-beam LIDAR on the roof, eight ultrasonic sensors, four cameras, and two planar LIDARs. However, adding or removing sensors is a simple task that allows adjustments. Furthermore, implementing the vehicle with all the



(a) Simulated 3D Prius front view



(b) Prius Sensors Position

Figura 8.4: Prius Gazebo Model [280]

kinematics and basic controls enables the study of other project aspects, such as motion control, platooning, stability, and detection and avoidance models.

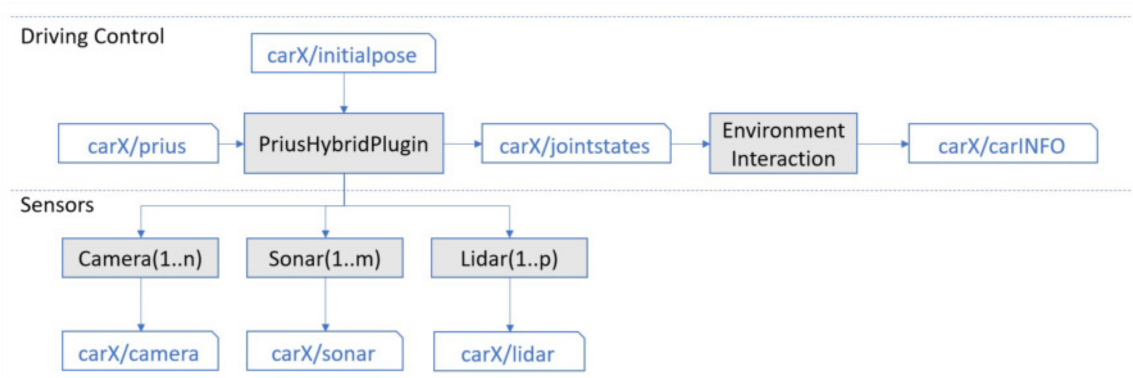


Figura 8.5: Prius Information Structure

The vehicle information flowchart is illustrated in Fig. 8.5. All the vehicles have the same model, and the simulation is composed of $n \in \mathbb{N}$ vehicles. The full set of cars can be defined as $car_n = \{i \in \mathbb{N} | 0 \leq i \leq n\}$. The information provided by each module/node is:

- $car_i/initialPose$: defines the vehicle's initial position
- $car_i/prius$: new vehicle settings – throttle, break, steering
- $car_i/jointstates$: conditions of each vehicle component - wheels and steering
- $car_i/carINFO$: vehicle's current state – throttle, brake, speed, latitude, longitude, steering, heading, etc.
- $car_i/camera$: vehicle's onboard cameras info
- $car_i/sonar$: vehicle's onboard sonars info
- $car_i/lidar$: vehicle's onboard LIDARs info

- PriusHybridPlugin: dynamics and vehicle model
- Environment Interaction: Gazebo calculation about interactions
- Camera($1 \dots n$), Sonar($1 \dots m$) and Lidar($1 \dots p$): sensor nodes

All the sensors can be added, removed, or modified in the file *prius.urdf*. The vehicle control is managed through the data sent to $car_i/prius$ topic, which works as the vehicle's input center, receiving throttle, brake, and steering. The throttle and brake have a limit from 0 to 1, and the steering has a range from $-30°$ to $+30°$. Its format is defined in the "Control.msg". To better understand the text, the rest of this text will refer to a generic simulated vehicle identified by the "i"index, unless in cases where some differentiation is necessary.

8.2.1.4 System's Outputs

As a simulator for learning purposes, the system's outputs are essential. Moreover, as a STEM application, with many details, many analyses must be performed using a mathematical approach. The output data will allow the study and comparison of each simulation, allowing the student to evaluate the impact of slight differences in the system's response in each experiment. The system's outputs are provided in *.csv* files generated during the simulation. The module *listener.py* is responsible for collecting the desired vehicle's data in the related topics and exporting that to a *.csv* file.

During the simulation, the topic $car_i/carINFO$ can be used to perform a Real-Time system evaluation, showing the vehicle's most important information, like coordinates, heading, speed, throttle, and brake conditions. The *listener.py* collects this data and adds some information to the simulation's output file, triggered by the car's movement or spent time. The output file contains the timestamp, coordinates, speed, speed error, throttle, brake, heading, heading error, and sonar information, in this version.

8.2.2 Control Algorithms

This section will introduce the controller models used in this simulator. Then, it will discuss the Cruise Controller (CC), the Line Follower characteristics, and the Obstacle Avoidance Strategy. The Prius model simulates sensors published to the $car_i/carINFO$ topic. This topic contains the primary data about the vehicle, like latitude, longitude, altitude, heading, speed, direction, steering angle, acceleration pedal percentage, and brake pedal percentage. All the data is updated every $0.01s$.

8.2.3 Vehicle Model

The vehicle model used in this work is based on the two-degree-of-freedom bicycle system, as shown in Fig. 8.6. This model considers the car's rotation around the z-axis (θ) and its lateral velocity. Assuming x and y as the vehicle's frame coordinates, respectively, and θ its rotation in

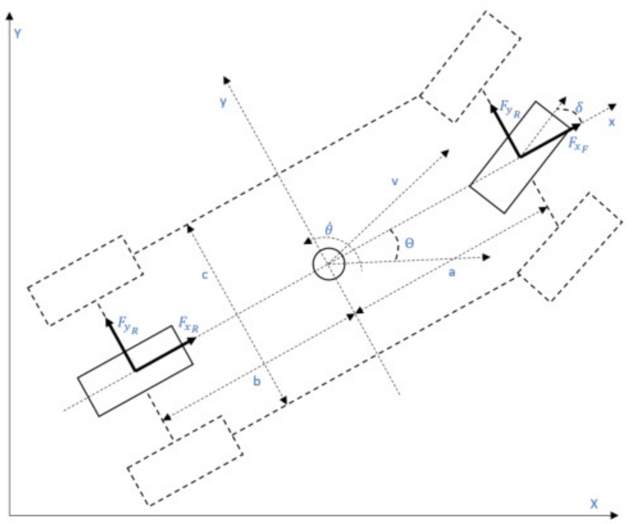


Figura 8.6: Vehicle 2D Model and Coordinates

the z-axis, X, Y, and Θ are their absolute equivalents in the global frame. Thus, the vehicle frame can be expressed using the rotation angle θ in the global structure ($\Theta = \theta$). Finally, The steering angle, described in the vehicle's frame, is defined as δ and admits that both wheels turn at the same value. By applying Euler-Newton equations [16], it is possible to simplify the vehicle's dynamics in the plane as:

$$m\ddot{x} = m\dot{y}\dot{\theta} + F_{x_F} + F_{x_R}, \tag{8.1a}$$

$$m\ddot{y} = -m\dot{x}\dot{\theta} + F_{y_F} + F_{y_R}, \tag{8.1b}$$

$$I\ddot{\theta} = aF_{y_F} - bF_{y_R} + c(-F_{x_{F,l}} + F_{x_{F,r}} - F_{x_{R,l}} + F_{x_{R,r}}), \tag{8.1c}$$

where I is the inertia moment, m is the vehicle mass, and F_x and F_y are the forces in x and y directions, and the subscriptions r and l indicates the force direction compound. Finally, the kinematic model, translated to X and Y coordinates, can be described as:

$$\dot{X} = \dot{x}\cos\Theta - \dot{y}\sin\Theta, \tag{8.2a}$$

$$\dot{Y} = \dot{x}\sin\Theta - \dot{y}\cos\Theta, \tag{8.2b}$$

$$\dot{\Theta} = \dot{\theta}. \tag{8.2c}$$

8.2.3.1 Cruise Controller (CC)

The Prius model does not allow direct speed control but only throttle and brake adjustments. So, this simulator adopts a Proportional Integral Derivative (PID) strategy for the vehicle speed controller. Although this controller is quite simple, it will help the student to develop basic control skills and move towards other implementations, including several autopilot strategies and tuning models [288]. In this way, the Cruise Controller (CC) will be responsible for keeping the constant speed during the vehicle's movement and adjusting it when necessary, changing the brake and the throttle pedals through the $car_i/prius$ topic. The PID equation is defined as follows:

$$\alpha(t) = K_P * \varepsilon^{\sigma}(t) + K_I * \int \varepsilon^{\sigma}(t)dt + K_D * \frac{\Delta\varepsilon^{\sigma}(t)}{dt}, \tag{8.3}$$

where K_P, K_I, and K_D denote the Proportional, Integral, and Derivative gain constants, respectively, $\varepsilon^{\sigma}(t)$ is the speed error, measured by the difference between the current speed value and the desired one and α is the expected acceleration. The α is then normalized to a value between $-1\ldots1$, representing the Throttle and Brake pedals usage. A positive value indicates that the Throttle pedal has been used while the brake is free. Conversely, the brake is pressed for a α negative value, and the Throttle pedal is free. The complete controller is illustrated in Fig. 8.7, where it is assumed that the time constant of the actuator is much bigger than the motor one, and the CC algorithm is summarized in Alg. 1.

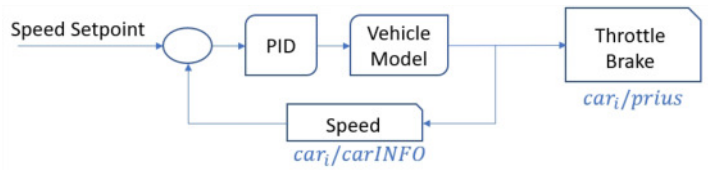


Figura 8.7: PID CC

Algorithm 1 Cruise Controller Algorithm

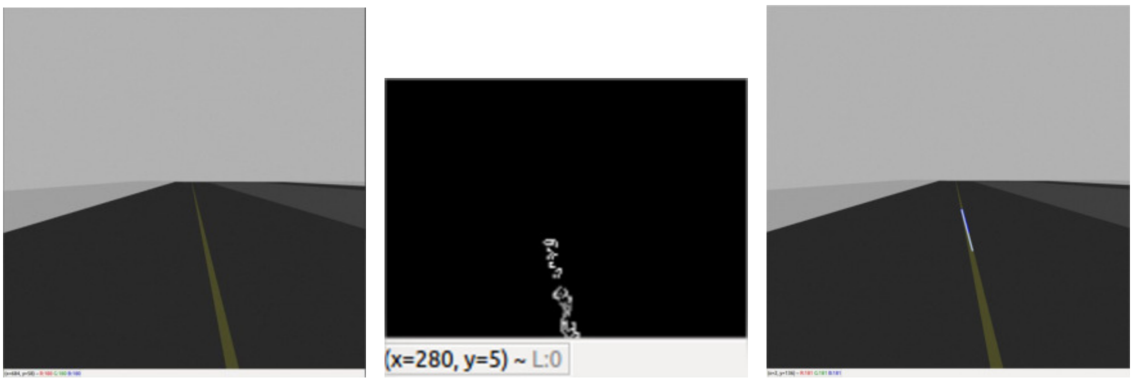
Input: Speed Set Point, Current Speed
Output: Throttle and Brake percentage

$\varepsilon^{\sigma} \leftarrow speed\_set\_point - current\_speed$
$\alpha \leftarrow PID(\varepsilon^{\sigma})$
$\alpha_{control} \leftarrow Normalized[-1\ldots1](\alpha)$
if $\alpha_{control} \geq 0$ **then**
    $throttle \leftarrow \alpha_{control}$
    $brake \leftarrow 0$
else
    $throttle \leftarrow 0$
    $brake \leftarrow \alpha_{control}$
end if

8.2.3.2 Line Follower

In this work, the vehicle will simulate a standard trajectory path-following method, using a road line [274]. The simulated car has several cameras, and one of them is used to identify the road line and follow it with real-time detection. The Line Follower (LF) algorithm processes the captured image and delivers information regarding the line position to the controller. The vehicle's controller will keep its center over the line with a second PID controller. The implemented algorithm is similar to the one proposed in [159]. Nevertheless, as Gazebo provides a realistic camera view, it is possible to implement algorithms without a real one, changing the image coordinates, frame rate, and data size, among other image capture characteristics, and evaluate the changes' impact on the controller.

An OpenCV node was implemented to read the data from the onboard front camera. This node subscribes to the topic $car_i/front\_camera$ and virtually receives all the images from the camera in an 800×800 pixels frame. Then, the LF algorithm filters the image to find a vertical line in the track, and the detection is performed using the Progressive Probabilistic Hough Transform (HT) [219]. This method is commonly used in image processing and can help detect any shape if it can be represented in mathematical form.



(a) Vehicle front camera view (b) Vector and position extraction (c) Line Detection - Final View

Figura 8.8: Line detection process

The Line Detection (LD) algorithm is illustrated in three frames of Fig. 8.8. The first one, in Fig. 8.8a, shows the vehicle camera simulated view. The LD algorithm applies a mask over this image to filter it, highlighting a particular color. This color can be adjusted following the Red Green Blue (RGB) model. The filtered image is then converted to a greyscale picture, as presented in Fig. 8.8b, allowing the edges detection using the Canny Edge detection [220], using vectors with Cartesian coordinates. Finally, these edges are integrated with the HT, defining a *most probably* line to be followed, as demonstrated in Fig. 8.8c.

The line coordinates are published in $car_i/line\_data$ topic and can be used by the LF controller module. This module is called *controller* and is responsible for the vehicle's motion controller. The vehicle heading error (ε^{θ}) related to the line reference is defined as a relative measurement, using the center of the image frame as illustrated in Fig. 8.9. In this figure, the *Detected Line*

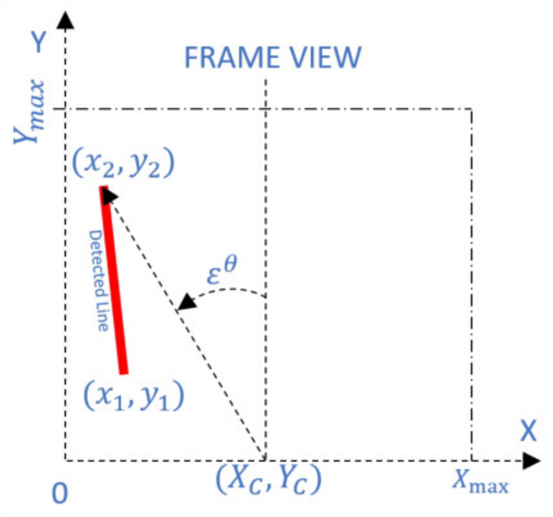


Figura 8.9: Vehicle Heading Error (ε^θ)

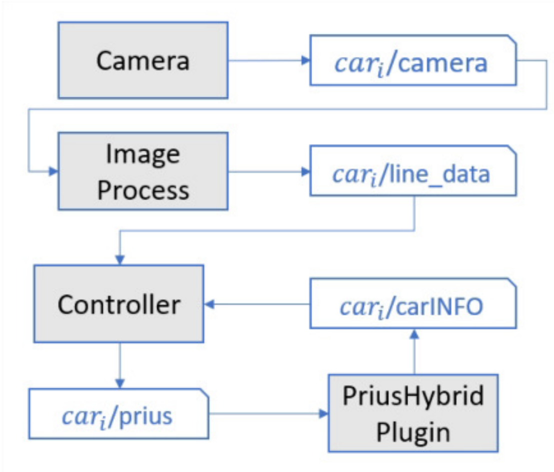


Figura 8.10: Line detection and Driving controller

is the output of the LD algorithm, with (x_1, y_1) and (x_2, y_2) respectively the initial and the final coordinates. The X_{max} and Y_{max} represent the frame limits and X_C is the frame center point in X axis. The ε^θ is defined as the angular difference between the X_C and the (x_2, y_2) coordinates, given by eq. 8.4. Finally, the *controller* calculates the car's *Steering Wheel Angle*, using the PID control action presented in 8.5.

$$\varepsilon^\theta(t) = \arcsin \frac{X_C - x_2}{y_2 - Y_C} \tag{8.4}$$

$$\theta_{wheels}(t) = K_P^\theta * \varepsilon^\theta(t) + K_I^\theta * \int \varepsilon^\theta dt + K_D^\theta * \frac{\Delta\varepsilon^\theta(t)}{dt}, \tag{8.5}$$

where K_P^θ, K_I^θ and K_D^θ denote the Proportional, Integrator, and Derivative gain constants, and θ_{wheels} is the *Steering Wheel Angle* to be applied to the vehicle. Figure 8.10 shows the general LF flowchart, including the controller action, while the complete LF algorithm can be observed in Alg. 2.

Algorithm 2 Line Follower Algorithm

Input: Image Frame
Output: Steering Angle

1: Mask image to find Vertical Lines
2: Filter image to obtain Data Vectors
3: $Line\_Vectors \leftarrow Hough\_Line\_Transform$
4: $Line\_Coordinates \leftarrow MERGE(Line\_Vectors)$
5: $\varepsilon^\theta \leftarrow eq.8.4$
6: $\theta_{wheels} \leftarrow PID(\varepsilon^\theta)$

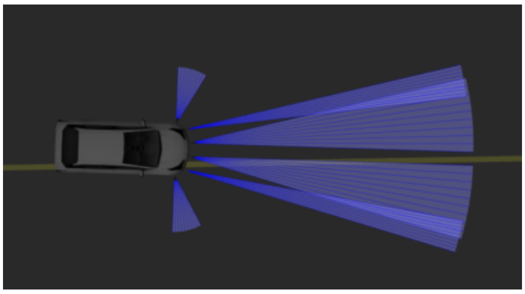


Figura 8.11: Sonar Visualization

8.2.3.3 Obstacle Detection and Avoidance

Obstacle detection and avoidance are among the most common autonomous vehicular applications, given the demanded safety conditions. So, in this simulator demonstration, sonars are used to detect and avoid unpredicted obstacles and help the vehicles to keep the LF algorithm. The car_i will use six sonars: four in the car's front and one on each side of it, as seen in Fig. 8.11. The simulator allows the user to change the sonar's positions and ranges and add or remove them in *prius.urdf* file. This simulator assumes that the obstacles are positioned near the reference line and have the same lateral size as the vehicles.

Algorithm 3 Line Follower with Detection and Avoidance Algorithm

Input: Sonars Info, Image Frame
Output: Steering Angle

while $Object\_Detected$ **do**
    if $Right\_Object\_Detected$ **then**
        $\theta_{wheels} \leftarrow Left\_Deviation$
    else if $Left\_Object\_Detected$ **then**
        $\theta_{wheels} \leftarrow Right\_Deviation$
    end if
end while
$Line\_Follower\_Algorithm$ (Alg. 2)

The sonars are used together with the LF algorithm. However, as the Detection and Avoidance (DA) algorithm has priority over the LF, it assumes the vehicle controller until the obstacle is out of view and the LF is reactivated. So, the algorithm 3 is an extension of the LF algorithm. When the sonars detect an obstacle, the DA controller turns the vehicle in the opposite direction, within a fixed θ_{wheels} value. This heading adjustment is continued until the four front sonars stop detecting the obstacle. Then, the lateral sonars avoid the vehicle trying to return to the line before it overtakes the obstacle. Finally, the LF algorithm uses the last information about the detected line to return to the desired trajectory. The DA block diagram is presented in Fig. 8.12.

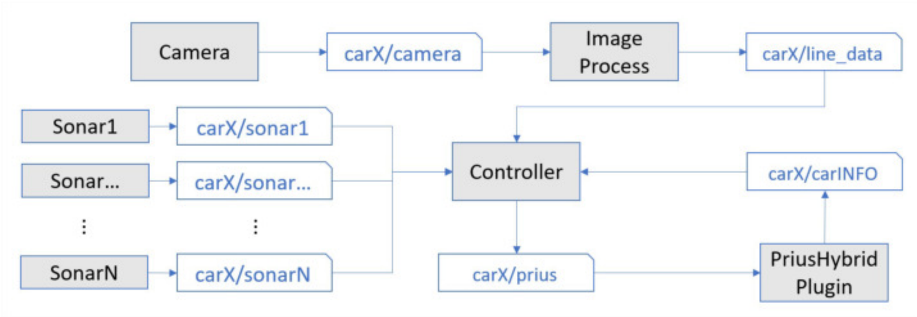


Figura 8.12: General Vehicle Architecture, with Line Detection and Detection and Avoidance modules

8.2.4 Experimental Validation

A control simulation environment should present several controller tools to the student. This section will introduce three primary devices developed in RosDrive that allow the student to analyze the vehicle's controller performance and elaborate on different strategies to guarantee its safety. The vehicle's controller performance can be defined in several ways, including fuel consumption, final speed, and acceleration. In this chapter, the performance is measured by the vehicle's capacity to track the setpoint, both in speed and heading adjustments.

8.2.4.1 Cruise Controller Implementation

The CC was developed as an independent module. So, it works as a black box implementation, where the inputs are the setpoint and current vehicle's speed, and the outputs are the throttle and brake percentage. At the same time, the controller parameters are adjusted inside the module. This architecture choice increases the simulator's flexibility, allowing the user to replace the controller and adjust its parameters.

Taking into account Fig. 8.3, the straight line between the points 7 and 1, without obstacles, was used to evaluate the CC and check how the vehicle behaves with several accelerations and decelerations. In this scenario, the vehicle speed setpoint was changed from 20.0 m/s, to 14.0 m/s, 16.0 m/s, 12.0 m/s and finally 0.0 m/s. All the speed settings are defined in the *controller.py* file in the parameters section. They are related to the vehicle's current position on the track.

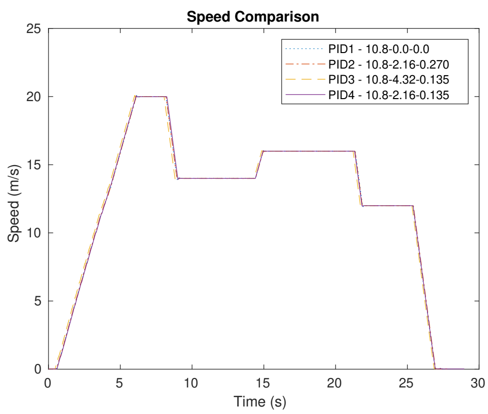
The controller parameters K_P, K_I, and K_D were defined with the Ziegler Nichols (ZN) empirical method [158]. The vehicle was accelerated from a rest position until it reached the first setpoint speed in the proposed scenario. Increasing K_P until the system oscillation limit, it was possible to determine the ultimate gain (K_u), at 18, with a 0.1 ms period. These values show how the vehicle's actuator has a rapid response since the oscillation period is fast. In this test, the Ziegler Nichols Tuning parameters are $K_P = 10.8$, $K_I = 2.16$, and $K_D = 0.135$. The system's response is presented in Fig. 8.13.

Tabela 8.1: Cruise Controller PID Settings

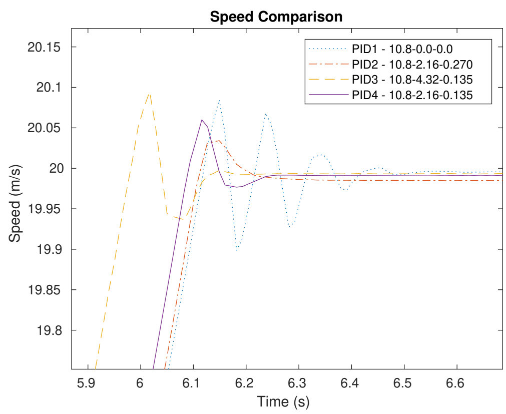
CC	K_P	K_I	K_D
$PID1$	10.8	0.0	0.0
$PID2$	10.8	2.16	0.270
$PID3$	10.8	4.32	0.135
$PID4$	10.8	2.16	0.135

As described above, the RosDrive was designed so that the student can change the system's characteristics and observe the impact on the vehicle's response. In addition to changing the controller model, the change of control parameters already implies different responses to be analyzed, providing the user with a practical study of the characteristics of each one of them. Three variations of the parameters obtained with ZN are proposed to exemplify their impacts on the vehicle's control action. These parameters are shown on the tab. 8.1, where $PID1$ is a proportional-only controller, $PID2$ increases the derivative component, $PID3$ enforces the integrator component, and finally $PID4$ shows the parameters obtained by the method ZN.

Figures 8.13a and 8.13b show in detail the impact of controller changes on the system response. In Fig. 8.13b it is possible to observe that the control proportional-only ($PID1$) presents a more significant oscillation and that the increase of the derivative component ($PID2$) makes the response slower, but with a smaller overlap. On the other hand, the increase in the integrator component ($PID3$) makes the system faster but with a greater overshoot on the setpoint. In this scenario, the parameters obtained by ZN show a better response as they are at an intermediate point of response time and overshoot. The controller error was minimal in all models, with a maximum steady-state value lower than 0.006 m/s.



(a) Vehicle's Speed Response



(b) Vehicle's Speed Detail Response

Figura 8.13: Vehicle Speed Response to different PID parameters

The PID CC example shows the student the basic vehicle controller models. It translates the conceptual controller view to a practical application, reducing the gap between the theoretical aspects and the implementation one, allowing the development of active skills and ope-

Tabela 8.2: HC PID parameters

HC	Speed (m/s)	K_P^θ	K_I^θ	K_D^θ	HC	Speed (m/s)	K_P^θ	K_I^θ	K_D^θ
REF	13	10.0	0.0	0.5	$PID8$	15	10.0	1.0	1.0
$PID5$	15	10.0	0.0	0.5	$PID9$	15	10.0	0.5	0.0
$PID6$	15	10.0	0.0	1.0	$PID10$	15	10.0	0.5	0.5
$PID7$	15	10.0	0.0	1.0	$PID11$	15	10.0	1.0	0.0

ning the doors to the student's creativity. A shortly CC RosDrive demonstrator is presented in https://youtu.be/QFVwgFyhaF4.

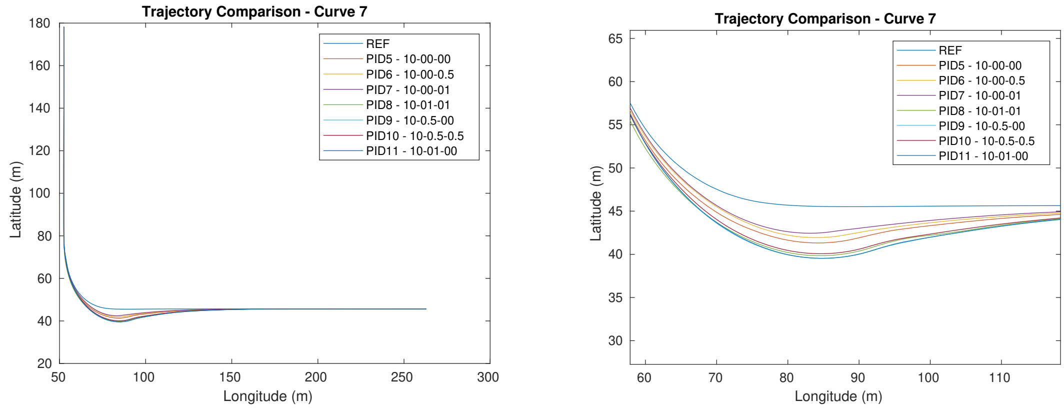
8.2.4.2 Line Follower (LF) Controller

The LF controller is responsible for the vehicle's heading adjustment, performed by the Heading Controller (HC). This control ensures that the car safely makes the circuit curves, preventing accidents. As can be seen in Fig. 8.3, in this scenario, tighter turns were chosen, allowing the student to analyze more complex situations, such as car skidding. Under these conditions, the vehicle's controller is adjusted in one curve and then evaluated its performance on the whole circuit.

Initially, the circuit's curves radius were analyzed to define the maximum speed that would prevent the vehicle from going off at the curve's tangent. The maximum speed (v_{out}) is given by $|v_{out}| = \sqrt{\mu \cdot |g| \cdot R}$, where μ is the friction's coefficient, $|g|$ is the gravity acceleration and R is the curvature ray. In the proposed scenario, it is defined that $\mu = 0.9$, $g = -9.8\ m/s$, and $R = 18.38$ m, which means that $v_{out} = 13.34\ m/s$.

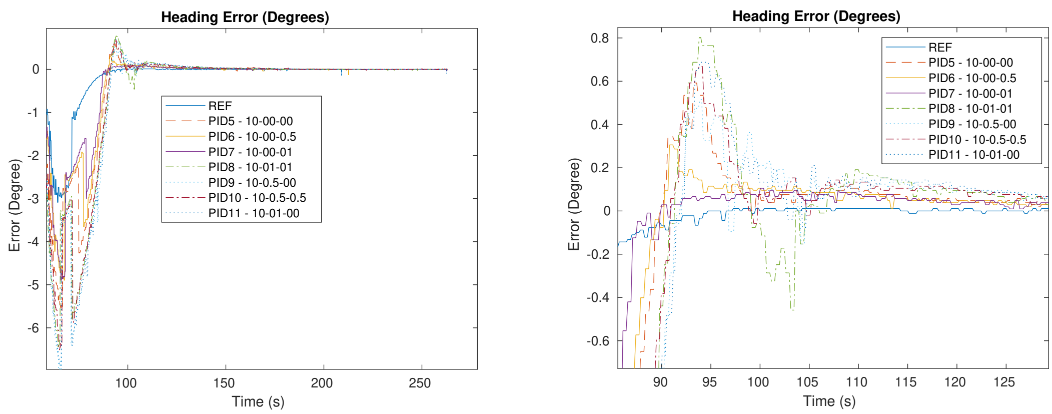
The heading controller has a different evaluation in comparison with the CC. In the CC, the setpoints are defined through a step function, while in the HC, the setpoints function follows the curve design, with a long transition phase. The system's response should be evaluated after the desired heading is constant. Due to this condition, a more complex scenario is proposed to assess the system's response in adversarial situations. Initially, the vehicle's trajectory was fixed with v_{out} as the heading reference. Then, the objective was to find the most suitable HC PID parameters for the system with $v = 15\ m/s$. It means that the HC will suffer from skidding. In this scenario, the student's experience determining the best HC PID parameters will be necessary since the ideal conditions presented in theory are not present. Furthermore, it will increase the student's perception of the problem and stimulate creative new solutions since the vehicle's speed increase will increase skidding, compromising the system's stability. It is also important to highlight that the user can set up any speed and check its response.

The HC PID parameters were obtained initially in curve 7 given the long straight lines before and after. The obtained parameters are presented in tab. 8.2. Figure 8.14a presents the vehicle's trajectory on curve 7, while Fig. 8.14b perform an in-depth view of the same curve. Both figures illustrate how the reference HC has a smoother trajectory, with no skidding. There is some skidding in all the HC PID configurations, with a speed setpoint of 15 m/s. However, these figures analysis allow the identification of the best controller performance, even under these conditions. So, in the proposed scenario, the $PID7$ presents a better response due to the derivative action, avoiding extreme adjustments and keeping the vehicle's trajectory near the REF trajectory. On the other hand, the integral action presented in $PID8$ and $PID11$ configurations produces more oscillation and increases the distance between the REF and the performed trajectory due to the skidding.



(a) Vehicle's Trajectory - Curve 7 (b) Vehicle's Detail Trajectory - Curve 7

Figura 8.14: Vehicle Trajectory Analysis (Curve 7) under different HC PID Settings



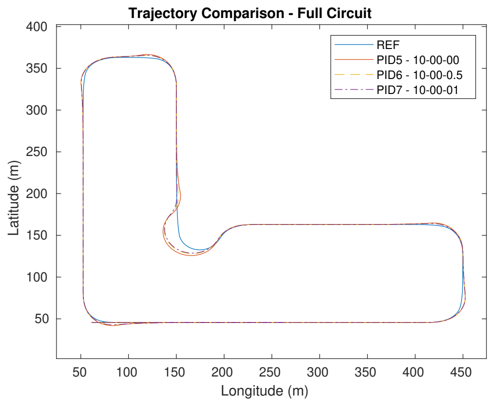
(a) Vehicle's Heading Error (ε^{θ}) (b) Vehicle's Heading Error (ε^{θ}) Detailed View

Figura 8.15: Vehicle Heading Error(ε^{θ}) at Curve 7 under different HC PID Settings

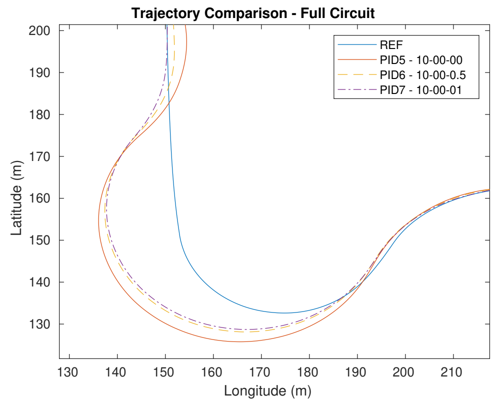
Figure 8.15 presents the vehicle's heading error (ε^{θ}), due to the different HC PID parameters. While the LF reference adjusts the heading setpoint, the vehicle's heading suffers from much oscillation, trying to respond to the new conditions. However, the system's response can be better studied after the transition, when the LF algorithm sets the new line. This situation can be observed in Fig. 8.14b. This figure highlights the smooth response of $PID7$, with little oscillation above the

REF response. Nevertheless, the systems' response with *PID*8 and *PID*11 have a considerable overshoot and take much more time until the stabilization.

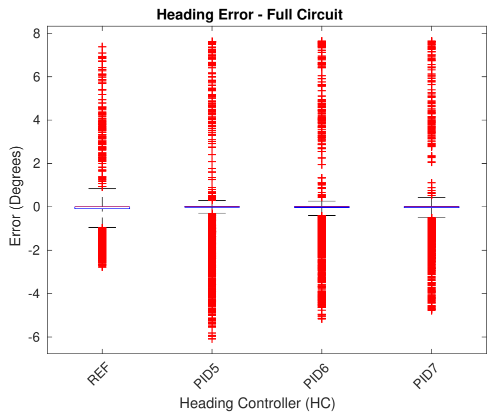
The HC PID response analysis can be extended to the entire circuit. Looking at the heading error (ε^{θ}) presented in Fig. 8.15, the best controllers response were performed by the PD configurations, namely the *PID*6 and *PID*7. Furthermore, a full lap was performed to evaluate the vehicle's heading controller, comparing its trajectory and the general ε^{θ}. Figure 8.16a present the vehicle's trajectory comparison in the full lap. It shows the vehicle's skidding in all the curves and the most distinguished one in curve 4. Thus, Fig. 8.16b highlights the vehicle's trajectory in this curve, showing that although all the HC PID configurations suffer from high skidding on this curve, the *PID*7 configuration provides a smaller skidding and is the faster one to stabilize the system after the curve. Finally, the Fig. 8.16c shows a comparison between the general ε^{θ} during the full lap. It demonstrates that the *REF* configuration has the smallest error variation during the circuit and that the *PID*7 error response is the most approximate to it.



(a) Vehicle's Trajectory - Full Lap



(b) Vehicle's Detail Trajectory - Curve 4

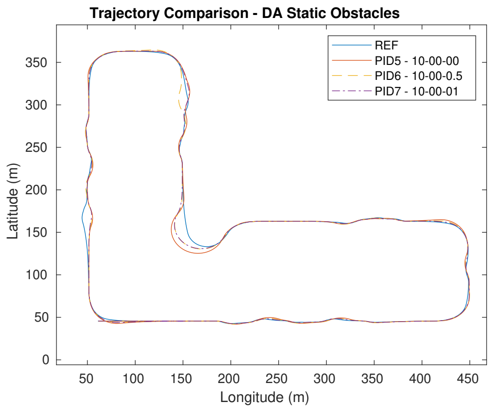


(c) Vehicle's Heading Error (ε^{θ}) at a Full Lap under best HC PID Settings

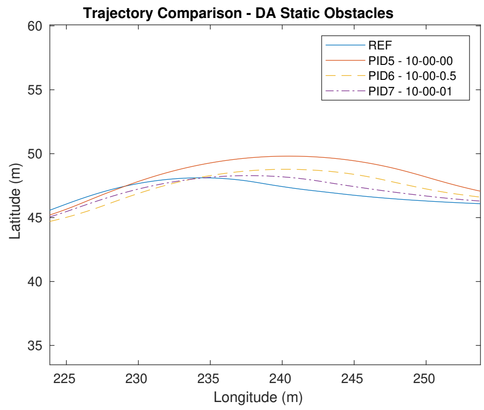
Figura 8.16: Vehicle Trajectory Analysis (Full Lap) under different HC PID Settings

This scenario was built to illustrate the simulator's flexibility, merging the LF algorithm with the HC method under an adversarial context. In this way, the student will be able to extend its capabilities, changing the controller's parameters and checking the system's response, proposing new situations, and evaluating them. Furthermore, it will help students build and reinforce their capabilities and skills without damaging any equipment by extrapolating the commonly encountered theoretical conditions.

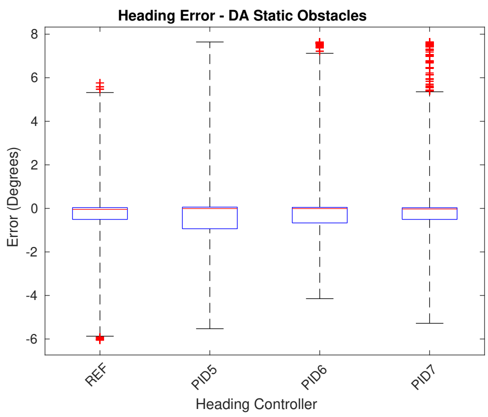
8.2.4.3 Obstacle Detection and Avoidance



(a) Vehicle's Trajectory Static Obstacle Avoidance - Full Lap



(b) Vehicle's Trajectory Detail Static Obstacle Avoidance



(c) Vehicle's Heading Error (ε^{θ}) with Static Obstacle Avoidance

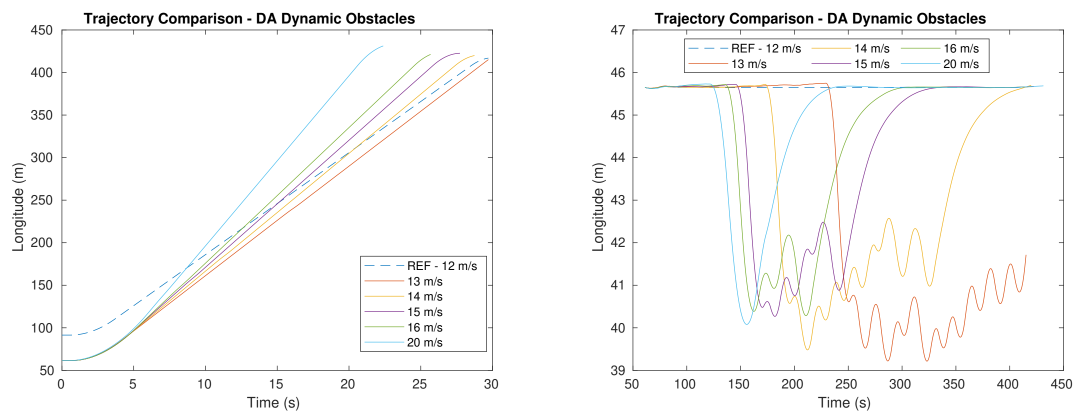
Figura 8.17: Vehicle Trajectory Analysis (Static Obstacle Avoidance) under different HC PID Settings

In addition to the analysis of the CC and HC controllers on the vehicle's performance in isolated scenarios, RosDrive allows the study of its interaction with other vehicles, whether static or dynamic.

Static Obstacle Detection and Avoidance The vehicle's ability to perform a trajectory avoiding several close obstacles was initially analyzed, comparing the heading error (ε^{θ}) given the HC parameters changes. The 19 static obstacles are illustrated in Fig. 8.3 and are modeled as Pickup vehicles. These obstacles are positioned on the straight circuit lines avoiding curves overtaking. Again, the REF vehicle running with a 13.0 m/s speed was presented against the $PID5$, $PID6$, and $PID7$ HC configuration, running at 15.0 m/s. The vehicle's trajectory is presented in Fig. 8.17a, illustrating that the skidding is still present, mostly in curve 4.

Furthermore, the vehicles do not necessarily follow the same trajectory to avoid obstacles. This situation is illustrated in the straight line between curves 4 and 5, where the vehicle with the HC $PID6$ avoids the last obstacle with a left turn and the others perform a right angle. This obstacle avoidance action responds to the first sonar activated in the vehicle. As the vehicle's trajectories are slightly different due to the HC response, the car's angular position at that point is not the same for all the configurations, providing sonar activation. The same situation is observed in the last obstacle between curve 7 with all the HC PID configurations compared with the REF.

To compare the HC PID's performance in a static detection and avoidance condition, Fig. 8.17b emphasizes the vehicle's movement over the second track obstacle. It is possible to observe that the HC PID performance follows the model presented in the LF algorithm, with $PID7$ providing the best system response compared to REF. However, the ε^{θ}'s variation between REF, $PID6$, and $PID7$ is similar, given the rapid heading transitions triggered by the ODA and the LF algorithms, as presented in Fig. 8.17c. On the other hand, the boxplots of $PID7$ ε^{θ} in both Figs. 8.16c and 8.17c show that although the maximum variation is similar, these errors appear more frequently, no longer presenting themselves as outliners but as values that are repeatedly perceived. These errors happen due to several obstacles and the constant need to adjust the vehicle's position caused by the HC action.



(a) Vehicle's Dynamic Obstacle Avoidance Longitude

(b) Vehicle's Dynamic Obstacle Avoidance trajectory

Figura 8.18: Vehicle Trajectory Analysis (Dynamic Obstacle Avoidance)

Dynamic Obstacle Detection and Avoidance RosDrive's flexibility allows different control models, algorithms, and movement strategies. Thus, it is possible to evaluate strategies for overtaking vehicles in motion, observing how the fastest vehicle behaves and if it can perform the maneuver safely. To conduct this demonstration, the car_1 and car_2, with $v_1(t) > v_2(t)$ were defined, with $x_2(0) = x_1(0) + 30m$. This way, car_1 has time to reach its maximum speed before the sonar detects the presence of car_2 and only then starts the overdrive process.

The *vehicles.launch* allows the setup and launches of the necessary vehicles with no additional development. In this file, the vehicle models, their initial positions, and the algorithms are instantiated. Initially car_1 (REF) and car_2 are defined with the same CC and HC parameters as $PID4$ and $PID7$. In the same file, the car_2 sonars are deactivated, avoiding its reaction to $car1$ presence. So, $car1$ accelerates, detects the presence of car_2, and performs the overtake action. As the same ODA strategy presented in section 8.2.3.3 is applied, after the obstacle detection, $car1$ will return to the line only after overtaking the obstacle when the lateral sonars indicate that there are no more obstacles there.

In this scenario, v_2 was defined as 13.0, 14.0, 15.0, 16.0, and 20.0 m/s and $v_1 = 12\ m/s$, while the front sonar's ranges are set to 20 m and the lateral ones are 2 m.

When theoretically studying physical systems, it is common to analyze that vehicles are points in space and that overtaking, for example, is just a matter of validating the relationship between space traveled in time, having as reference the speed of the two points. However, in a realistic simulator, vehicles cannot be treated as points in space but as bodies that can collide and must avoid this to remain safe. Thus, the overtaking process begins with detecting the body ahead, followed by a diversion action and consequent movement.

This controller action of car_1 is illustrated in Fig. 8.18. Fig. 8.18a illustrates a more simplified view of the system, indicating the longitudinal trajectory of car_1 and car_2 under all different velocity conditions of car_1. It is important to note that the movement of car_2 is the same in all scenarios, as its speed is constant, and its movement is not affected by car_1. In this figure, it is possible to observe the crossing point when the curve referring to $car1$ crosses the curve of car_2. Thus, it is seen that the speed of 13.0 m/s, car_1 is not able to exceed car_2 on the desired route, indicating an unsafe maneuver.

However, extending this view to a 2D dimension, as presented in Fig. 8.18b, it is possible to analyze how the overtake movement is performed. Under the proposed conditions, only the vehicle with a 20 m/s speed has overtaken car_2 with no oscillatory movement. With other speeds, car_1 lateral sonar detects car_2 presence, and adjust car_1 heading position, avoiding a collision. This figure illustrates how car_1 with a 13.0 m/s speed cannot overtake car_2 in the desired time and how the same movement with 14.0 m/s is unsafe since the overtaken process ends just at the limit of the desired trajectory.

So, the student has more information to check how the movement was performed and propose different safety strategies. In addition, this scenario allows the student to create new sonar detection algorithms, different controller strategies, and some intelligent systems to increase the system's safety.

8.3 WiCAR - Simulating towards the Wireless Car

In the past decade, Wireless Sensor Networks (WSN) have been widely adopted and supporting several innovative applications in a multitude of domains, such as in health, security, and agricultural. Nowadays, the increasing miniaturization of modern embedded systems, together with the advancements in the area of WSNs and energy harvesting, have opened up new possibilities to fit wireless communications into an unexpected series of applications. The automotive industry, has understandably been reluctant to adopt WSN, mostly pointing out its non-deterministic communication behaviour, unreliability due to interference and security issues. Therefore, wireless has been confined to some limited functionalities of infotainment systems and its adaption in critical systems has been non-existent in vehicles, although it has been already enabling a series of critical scenarios in other industrial domains.

The day-to-day automobile has gradually evolved from fully mechanical design to a fully electronically equipped modern car. The existing subsystems of a modern car consist of several sensors and actuators that are coupled with hundreds of Electronic Control Units (ECU) that are interconnected through thick wired harnesses and communicate based on real-time communication protocols. These wired harnesses can increase the overall weight of the car resulting reduction of the performance of the vehicle in terms of fuel consumption. Thus, the excess weight of the car also can be extrapolated to an environmental issue.

Current trend, is to continue to increase the number of application modules and complexity in the vehicle, by fitting newer models with improved advanced driving assistance systems (ADAS) to increase their safety. However, this effort is not being applied to the millions of older vehicles that will continue to share the roads in the next 15 years, partially due to the tremendous complexity involved in retrofitting such vehicles. Wireless communications can potentially become an enabling technology to support such possibility, considering its flexibility and ease of deployment, by exploring the innovative plug-and-play possibilities introduced by these networked sensor networks. Ideally, additional additional sensing arrays could be introduced into the vehicle with minimum complexity, and without requiring complex re-wiring. However, ADAS pose stringent requirements to a system's control and communications, in terms of timeliness and reliability, and these properties must be ensured by the communications technology. The improvements to the low-power, low-rate IEEE 802.15.4 standard [166], introduced by the .e amendment, enables interesting features such as guaranteed bandwidth, deterministic delay and several other improved reliability support via the introduction of multi-channel techniques. These characteristics turn this communication technology as a prominent candidate to support wireless ADAS as well as other non-critical applications.

However, to effectively test and validate these systems, there is a need for tools that can support the simulation of these complex communication infrastructures from the control and the networking perspective, focusing on the interplay between these two dimensions. This paper introduces a co-simulation framework that enables the simulation of an ADAS application scenario in these two fronts, analyzing the relationship between vehicle dynamics, i.e. speed and braking force, and the delay required for the system to operate safely, exploring the performance limits of different network configurations of the DSME protocol.

8.3.1 Related Work

The research community has continuously looked into the possibility of using Wireless Sensor and Actuator Networks (WSANs) in intra-car communication. One of the foremost motivation for its implementation is to reduce the weight of the car and increase the overall performance in terms of fuel economy and reliability. Researchers in [262] investigated the design aspects of WSANs in intra-car systems and if whether they could become a viable solution to partially replace or enhance current wired measurement and control subsystems.

In [180], authors used IEEE 802.15.4 Compliant and ZigBee RF Transceivers to create a Blind Spot Information System (BLIS). BLIS systems implemented by many car manufacturers (e.g., General Motors, Ford, and Volvo) are based on costly hardware components such as cameras and radars. The proposed intra-car system in this work was non-intrusive at the same time cost-efficient. This work provided important information on the ideal location for sensors in an intra-car system, which we have adopted in our intra-car scenario depicted in Figure 3.

Case studies such as [124] have proven that multi-hop has the potential for providing additional reliability, robustness, and energy usage improvements over existing single-hop approaches. In their study, they state that aggregating data in one or several processing centers in the vehicle is critical for the monitoring capabilities of the sensors, which are constrained by both energy and computational power. Multi-hop systems, despite its large overhead, can enhance system reliability, robust performance, and reduce communication energy. In our work, we look into a communication technology which features multi-hop and multi-channel capabilities and hence can enhance the performance of the network.

There have also been several simulation studies [62], [278] on implementing low power and low rate wireless sensor networks for intra-vehicle communications. These authors considered ZigBee to be a good candidate because of its mesh networking capabilities and low power consumption. Zigbee solves multi-path fading using Direct Sequence Spread Spectrum (DSSS) technology and interference resilience using Carrier Sense Multiple Access (CSMA). The propagation channel inside a vehicle is closed and is affected by the mechanical vibrations caused by the movement of the vehicle. Hence authors propose a simulation of the physical layer of the ZigBee network and the propagation channel inside a vehicle along with an adaptive equalizer at the receiver. Though Zigbee had mesh capabilities, determinism is not assured in such networks due to the usage of a contention-based mechanism for transmission. From our previous works [165], [167] we were able to confirm that DSME had the capability to communicate under strict time bounds and support time-critical applications. In this work, we rely on DSME which supports both a contention-based to be a possible candidate for intra-car communication systems.

8.3.2 Co-simulation Framework

In this work we built a Wireless-ADAS co-simulation framework that combines the network simulation capabilities of OMNeT++/INET and the ability to emulate the vehicle physics and sensors behaviour in 3D scenarios using the Gazebo robotics simulator. This will enable us to analyze the mutual impact between the control and the networking aspects. The integration is done over the Robotics Operating System (ROS), based our previous works in [295], [294] which focused on inter-vehicle communications (i.e. using ETSI ITS-G5) to enable a cooperative platooning function. A general Architecture for our framework is presented in Figure 8.19. The integration of the network model is supported by the openDSME open-source framework [151] to implement the DSME protocol on top of the IEEE 802.15.4 physical layer. Two kinds of nodes are implemented in OMNeT++/INET simulation: the sensor nodes and the sink, corresponding to 8 end-devices and a PAN Coordinator respectively. In the OMNET++/INET side, the displayed outward 8 nodes (sensor nodes - IEEE 802.15.4 End Devices) correspond to the wireless radar/sonar modules implemented in the Gazebo vehicle model to achieve a 360 degree coverage of the vehicle without any blind spots. At the center of the layout, the "sink"node (IEEE 802.15.4 PAN Coordinator) is also displayed and corresponds to the Application Unit (AU) wireless interface. The AU is responsible for the ADAS system control implementation. It processes the sensor inputs and reacts accordingly, by interfacing the vehicle's steering and braking systems.

8.3.2.1 Data Workflow

In order to support data flowing between the Gazebo and OMNeT++ simulators, the ROS publish/subscribe middleware support was crucial. For each node in the OMNET++/INET simulation, there is a corresponding sensor in the Gazebo vehicle model which publishes its relevant data into a rostopic i.e. "/car1/sensors/sonar1". In the OMNET++/INET side, each node subscribes to the corresponding rostopic and prepares a message that is en-queued into the openDSME MAC layer to be transmitted to the sink node, which role is assumed by the network PAN Coordinator. OpenDSME handles the transmission and, if successful, the sink node publishes a rostopic with the sensor data that is subscribed by the AU. The AU then uses this input to feed its control loop. As for the Gazebo model, a Toyota Prius car model (visible at fig.8.19) is used as the baseline deployment for this WSN layout with seven sonars and a radar. With this general layout architecture, different ADAS scenarios can be implemented, by changing sensors or their characteristics, the vehicle model, the track and the surrounding environment, enabling the possibility to extensively test and validate a ADAS behaviour and explore its performance limits pre-deployment.

For the upcoming ADAS, vehicles are increasingly being equipped with a wide variety of sensors, in order to get a good awareness of their surroundings. In addition, Sensors are already being deployed in current ADAS to evaluate the status of some of the vehicle components (i.e., steer, brakes) to detect stress and prevent any failure. In this framework, all these sensors, can be implemented in a vehicle model, and later be integrated into the network model as a new node that feeds data into the AU, for a integrated perspective of the system on a multitude of scenarios.

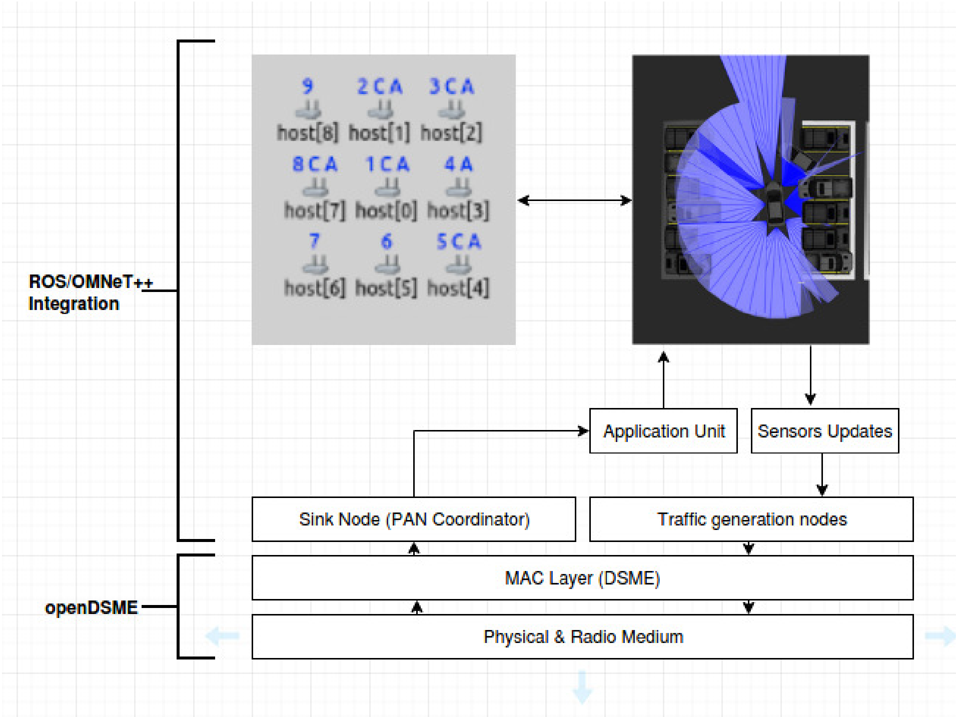


Figura 8.19: Integration Architecture

8.3.3 Network Specification

For our intra-car system, we used the DSME MAC behavior of IEEE 802.15.4e because of its deterministic capabilities. The DSME network provides deterministic communication using its beacon-enabled mode. This mode is supported by multisuperframes that may contain stacks of superframes, as shown in Fig. 2. Each superframe comprises a Contention Access Period (CAP) in which the nodes contend to access the channel and a Contention Free Period (CFP) in which the nodes send the data using Guaranteed timeslots (GTSs). It is in this period that the vehicle's sensors are accommodated, for guaranteed service.

The superframe is defined by BO, the Beacon Order which is the transmission interval of a beacon in a superframe. MO is the Multi superframe Order that represents the Enhanced Beacon interval of a multi-superframe, and SO is the Superframe Order that represents the beacon interval of a superframe. The number of superframes in a multisuperframe is given by 2^{MO-SO}. These values are conveyed to the nodes by an Enhanced Beacon (EB) at the beginning of each Multisuperframe. Reducing the values of SO and MO reduces the size of the timeslots and the number of superframes in a multi superframe duration, but also decreases the network's latency. In what follows we evaluate the relationship between such network settings and latency in the context of a ADAS application as a proof-of-concept.

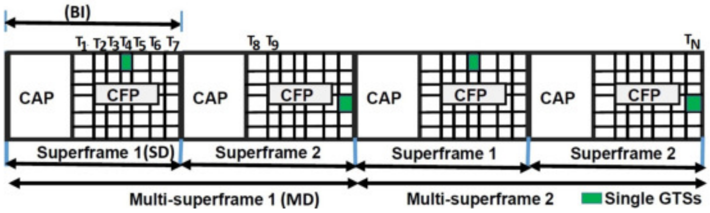


Figura 8.20: DSME superframe

8.3.4 Performance analysis

To showcase our proposal and simulation tool, we evaluate a parking lot Wireless ADAS scenario presented in Figure 8.21. When roaming inside a parking lot while searching for a parking spot, a driver can suffer from decreased perception of the overall environment. As his attention diverges from the driving actions into his visual search for the parking space, his ability to respond to unexpected situations is hindered, and may not be capable of perceiving an obstacle in time to avoid it. In this case, we consider the obstacle as a car that suddenly exits a parking space form the right-hand side of our vehicle. We push the requirements of the scenario to a point in which a typical driver would be unable to stop the car in time due to his reaction times. In this scenario, we consider the car can be traveling up to 30 Km/h (typical maximum speed inside a parking lot) and is fitted with an array of sensors covering a 360 degree field of view.

We evaluate this scenario from the two complementary perspectives. Firstly, we take the application perspective, by varying the braking capacity of the vehicle and its speed, and then the network perspective, by varying the MO and SO settings, and thus its worst-case delay. This is one of the greatest advantages of our co-simulation tool, which enables a multi-dimensional assessment of an application scenario.

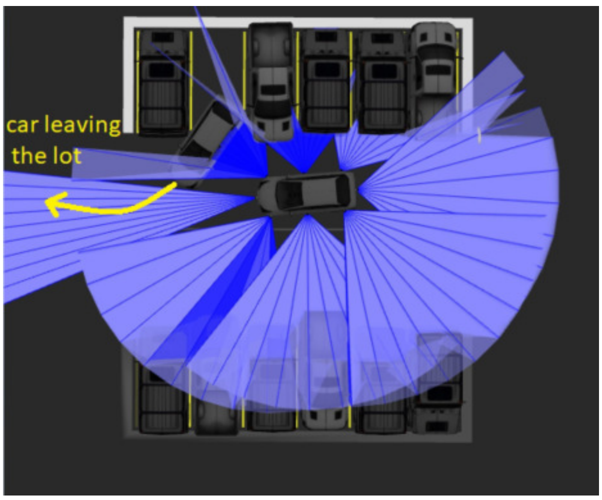


Figura 8.21: Scenario taken for evaluation

8.3.4.1 Impact of braking force

Braking capacity is one of the common parameters in any car that deteriorates over time. This is a result of the loss of friction in the clamping mechanism while actuating a brake. In a 100% operational brake, the clamping load is assumed to act on all friction surfaces equally. The loss in this force is only generated when the wheel does not lock because the friction of a sliding wheel is much lower than a rotating one.

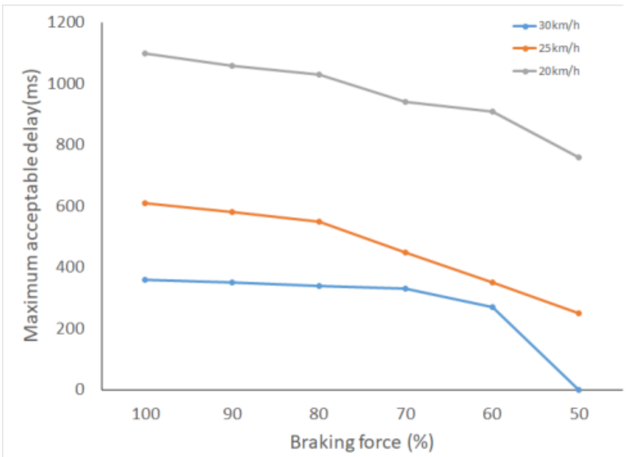


Figura 8.22: Maximum acceptable delay for the braking force applied

In this experiment, we study the limits of this system by averaging the results for several trials for different braking forces and calculating the maximum acceptable delay for the vehicle to operate without a crash. From the results in Figure 8.22, it is evident and expected, that the braking force and vehicle speed impose different requirements into the network delay. Decreased braking capacity or higher speeds demand lower communication delays to avoid the crash. At 30 km/h, with a 50% braking capacity, the vehicle is unable to avoid hitting the car leaving the parking space, independently of the delay. This is the point where we reach the performance limit of the control system as dictated by vehicle dynamics.

8.3.4.2 Impact of network settings

We carried out several trials for these application settings, and different network MO/SO settings, to explore the performance limits of the Wireless ADAS scenario. Figure 8.23 presents the communication's delay tolerances, for different speeds (25 and 30 km/h) and braking capacities (100% to 50%), to prevent a crash, superimposed by the overall bounded delay at different network settings.

As observed in Figure 8.23, if the vehicle travels at 25 km/h in the parking lot, and has its braking capacity at 80%, it can still allow approximately 550 ms of delay in the ADAS communications; therefore, a (BO/SO/MO) = (6/4/5) setting suffices. This is important considering the usage of a higher MO can support the allocation of additional superframes and support additional nodes, particularly if CAP reduction is activated, increasing the scalability of the system. Thus

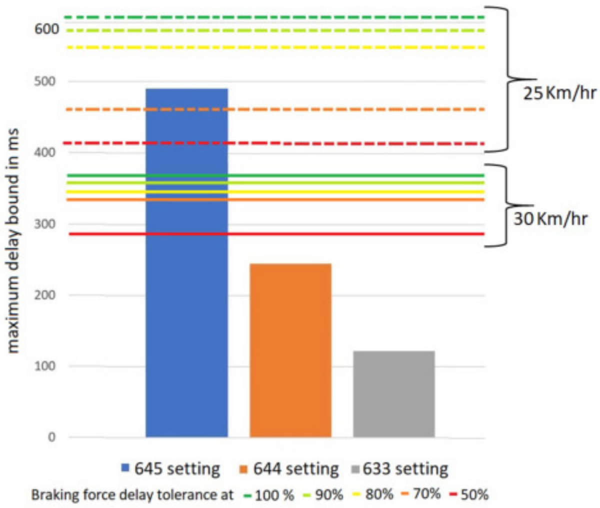


Figura 8.23: Impact of static scheduling and braking force on the crash rate

finding this trade-of between delay and scalability, in parallel with speed and braking capacity, can lead to increased efficiency and safety. When the braking capacity reduces to 70 or 60 %, the maximum acceptable delay decreases steeply and can only be met by lower MO/SO network settings. This is also the case for a speed of 30Km/h, that even at 100% braking capacity, only (BO/SO/MO) = (6/4/4) settings or lower, can meet the imposed delay requirement of approximately 360 ms. These results show us that for those settings, at the targeted speed for our scenario of 30 km/h, our system can still guarantee the safety of the vehicle even with its braking capacity impaired by 50%

8.3.4.3 Impact of delay

One of the important prerequisite for the safe functioning of this scenario is the ability to adhere to a maximum speed limit of 30 Km/h. To achieve this we must be able to provide a maximum delay bound of 350 ms which is a crucial aspect for safe functioning. Hence, we must verify the determinism of the network. In the worst-case scenario, the maximum time a superframe can take to accommodate a transmission will be the size of the superframe. Hence by varying the size of the superframe, we will be able to control the latency of the network and determine definite bounds. The following experiment is carried out with (BO/MO/SO) = (6/4/4) setting with fixed static schedule. As previously mentioned, the results strictly adhere to the limit of the worst-case delay. We experience a maximum delay of 0.23s and it is bounded as seen in Figure 8.24. This also means we will be also able to operate the application at a steady speed of 30 km/h with a fixed delay using this setting.

The fluctuations of delay values in these static settings can be attributed to the arrival time of the packet. The packets that are served immediately with respect to its arrival result in a much lower delay. The worst-case delay is produced when the sensor data arrival happens at the end of the first superframe and gets scheduled for its adjacent superframe. One significant advantage of

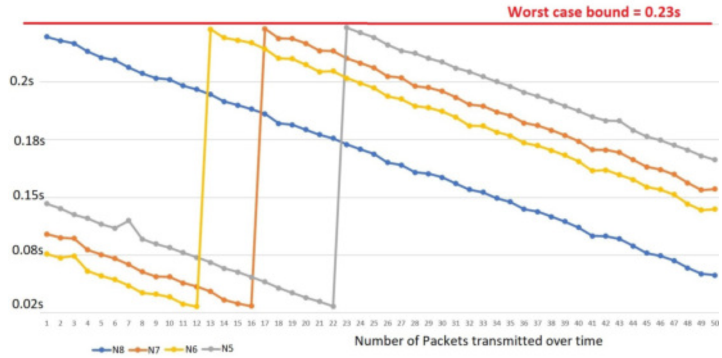


Figura 8.24: Delay with Static scheduling for MO=4 and SO=4

static scheduling is that the user has the possibility to vary the network settings and fix a steady worst-case bound based on the network prerequisite.

8.4 Conclusion

In this chapter we show two extended versions of CopaDrive, showing how it can be used in a teaching environment and how it can be used for the validation of different safety applications involving autonomous vehicles. With this, we were able to show that the foundations used in the tool allow its expansion to other scenarios involving CPS. Thus, its integration with HIL and the proposed testbed show even more validity, since the same bases used for the validation of control algorithms and analysis of network impacts used for the ETSI ITS-G5 model can be used for other communication protocols and different control strategies.

Capítulo 9

Conclusions and Future Work

This chapter summarizes the main conclusions presented in previous chapters, including the main challenges and some of the presented solutions. Otherwise, it also presents some future works to be developed in future research and possible new approaches for the solutions here presented.

9.1 Main Conclusions

Throughout this dissertation, we show how an integrated validation tool can contribute to the development of CPSs to increase their performance and safety. To do so, we take as a base an application of considerable impact in vehicular transportation, the Co-VP. However, it presents several challenges to its implementation in real environments due to the necessity of real-time communication and controller actions.

These challenges start in modeling the system, which must consider the inherent aspects of controlling multiple agents simultaneously. The control strategies must consider individual and collective actions to ensure the system's stability in the most varied conditions. These conditions range from the simplest movements, where the platoon must keep the distance between its members in straight lines, to more complex situations involving curves and obstacles. Thus, control systems have evolved to meet the demands of CPS systems, seeking more robust models capable of ensuring stability and maintaining their applicability to embedded systems.

The evolution of communication networks is also an essential pillar for constructing Co-VP systems. It is reinforced by the joint efforts of world organizations to adapt and define standards for these communications and studies that show their viability in disseminating information between vehicles. However, the lack of definition of a single model hinders some implementation efforts, limiting its applicability to a region. Thus, the choice is made for the most appropriate model for the study region, which can be easily adapted. In the case of this dissertation, ETSI ITS-G5 was chosen due to the European effort around its implementation.

The state of the art of Co-VP and Co-CPS applications shows that the gap between control systems and network performance studies is still a challenge, as these are analyzed as decoupled issues. However, the impact of communications on Co-VP systems is quite noticeable in their implementations, as delays and packet losses ultimately affect system security too much. Moreover, by assuming an open communication posture, this application is subject to malicious or non-malicious incursions that may affect the received messages and cause accidents due to security flaws.

Therefore, the joint analysis of aspects related to control and communication impacts on Co-VP systems shows how parameters such as PLR and Delay impact the security of the platoon. For example, message flow control over the communication channel minimizes packet collisions, ensuring that more data is delivered reliably. At the same time, the control systems must be modeled within limits that can withstand these failures for some time or even detect them to avoid accidents and take safety measures.

Because of the complexity involved in ensuring that these safety actions are carried out, validation of Co-VP systems is seen as a critical action. Until a solution is ready to be applied in an actual vehicle, measures must ensure the model's reliability. Therefore, simulators, hybrid models, and even scale models must be carefully thought out to meet the demands of each system and ultimately mimic real-world conditions to ensure the applicability of the algorithms and protocols analyzed in case studies.

In this context, CopaDrive was presented as proof of the statement posed earlier about using an integrated framework for validating Co-VP systems. Thus, we show in this thesis that using a tool based on ROS allows us to validate different control system performances while investigating communications' impacts. To do this, we develop the framework and validate the same control algorithm under various implementation aspects. We compare the impacts of sending CAM messages through simulation, defined by ETSI ITS-G5 at fixed frequencies or triggered by triggers. In this model, we show the network occupancy and prove the efficiency of the trigger-triggered models, as they occupy less of the network under simpler conditions.

Extending this analysis to validate safety devices, we implemented HIL based on the simulator and OBUs using the ITS-G5 stack. This integration was accomplished using a bridge between the ROS and the OBUs, capable of translating the messages sent from one side to the other. In this test, it was possible to observe the communication frequency of these OBUs and evaluate their operation. Furthermore, the same control model was used in both simulations because of the framework integration, which reduced the migration time between the platforms.

The same framework also allowed the validation of the proposed control model on a three-car robotics testbed. In this model, the on-board controller was again kept, and the bridge between the ROS and the OBUs. Thus, after the necessary adaptation of the vehicle sensors and information capture, it was possible to evaluate the ability of the vehicles to follow each other in a Co-VP application safely, closing the development cycle.

Using the framework, it was possible to develop different control model optimizations to increase the Co-VP application's safety. So, the integration problem between the longitudinal and lateral controls of the platoon was addressed. Since the Co-VP vehicles will easily be subject to different maneuvers, neglecting their lateral control is not a valid option. So, we present an integrated controller capable of solving the cutting-corner problem, keeping the vehicles aligned even after complex turns and obstacle deflections. Furthermore, the extension of this controller to a Look-Ahead model further increased the maximum size of the platoon while maintaining its stability and decreasing distance and heading errors.

From a control model validated from the communications perspective, it was possible to guarantee the second part of the proposed thesis statement. Thus, we used CopaDrive to show that the profiles used in ETSI ITS-G5 can be restricted for Co-VP applications, increasing their performance and reducing lateral and longitudinal errors. Furthermore, we show that this restriction does not significantly increase congestion or packet loss in the network, ensuring application security. Soon, we were able to lead the platoon traversing complex scenarios with higher quality, performing tricky turns, and avoiding obstacles.

Finally, we show that CopaDrive can be used in several vehicular systems applications. Thus, we build a model to be used in the classroom, allowing the validation of different control models and closing the gap between theory and practice in control teaching. On the other hand, we also show its ability to use other communication models, such as IEEE 802.15.4e, for intra-vehicular communications.

For all the above, we believe that this thesis has met the proposed objectives, presenting a flexible framework capable of validating different models of Co-VP systems. Moreover, this thesis showed how this framework served as a basis for validating a control and communication model that optimized the performance of Co-VP systems, increasing their safety.

9.2 Future Research Directions

Co-VP applications have been increasing in complexity and safety requirements. Thus, we believe the research presented here has three directions concerning the topics introduced. We hope to integrate an LTE C-V2V communication module starting with the framework. We consider this an exciting alternative to ETSI ITS-G5 that meets the European Union requirements and can work in parallel, acting as a backup for each other.

Although CopaDrive is currently associated with the Gazebo, its modular and ROS-based structure allows other 3d robot simulators to be used. Two of these models thought for future integration include the use of CARLA and Webbots. CARLA presents a very light and interesting interface, while Webbots presents a new model of integration with ROS that shows a new way forward. Both scenarios can be compared with the current model, changing part of the integration paradigm.

Still, within the framework, we see the need to evaluate the system's ability to respond to external stimuli, such as a USR. Thus, it will be possible to analyze a Co-VP project integrating V2V and V2I communication in more complex scenarios than those presented here. This implementation will also include using different OBUs, through integration with HIL, to compare the response capacity of different types of ITS-G5 implementations. We are also looking for the implementation of tools to monitor the system in real time, seeking to detect flaws in the vehicle itself that may cause it to work improperly, increasing the possibility of accidents.

We understand that improving the positioning system of each robot contributes to a better implementation of the control system. In this way, a future work is improving this positioning system and the speed control of the vehicle. Thus, it will be possible to reduce the distance between the vehicles during the platooning and increase their speed.

We believe that the PID approach is simple and efficient. However, the expansion of CopaDrive to use different control models is a natural direction for future projects. In this line of research, a more modern implementation using an MPC model can serve as a comparison for the PID system implemented here in terms of performance and response time. Artificial intelligence systems can also add to the lateral and longitudinal control of the platoon, reducing oscillations in the movement settings of these vehicles. This implementation also has the potential to solve a possible loss of messages, allowing the vehicle controller to take actions to ensure the safety of the system, even if communication is failing.

On the other hand, the evaluation of the communication models used allowed us to observe the effects of communication failures on the vehicles. Thus, the implementation of a communication failure detection layer can add to the system, ensuring a correct and fast action in case of failure. In this sense, a real-time detection of communication delays or packet losses is one of the future works proposed to complement the work done here. This detection can be extended to the treatment of communication security, minimizing the possibility of malicious interference that may represent attacks on the Co-VP system. This model will be based on a modular safety and security layer that can be inserted in the system, independent of the chosen communication model [100]. Such a layer is already being developed and has shown good results by adding safety and security information to the messages and decoding them at each reception.

Finally, in terms of an expanded tool, we believe that CopaDrive can be effectively evaluated as a teaching tool. To this end, we direct part of our future research to the simplification of interfaces and modularity of its components. Thus, we already have work in progress for its integration with other communication models like the Visible Light Communication (VLC) and also with the creation of interfaces for modifying the control parameters.

Anexo A

CopaDrive Install Instructions

A.1 Repository

In this section, we present how to download the tools and CopaDrive Simulator repository. We also present the necessary steps in order to run the system.

A.1.1 Main Requirements:

- OS: Ubuntu 18.0
- ROS: Melodic
- 3d Simulator: Gazebo 9
- Network Simulator: Omnet++ 5.6

A.1.2 Setup Project

1. Install ROS Melodic following the instructions of:
   http://wiki.ros.org/melodic/Installation/Ubuntu.

2. Install Omnet++ following the instructions of:
   https://doc.omnetpp.org/omnetpp/InstallGuide.pdf

3. Download files from Github

   CopaDrive Simulator: https://github.com/enioprates/copaDrive

   Artery: https://github.com/enioprates/artery

4. CopaDrive Setup

   - **CopaDrive Simulator**

   Delete ".cache"files from .../CISTER_car_simulator/Build folder

   Open a new terminal inside .../CISTER_car_simulator

Type: catkin_make

- **CopaDrive Simulator Controllers**

Delete ".cache"files from .../CISTER_image_processing/Build folder

Open a new terminal inside .../CISTER_image_processing

Type: catkin_make

- **Artery Compilation**

Delete ".cache"files from .../artery/Build

Open a new terminal inside .../artery

make vanetza

make inet

make veins

delete /build

Open a new terminal inside .../artery

mkdir build

cd build

cmake ..

A.1.3 How to run CopaDrive Simulator

1. Run the simulator:

   Open a new terminal inside .../CISTER_car_simulator

   source devel/setup.launch

   roslaunch car_demo demo_t.launch

2. PAUSE the simulation and reset the time!

3. Starting the Vehicle Controllers:

   Open a new terminal inside .../CISTER_image_processing

   source devel/setup.launch

   roslaunch image_processing vehicle.launch

4. Starting the Network Simulator (Omnte++):

   Open a new terminal inside .../artery

   cmake –build build –target run_gazebo-platoon

   Start the Omnet++

   Start the Gazebo simulation

Referências

[1] 3GPP. 3GPP TS 23.303 V15.1.0 - 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Proximity-based services (ProSe); Stage 2 (Release 15). Technical Specification TS 23.303, 3rd Generation Partnership Project, junho 2018.

[2] 5GAA Automotive Association. Deployment band configuration for C-V2X at 5.9 GHz in Europe. Position paper, 5GAA Automotive Association, 2021.

[3] AB Volvo. Volvo Trucks and FedEx demonstrate Truck Platooning, junho 2018.

[4] K. Abboud, H. A. Omar, e W. Zhuang. Interworking of DSRC and Cellular Network Technologies for V2X Communications: A Survey. *IEEE Transactions on Vehicular Technology*, 65(12):9457–9470, dezembro 2016. Conference Name: IEEE Transactions on Vehicular Technology.

[5] T. Acarman, Yiting Liu, e U. Ozguner. Intelligent cruise control stop and go with and without communication. Em *2006 American Control Conference*, páginas 4356–4361, Minneapolis, Minnesota, USA, junho 2006. American Control Conference.

[6] ADACORSA Consortium. ADACORSA, maio 2022. https://adacorsa.eu/.

[7] Michael Aeberhard, Thomas Kühbeck, Bernhard Seidl, e et al. Automated Driving with ROS at BMW. Em *ROSCon Hamburg 2015*, Hamburg, Germany, setembro 2015. Open Robotics.

[8] Aljawharah Alnasser, Hongjian Sun, e Jing Jiang. Cyber security challenges and solutions for V2X communications: A survey. *Computer Networks*, 151:52–67, março 2019.

[9] Mani Amoozadeh, Bryan Ching, Chen-Nee Chuah, Dipak Ghosal, e H. Michael Zhang. VENTOS: Vehicular Network Open Simulator with Hardware-in-the-Loop Support. *Procedia Computer Science*, 151:61–68, 2019.

[10] Mani Amoozadeh, Hui Deng, Chen-Nee Chuah, H. Michael Zhang, e Dipak Ghosal. Platoon management with cooperative adaptive cruise control enabled by VANET. *Vehicular Communications*, 2(2):110–123, abril 2015.

[11] Mani Amoozadeh, Arun Raghuramu, Chen-nee Chuah, Dipak Ghosal, H. Michael Zhang, Jeff Rowe, e Karl Levitt. Security vulnerabilities of connected vehicle streams and their impact on cooperative driving. *IEEE Communications Magazine*, 53(6):126–132, junho 2015.

[12] André de Matos Pedro. *Dynamic contracts for verification and enforcement of real-time systems properties*. PhD, Universidade do Minho, maio 2018.

[13] Maytheewat Aramrattana, Tony Larsson, Jonas Jansson, e Arne Nåbo. A simulation framework for cooperative intelligent transport systems testing and evaluation. *Transportation Research Part F: Traffic Psychology and Behaviour*, 61:268–280, fevereiro 2019.

[14] G. Araniti, C. Campolo, M. Condoluci, A. Iera, e A. Molinaro. LTE for vehicular networking: a survey. *IEEE Communications Magazine*, 51(5):148–157, maio 2013. Conference Name: IEEE Communications Magazine.

[15] Aqsa Aslam, Pedro M. Santos, Frederico Santos, e Luís Almeida. Empirical Performance Models of MAC Protocols for Cooperative Platooning Applications. *Electronics*, 8(11):1334, novembro 2019.

[16] Rachid Attia, Rodolfo Orjuela, e Michel Basset. Combined longitudinal and lateral control for automated vehicle guidance. *Vehicle System Dynamics*, 52(2):261–279, fevereiro 2014.

[17] Henry-Joseph Audéoud e Martin Heusse. Quick and Efficient Link Quality Estimation in Wireless Sensors Networks. Em *2018 14th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, páginas 87–90, fevereiro 2018.

[18] R. Avudaiammal, K. Jasmine Mystica, K. C. Akella, M. Arun Gokul, e R. Samuel. Bidirectional Vehicle Platooning Based Intelligent Transportation System. Em *2020 International Conference on Innovative Trends in Information Technology (ICITIIT)*, páginas 1–6, Indian Institute of Information Technology Kottayam, fevereiro 2020. IEEE.

[19] Ali Balador, Chumeng Bai, e Foroogh Sedighi. A Comparison of Decentralized Congestion Control Algorithms for Multiplatooning Communications. Em *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, páginas 674–680, Kyoto, Japan, março 2019. IEEE.

[20] Kritika Bansal e Pankaj Mukhija. Event-triggered control of vehicular platoon system with time-varying delay and sensor faults. *Journal of Automobile Engineering*, 234(14):11, 2020.

[21] P. Barooah e J.P. Hespanha. Error Amplification and Disturbance Propagation in Vehicle Strings with Decentralized Linear Control. Em *IEEE Conference on Decision and Control, 2005*, páginas 4964–4969, dezembro 2005. ISSN: 0191-2216.

[22] M. H. Basiri, N. L. Azad, e S. Fischmeister. Attack Resilient Heterogeneous Vehicle Platooning Using Secure Distributed Nonlinear Model Predictive Control. Em *2020 28th Mediterranean Conference on Control and Automation (MED)*, páginas 307–312, Saint-Raphaël, France, setembro 2020. MED. ISSN: 2473-3504.

[23] Mohammad Hossein Basiri, Mohammad Pirani, Nasser L. Azad, e Sebastian Fischmeister. Security of Vehicle Platooning: A Game-Theoretic Approach. *IEEE Access*, 7:185565–185579, 2019. Conference Name: IEEE Access.

[24] Mohammad Hossein Basiri, Mohammad Pirani, Nasser L. Azad, e Sebastian Fischmeister. Security-aware optimal actuator placement in vehicle platooning. *Asian Journal of Control*, n/a(n/a):1–18, março 2021.

[25] A. Bayuwindra, O L. Aakre, J. Ploeg, e H. Nijmeijer. Combined lateral and longitudinal CACC for a unicycle-type platoon. Em *IEEE Intelligent Vehicles Symposium (IV), 2016*, páginas 527–532, junho 2016.

[26] A. Bayuwindra, J. Ploeg, E. Lefeber, e H. Nijmeijer. Combined Longitudinal and Lateral Control of Car-Like Vehicle Platooning With Extended Look-Ahead. *IEEE Transactions on Control Systems Technology*, páginas 1–14, 2019.

[27] Richard Bishop. U.S. States Are Allowing Automated Follower Truck Platooning While The Swedes May Lead In Europe, fevereiro 2020. Section: Transportation.

[28] Felipe Boeira, Mikael Asplund, e Marinho P. Barcellos. Mitigating Position Falsification Attacks in Vehicular Platooning. Em *2018 IEEE Vehicular Networking Conference (VNC)*, páginas 1–4, Taipei, Taiwan, dezembro 2018. IEEE. ISSN: 2157-9865.

[29] Luciano Bozzi, Lorenzo Di Giuseppe, Luigi Pomante, Marco Pugliese, Marco Santic, Fortunato Santucci, e Walter Tiberti. TinyWIDS: a WPM-based Intrusion Detection System for TinyOS2.x/802.15.4 Wireless Sensor Networks. Em *Proceedings of the Fifth Workshop on Cryptography and Security in Computing Systems - CS2 '18*, páginas 13–16, Manchester, United Kingdom, 2018. ACM Press.

[30] Brandi Simonsen, Sarah Fairbanks, Amy Briesch, Diane Myers, e George Sugai. Evidence-based Practices in Classroom Management: Considerations for Research to Practice. *Education and Treatment of Children*, 31(1):351–380, 2008.

[31] C. Perkins, Ed., D. Johnson, e J. Arkko. Request for Comments: 6275, julho 2011.

[32] Hanwen Cao, Sandip Gangakhedkar, Ali Ramadan Ali, Mohamed Gharba, e Joseph Eichinger. A Testbed for Experimenting 5G-V2X Requiring Ultra Reliability and Low-Latency. Em *WSA 2017 - 21th International ITG Workshop on Smart Antennas*, páginas 1–4, Berlin, Germany, março 2017. IEEE.

[33] CAR 2 CAR Communication Consortium. Basic System Profile. RS 2037, CAR 2 CAR Communication Consortium, março 2020.

[34] CAR 2 CAR Communication Consortium. CAR 2 CAR Communication Consortium, dezembro 2020.

[35] C. F. Caruntu, C. Braescu, A. Maxim, R. C. Rafaila, e A. Tiganasu. Distributed model predictive control for vehicle platooning: A brief survey. Em *2016 20th International Conference on System Theory, Control and Computing (ICSTCC)*, páginas 644–650, Sinaia, Romania, outubro 2016. IEEE.

[36] Giammarco Cecchini, Alessandro Bazzi, Barbara M. Masini, e Alberto Zanella. Performance comparison between IEEE 802.11p and LTE-V2V in-coverage and out-of-coverage for cooperative awareness. Em *2017 IEEE Vehicular Networking Conference (VNC)*, páginas 109–114, Turin, Italy, novembro 2017. IEEE. ISSN: 2157-9865.

[37] Huajun Chai, H.M. Zhang, Dipak Ghosal, e Chen-Nee Chuah. Dynamic traffic routing in a network with adaptive signal control. *Transportation Research Part C: Emerging Technologies*, 85:64–85, dezembro 2017.

[38] Ratchatin Chancharoen, Angkee Sripakagorn, e Kuntinee Maneeratana. An Arduino kit for learning mechatronics and its scalability in semester projects. Em *2014 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, páginas 505–510, dezembro 2014.

[39] Hossein Chehardoli, Mohammad R. Homaeinezhad, e Ali Ghasemi. Control design and stability analysis of homogeneous traffic flow under time delay: A new spacing policy. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 233(3):622–635, fevereiro 2019.

[40] Hossein Chehardoli e Mr Homaeinezhad. Stable control of a heterogeneous platoon of vehicles with switched interaction topology, time-varying communication delay and lag of actuator. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 231(22):4197–4208, novembro 2017.

[41] Chen Chen, Jiange Jiang, Ning Lv, e Siyu Li. An Intelligent Path Planning Scheme of Autonomous Vehicles Platoon Using Deep Reinforcement Learning on Network Edge. *IEEE Access*, 8:99059–99069, 2020.

[42] JunJie Chen, Wei ShangGuan, Baigen Cai, Manoj Bhat, e Yu Du. Intelligent Platoon Operating Slot Optimization Method based on Drivers' Overtaking Behavior. Em *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, páginas 1947–1952, outubro 2019. ISSN: null.

[43] Na Chen, Meng Wang, Tom Alkim, e Bart van Arem. A Robust Longitudinal Control Strategy of Platoons under Model Uncertainties and Time Delays. *Journal of Advanced Transportation*, 2018:1–13, 2018.

[44] Tianshu Chu e Uros Kalabic. Model-based deep reinforcement learning for CACC in mixed-autonomy vehicle platoon. Em *2019 IEEE 58th Conference on Decision and Control (CDC)*, páginas 4079–4084, Nice, France, dezembro 2019. IEEE.

[45] Elena Cinque, Francesco Valentini, Alessio Iovine, e Marco Pratesi. An adaptive strategy to mitigate instability in the ETSI DCC: Experimental validation. Em *2017 15th International Conference on ITS Telecommunications (ITST)*, páginas 1–6, Warsaw, Poland, maio 2017. IEEE.

[46] CMU Portugal. FLOYD. https://www.cmuportugal.org/large-scale-collaborative-research-projects/floyd/.

[47] Cohda Wireless. Cohda Stack, 2019. https://cohdawireless.com/solutions/v2x-stack/.

[48] Cohda Wireless. Mk5 OBU, 2019. https://cohdawireless.com/solutions/hardware/mk5-obu/.

[49] CAR 2 CAR Communication Consortium. Survey on ITS-G5 CAM statistics. Released for public sharing C2CCC-TR-2052, CAR 2 CAR Communication Consortium, 2018.

[50] Coppelia Robotics. Coppelia Robotics, dezembro 2021. https://www.coppeliarobotics.com/.

[51] A. Cruz-Martín, J.A. Fernández-Madrigal, C. Galindo, J. González-Jiménez, C. Stockmans-Daou, e J.L. Blanco-Claraco. A LEGO Mindstorms NXT approach for teaching at Data Acquisition, Control Systems Engineering and Real-Time Systems undergraduate courses. *Computers & Education*, 59(3):974–988, novembro 2012.

[52] Cyberbotics Ltd. Webots documentation: Webots User Guide, 2020. https://cyberbotics.com/doc/guide/index.

[53] Amirhossein Hosseinzadeh Dadash. *A Cyber-Physical Testbed for Wireless Networked Control Systems*. Master degree, Universit of Gavle, FACULTY OF ENGINEERING AND SUSTAINABLE DEVELOPMENT, janeiro 2020.

[54] Soodeh Dadras, Sara Dadras, e Chris Winstead. Collaborative Attacks on Autonomous Vehicle Platooning. Em *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, páginas 464–467, Windsor, Ontario, Canada, agosto 2018. IEEE.

[55] Defense Advanced Research Projects Agency. Request for Comments 791, julho 1981. https://www.rfc-editor.org/rfc/rfc791.txt.

[56] Defense Advanced Research Projects Agency. Request for Comments 793, setembro 1981. https://rfc-editor.org/rfc/rfc793.txt.

[57] Mario di Bernardo, Alessandro Salvi, e Stefania Santini. Distributed Consensus Strategy for Platooning of Vehicles in the Presence of Time-Varying Heterogeneous Communication Delays. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):102–112, fevereiro 2015.

[58] J. Ding, H. Pei, J. Hu, e Y. Zhang. Cooperative Adaptive Cruise Control in Vehicle Platoon under Environment of i-VICS. Em *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, páginas 1246–1251, Maui, Hawaii, USA, novembro 2018. IEEE.

[59] V. S. Dolk, J. Ploeg, e W. P. M. H. Heemels. Event-Triggered Control for String-Stable Vehicle Platooning. *IEEE Transactions on Intelligent Transportation Systems*, 18(12):3486–3500, dezembro 2017.

[60] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, e Vladlen Koltun. CARLA: An Open Urban Driving Simulator. *arXiv:1711.03938 [cs]*, novembro 2017. arXiv: 1711.03938.

[61] Falko Dressler, Florian Klingler, Michele Segata, e Renato Lo Cigno. Cooperative Driving and the Tactile Internet. *Proceedings of the IEEE*, 107(2):436–446, fevereiro 2019.

[62] A.V. Durga Ganesh Reddy e Barathram Ramkumar. Simulation studies on ZigBee network for in-vehicle wireless communications. Em *2014 International Conference on Computer Communication and Informatics*, páginas 1–6, Coimbatore, India, janeiro 2014. IEEE.

[63] E. Ahmed e H. Gharavi. Cooperative Vehicular Networking: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):996–1014, março 2018.

[64] E-Maxxdude. Ford Fiesta® ST Rally: 1/10 Scale Electric Rally Racer with TQ 2.4GHz radio system - https://traxxas.com/products/models/electric/ford-fiesta-st-rally, agosto 2017. Library Catalog: traxxas.com.

[65] D. Eckhoff, N. Sofra, e R. German. A performance study of cooperative awareness in ETSI ITS G5 and IEEE WAVE. Em *2013 10th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, páginas 196–200, Banff, AB, Canada, março 2013. IEEE.

[66] ECSSEL JU. SafeCOP project Overview - http://www.safecop.eu/?page/_id=18, maio 2016. Library Catalog: www.safecop.eu.

[67] S. Eichler. Performance Evaluation of the IEEE 802.11p WAVE Communication Standard. Em *2007 IEEE 66th Vehicular Technology Conference*, páginas 2199–2203, Baltimore, MD, USA, setembro 2007. IEEE.

[68] Basem M. ElHalawany, Ahmad A. Aziz El-Banna, e Kaishun Wu. Physical-Layer Security and Privacy for Vehicle-to-Everything. *IEEE Communications Magazine*, 57(10):84–90, outubro 2019.

[69] Erik Dahlman, Stefan Parkvall, e Johan Skold. *4G: LTE/LTE-Advanced for Mobile Broadband*. Elsevier, Academic Press, 2 edição, 2014.

[70] Erik Larsson, Gustav Sennton, e Jeffrey Larson. The vehicle platooning problem: Computational complexity and heuristics. *Transportation Research Part C: Emerging Technologies*, 60:258–277, novembro 2015.

[71] ETSI. Centre for Testing & Interoperability, dezembro 2020. https://portal.etsi.org/Services/Centre-for-Testing-Interoperability.

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

[73] European Telecommunications Standards Institute. ETSI EN 302 665 V1.1.1 Intelligent Transport Systems (ITS); Communications Architecture. Relatório técnico V1.1.1, European Telecommunications Standards Institute, julho 2010.

[74] European Telecommunications Standards Institute. ETSI TS 102 637-1 V1.1.1 Intelligent Transport Systems (ITS);Vehicular Communications; Basic Set of Applications; Part 1: Functional Requirements. Relatório técnico, European Telecommunications Standards Institute, setembro 2010.

[75] European Telecommunications Standards Institute. ETSI TS 102 687 V1.1.1 Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range; Access layer part. Relatório técnico, European Telecommunications Standards Institute, julho 2011.

[76] European Telecommunications Standards Institute. ETSI EN 302 663 V1.2.0 Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band. Relatório técnico, European Telecommunications Standards Institute, novembro 2012.

[77] European Telecommunications Standards Institute. ETSI EN 302 636-3 V1.1.2 Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 3: Network Architecture. Relatório técnico, ETSI, março 2014.

[78] European Telecommunications Standards Institute. ETSI EN 302 637-3 V1.2.1 Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service. Relatório técnico, European Telecommunications Standards Institute, setembro 2014.

[79] European Telecommunications Standards Institute. ETSI EN 302 636-5-1 V2.1.0 Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 5: Transport Protocols; Sub-part 1: Basic Transport Protocol. Relatório técnico, ETSI, maio 2017.

[80] European Telecommunications Standards Institute. ETSI TS 103 097 V1.3.1 - Intelligent Transport Systems (ITS);Security; Security header and certificate formats. Technical Specification 103 097, European Telecommunications Standards Institute, outubro 2017.

[81] European Telecommunications Standards Institute. ETSI TS 133 185 V14.1.0. - 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects;Security aspect for LTE support of Vehicle-to-Everything (V2X) services - (Release 14). Relatório técnico, European Telecommunications Standards Institute, outubro 2017.

[82] European Telecommunications Standards Institute. ETSI EN 302 637-2 V1.4.0 Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service. Relatório técnico V1.4.0, ETSI, agosto 2018.

[83] European Telecommunications Standards Institute. ETSI TS 102 940 V1.3.1 - Intelligent Transport Systems (ITS);Security; ITS communications security architecture and security management. Technical Specification TS 102 940, European Telecommunications Standards Institute, abril 2018.

[84] European Telecommunications Standards Institute. ETSI TR 103 299 V2.1.1 Intelligent Transport Systems (ITS); Cooperative Adaptive Cruise Control (CACC); Pre-standardization study. Relatório técnico, European Telecommunications Standards Institute, junho 2019.

[85] Faten Fakhfakh, Mohamed Tounsi, e Mohamed Mosbah. Vehicle Platooning Systems: Review, Classification and Validation Strategies. *International Journal of Networked and Distributed Computing*, 8(4):11, 2020.

[86] A. Farag, A. Hussein, O. M. Shehata, e E. I. Morgan. Design and Validation of A Novel Adaptive Cruise Control Law for a Platoon Of Vehicles. Em *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, páginas 81–86, Giza, Egypt, outubro 2020. IEEE.

[87] Federal Higway Administration. Cooperative Driving Automation | FHWA - CARMA project, 2020. https://highways.dot.gov/research/operations/Cooperative-Driving-Automation.

[88] Feedback PLC. Welcome to Feedback plc, maio 2020. Library Catalog: fbkmed.com.

[89] A Feher, L. Nagy, e L. Marton. Testbed for Mobile Robot Platoon Control System Evaluation. Em *2019 IEEE 19th International Symposium on Computational Intelligence and Informatics and 7th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Sciences and Robotics (CINTI-MACRo)*, páginas 000137–000142, Szeged, Hungary, novembro 2019. IEEE.

[90] H. Feng, C. Wong, C. Liu, e S. Xiao. ROS-Based Humanoid Robot Pose Control System Design. Em *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, páginas 4089–4093, Miyazaki, Japan, outubro 2018.

[91] S. Feng, H. Sun, Y. Zhang, J. Zheng, H. X. Liu, e L. Li. Tube-Based Discrete Controller Design for Vehicle Platoons Subject to Disturbances and Saturation Constraints. *IEEE Transactions on Control Systems Technology*, 28(3):1066–1073, maio 2020.

[92] Alessandro Fermi, Maurizio Mongelli, Marco Muselli, e Enrico Ferrari. Identification of safety regions in vehicle platooning via machine learning. Em *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, páginas 1–4, Imperia, Italy, junho 2018. IEEE.

[93] Bruno Fernandes. Implementation and Analysis of IEEE and ETSI Security Standards for Vehicular Communications. *Mobile Netw Appl*, 23(23):469–478, 2018.

[94] Pedro Fernandes e Urbano Nunes. Platooning With IVC-Enabled Autonomous Vehicles: Strategies to Mitigate Communication Delays, Improve Safety and Traffic Flow. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):91–106, março 2012.

[95] A. Festag. Cooperative intelligent transport systems standards in europe. *IEEE Communications Magazine*, 52(12):166–172, dezembro 2014. Conference Name: IEEE Communications Magazine.

[96] Andreas Festag. Standards for vehicular communication—from IEEE 802.11p to 5G. *Elektrotechnik und Informationstechnik*, 132(7):409–416, novembro 2015.

[97] Enio Vasconcelos Filho, Nuno Guedes, Bruno Vieira, Miguel Mestre, Ricardo Severino, Bruno Gonçalves, Anis Koubaa, e Eduardo Tovar. Towards a Cooperative Robotic Platooning Testbed. Em *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), 2020*, páginas 332–337, Ponta Delgada, Portugal, abril 2020. IEEE.

[98] Enio Vasconcelos Filho, Pedro M. Santos, Ricardo Severino, Anis Koubaa, e Eduardo Tovar. Improving the Performance of Cooperative Platooning With Restricted Message Trigger Thresholds. *IEEE Access*, 10:45562–45575, 2022.

[99] Enio Vasconcelos Filho, Ricardo Severino, Anis Koubaa, e Eduardo Tovar. An Integrated Lateral and Longitudinal Look Ahead Controller for Cooperative Vehicular Platooning. Em *4th EAI International Conference on Intelligent Transport Systems (EAI INTSYS 2020).*, página 18, Online, dezembro 2020. EAI.

[100] Enio Vasconcelos Filho, Ricardo Severino, Anis Koubaa, e Eduardo Tovar. A Wireless Safety and Security Layer Architecture for Reliable Co-CPS. Em *DCE21- Symposium on Electrical and Computer Engineering: Book of Abstracts*, volume 1, página 3, Porto, Portugal, junho 2021. FEUP.

[101] Enio Vasconcelos Filho, Ricardo Severino, Joao Rodrigues, Bruno Gonçalves, Anis Koubaa, e Eduardo Tovar. CopaDrive: An Integrated ROS Cooperative Driving Test and Validation Framework. Em *Robot Operating System (ROS)*, volume 962, páginas 121–174. Springer International Publishing, Cham, 2021. Series Title: Studies in Computational Intelligence.

[102] Alessio Filippi, Kees Moerman, Gerardo Daalderop, Paul D Alexander, Franz Schober, e Werner Pfliegl. Ready to roll: Why 802.11p beats LTE and 5G for V2x. Relatório técnico, NXP Semiconductors, abril 2016.

[103] Alessio Filippi, Kees Moerman, Vincent Martinez, Andrew Turley, Onn Haran, e Ron Toledano. IEEE802.11p ahead of LTE-V2V for safety applications. *NXP Autotalks*, 1(1):19, 2017.

[104] Sydney J. Freedberg Jr. Army Secretary Rides Robot Truck: 'Critical' Tech For Big Six, janeiro 2018.

[105] Frost. Global Autonomous Driving Market Outlook, 2018, março 2018. https://info.microsoft.com/rs/157-GQE-382/images/K24A-2018/Frost/26/Sullivan/-Global/Autonomous/Driving/Outlook.pdf.

[106] Alexander Frötscher, Bernhard Monschiebl, Anastasios Drosou, Erol Gelenbe, Martin J. Reed, e Mays Al-Naday. Improve cybersecurity of C-ITS Road Side Infrastructure Installations: the SerIoT - Secure and Safe IoT approach. Em *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, páginas 1–5, Graz, Austria, novembro 2019. IEEE. ISSN: 2378-1297.

[107] Fadri Furrer, Michael Burri, Markus Achtelik, e Roland Siegwart. RotorS—A Modular Gazebo MAV Simulator Framework. Em Anis Koubaa, organizador, *Robot Operating System (ROS)*, volume 625, páginas 595–625. Springer International Publishing, Cham, 2016. Series Title: Studies in Computational Intelligence.

[108] Ahmad Adamu Galadima. Arduino as a learning tool. Em *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*, páginas 1–4, setembro 2014.

[109] W. Gao, Y. Shi, e S. Chen. Scalable Platooning Based on Directed Information Flow Topology With Granulating Method. *IEEE Access*, 7:176634–176645, 2019. Conference Name: IEEE Access.

[110] A. Ghasemi, R. Kazemi, e S. Azadi. Stable Decentralized Control of a Platoon of Vehicles With Heterogeneous Information Feedback. *IEEE Transactions on Vehicular Technology*, 62(9):4299–4308, novembro 2013.

[111] Giulia Giordano, Michele Segata, Franco Blanchini, e Renato Lo Cigno. The joint network/control design of platooning algorithms can enforce guaranteed safety constraints. *Ad Hoc Networks*, 94:101962, novembro 2019.

[112] Siyuan Gong e Lili Du. Cooperative platoon control for a mixed traffic flow including human drive vehicles and connected and autonomous vehicles. *Transportation Research Part B: Methodological*, 116:25–61, outubro 2018.

[113] Siyuan Gong, Anye Zhou, Jian Wang, Tao Li, e Srinivas Peeta. Cooperative Adaptive Cruise Control for a Platoon of Connected and Autonomous Vehicles Considering Dynamic Information Flow Topology. *21st IEEE International Conference on Intelligent Transportation Systems*, 1(1):6, 2018.

[114] Jose Gonçalves, Vitor H. Pinto, e Paulo Costa. A Line Follower Educational Mobile Robot Performance Robustness Increase Using a Competition as Benchmark. Em *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, páginas 934–939, abril 2019. ISSN: 2576-3555.

[115] Ministry of Transportation Government of Ontario. Cooperative Truck Platooning Conditions, junho 2021.

[116] Graphics Lab and Humanoid Robotics Lab. Dynamic Animation and Robotics Toolkit - DART, dezembro 2021.

[117] Radhika Grover, Shoba Krishnan, Terry Shoup, e Maryam Khanbaghi. A competition-based approach for undergraduate mechatronics education using the arduino platform. Em *Fourth Interdisciplinary Engineering Design Education Conference*, páginas 78–83, março 2014. ISSN: 2161-5330.

[118] Armin Gräter, Eckard Steiger, Manfred Harrer, e Mats Rosenquist. Connected, Cooperative and Automated Driving - Update of ERTRAC Roadmap. Relatório técnico 9.0, ERTRAC Working Group, julho 2021.

[119] Jacopo Guanetti, Yeojun Kim, e Francesco Borrelli. Control of Connected and Automated Vehicles: State of the Art and Future Challenges. *Annual Reviews in Control*, 45(1):18–40, abril 2018. arXiv: 1804.03757.

[120] K. Halder, U. Montanaro, S. Dixit, M. Dianati, A. Mouzakitis, e S. Fallah. Distributed H Controller Design and Robustness Analysis for Vehicle Platooning Under Random Packet Drop. *IEEE Transactions on Intelligent Transportation Systems*, 1(1):1–14, 2020. Conference Name: IEEE Transactions on Intelligent Transportation Systems.

[121] Randolph Hall e Chinan Chin. Vehicle sorting for platoon formation: Impacts on highway entry and throughput. *Transportation Research Part C: Emerging Technologies*, 13(5-6):405–420, outubro 2005.

[122] Kimberly A. Hambuchen, Monsi C. Roman, Amy Sivak, Angela Herblet, Nathan Koenig, Daniel Newmyer, e Robert Ambrose. NASA's Space Robotics Challenge: Advancing Robotics for Future Exploration Missions. Em *AIAA SPACE and Astronautics Forum and Exposition*, Orlando, FL, setembro 2017. American Institute of Aeronautics and Astronautics.

[123] He Hao e Prabir Barooah. Stability and robustness of large platoons of vehicles with double-integrator models and nearest neighbor interaction: STABILITY AND ROBUSTNESS OF LARGE PLATOONS OF VEHICLES. *International Journal of Robust and Nonlinear Control*, 23(18):2097–2122, dezembro 2013.

[124] Morteza Hashemi, Wei Si, Moshe Laifenfeld, David Starobinski, e Ari Trachtenberg. Intra-car multihop wireless sensor networking: a case study. *IEEE Communications Magazine*, 52(12):183–191, dezembro 2014.

[125] Defeng He, Tianxiang Qiu, e Renshi Luo. Fuel efficiency-oriented platooning control of connected nonlinear vehicles: A distributed economic MPC approach. *Asian Journal of Control*, 22(4):1628–1638, 2020.

[126] Larry Higgs. Another bus lane would ease traffic to NYC, but there's a high-tech reason it may not happen, junho 2019. Section: Traffic.

[127] Haimin Hu, Ye Pu, Mo Chen, e Claire J. Tomlin. Plug and Play Distributed Model Predictive Control for Heavy Duty Vehicle Platooning and Interaction with Passenger Vehicles. Em *2018 IEEE Conference on Decision and Control (CDC)*, páginas 2803–2809, FL, USA, dezembro 2018. IEEE.

[128] Rasheed Hussain e Sherali Zeadally. Autonomous Cars: Research Results, Issues and Future Challenges. *IEEE Communications Surveys and Tutorials*, 21(2):1275 – 1313, outubro 2018.

[129] Ahmed Hussein, Fernando García, e Cristina Olaverri-Monreal. ROS and Unity Based Framework for Intelligent Vehicles Control and Simulation. Em *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, páginas 1–6, setembro 2018.

[130] Ian Chen e Carlos Agüero. Vehicle and city simulation with Gazebo and ROS. Em *ROSCon Vancouver 2017*, Vancouver, Canada, setembro 2017. Open Robotics.

[131] IEEE. IEEE Std 802.11. Relatório técnico, IEEE, 2007. IEEE Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[132] IEEE. IEEE Guide for Wireless Access in Vehicular Environments (WAVE) Architecture - Redline. Relatório técnico 1609.0-2019, IEEE, abril 2019.

[133] Adafruit Industries. Teensy 3.2 + header - https://www.adafruit.com/product/2756. Library Catalog: www.adafruit.com.

[134] Insectt Project. InSecTT | InSecTT. https://www.insectt.eu/.

[135] InvenSense Inc. MPU 92-50, Product Specification - https://cdn.sparkfun.com/assets/learn/tutorials/5/5/0/MPU9250REV1.0.pdf, janeiro 2014.

[136] J. Postel. Request for Comments 768, agosto 1980. https://www.rfc-editor.org/rfc/rfc768.txt.

[137] Naveen Chakravarthy Jakka. *Emergency Braking in Platooning with Communication Loss*. Tese de doutoramento, Delft University of Technology, julho 2020.

[138] Andrew Jamison, Anette Kolmos, e Jette Egelund Holgaard. Hybrid Learning: An Integrative Approach to Engineering Education. *Journal of Engineering Education*, 103(2):253–273, abril 2014.

[139] Joao P. Jansch-Porto e Geir E Dullerud. Decentralized control with moving-horizon linear switched systems: Synthesis and testbed implementation. Em *2017 American Control Conference (ACC)*, páginas 851–856, Seattle, WA, USA, maio 2017. IEEE.

[140] JetsonHacks. NVIDIA Jetson TX2 J21 Header Pinout - https://www.jetsonhacks.com/nvidia-jetson-tx2-j21-header-pinout/, fevereiro 2020.

[141] Dongyao Jia, Kejie Lu, e Jianping Wang. A Disturbance-Adaptive Design for VANET-Enabled Vehicle Platoon. *IEEE Transactions on Vehicular Technology*, 63(2):527–539, fevereiro 2014.

[142] Dongyao Jia, Kejie Lu, Jianping Wang, Xiang Zhang, e Xuemin Shen. A Survey on Platoon-Based Vehicular Cyber-Physical Systems. *IEEE Communications Surveys & Tutorials*, 18(1):263–284, 2016.

[143] Dongyao Jia e Dong Ngoduy. Platoon based cooperative driving model with consideration of realistic inter-vehicle communication. *Transportation Research Part C: Emerging Technologies*, 68:245–264, julho 2016.

[144] Dongyao Jia, Jie Sun, Anshuman Sharma, Zuduo Zheng, e Bingyi Liu. Integrated simulation platform for conventional, connected and automated driving: A design from cyber-physical systems perspective. *Transportation Research Part C: Emerging Technologies*, Pre-print:21, 2020.

[145] Li Jin, Mladen Čičič, Saurabh Amin, e Karl H. Johansson. Modeling the Impact of Vehicle Platooning on Highway Congestion: A Fluid Queuing Approach. Em *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (part of CPS Week) - HSCC '18*, páginas 237–246, Porto, Portugal, 2018. ACM Press.

[146] Jing Zhou e Huei Peng. Range policy of adaptive cruise control vehicles for improved flow stability and string stability. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):229–237, junho 2005.

[147] Adit Joshi. A Novel Approach for Validating Adaptive Cruise Control (ACC) Using Two Hardware-in-the-Loop (HIL) Simulation Benches. SAE Technical Paper 2019-01-1038, SAE International, Warrendale, PA, abril 2019. ISSN: 0148-7191, 2688-3627.

[148] Mouna Karoui, Antonio Freitas, e Gerard Chalhoub. Performance comparison between LTE-V2X and ITS-G5 under realistic urban scenarios. Em *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, páginas 1–7, Antwerp, Belgium, maio 2020. IEEE.

[149] Oussama Karoui, Emna Guerfala, Anis Koubaa, Mohamed Khalgui, Eduardo Tovard, Naiqi Wu, Abdulrahman Al-Ahmari, e Zhiwu Li. Performance evaluation of vehicular platoons using Webots. *IET Intelligent Transport Systems*, 11(8):441–449, outubro 2017.

[150] Oussama Karoui, Mohamed Khalgui, Anis Koubâa, Emna Guerfala, Zhiwu Li, e Eduardo Tovar. Dual mode for vehicular platoon safety: Simulation and formal verification. *Information Sciences*, 402:216–232, setembro 2017.

[151] Florian Kauer, Maximilian Köstler, Tobias Lübkert, e Volker Turau. OpenDSME - A portable framework for reliable wireless sensor and actuator networks. Em *2017 International Conference on Networked Systems (NetSys)*, páginas 1–2, março 2017.

[152] Pooja Kavathekar e Yangquan Chen. Detc2011/Mesa-47861 Draft: Vehicle Platooning: A Brief Survey and Categorization. Relatório técnico, Proceedings of The ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Washington, DC, USA, 2011.

[153] E. Kayacan. Multiobjective H Control for String Stability of Cooperative Adaptive Cruise Control Systems. *IEEE Transactions on Intelligent Vehicles*, 2(1):52–61, março 2017.

[154] John B Kenney. Dedicated Short-Range Communications (DSRC) Standards in the United States. *Proceedings of the IEEE*, 99(7):22, julho 2011.

[155] M. E. Khatir e E. J. Davidson. Bounded stability and eventual string stability of a large platoon of vehicles using non-identical controllers. Em *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, volume 1, páginas 1111–1116 Vol.1, Nassau, Bahamas, dezembro 2004. IEEE.

[156] Roozbeh Kianfar, Mohammad Ali, Paolo Falcone, e Jonas Fredriksson. Combined longitudinal and lateral control design for string stable vehicle platooning within a designated lane. Em *IEEE Conference on Intelligent Transportation Systems (ITSC), 2014*, páginas 1003–1008, outubro 2014. ISSN: 2153-0017.

[157] Min Su KIm, Raimarius Delgado, e Byoung Wook Choi. Comparative Study of ROS on Embedded System for a Mobile Robot. *Journal of Automation, Mobile Robotics and Intelligent Systems*, 12(3):61–67, dezembro 2018.

[158] Kanako Kobatake, Tadatsugi Okazaki, e Masakazu Arima. Study on Optimal Tuning of PID Autopilot for Autonomous Surface Vehicle. *IFAC-PapersOnLine*, 52(21):335–340, 2019.

[159] Andras Kondakor, Zsombor Torcsvari, Akos Nagy, e Istvan Vajk. A Line Tracking Algorithm Based on Image Processing. Em *2018 IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, páginas 000039–000044, Timisoara, maio 2018. IEEE.

[160] S. Kuehlmorgen, P. Schmager, A. Festag, e G. Fettweis. Simulation-Based Evaluation of ETSI ITS-G5 and Cellular-VCS in a Real-World Road Traffic Scenario. Em *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, páginas 1–6, Chicago, USA, agosto 2018. IEEE. ISSN: 2577-2465.

[161] Sebastian Kuhlmorgen, Ignacio Llatser, Andreas Festag, e Gerhard Fettweis. Performance Evaluation of ETSI GeoNetworking for Vehicular Ad Hoc Networks. Em *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, páginas 1–6, Glasgow, United Kingdom, maio 2015. IEEE.

[162] Elis Kulla, Ningling Jiang, Evjola Spaho, e Noritaka Nishihara. A Survey on Platooning Techniques in VANETs. Em Leonard Barolli, Nadeem Javaid, Makoto Ikeda, e Makoto Takizawa, organizadores, *Complex, Intelligent, and Software Intensive Systems*, volume 772, páginas 650–659. Springer International Publishing, Cham, 2019. Series Title: Advances in Intelligent Systems and Computing.

[163] Kirsi Kuosa, Damiano Distante, Anne Tervakari, Luigi Cerulo, Alejandro Fernández, Juho Koro, e Meri Kailanto. Interactive Visualization Tools to Improve Learning and Teaching in Online Learning Environments:. *International Journal of Distance Education Technologies*, 14(1):1–21, janeiro 2016.

[164] Harrison Kurunathan, Ricardo Severino, Enio Filho, e Eduardo Tovar. WiCAR - Simulating Towards the Wireless Car. Em *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops*, volume 12235, páginas 136–147. Springer International Publishing, Cham, 2020. Series Title: Lecture Notes in Computer Science.

[165] Harrison Kurunathan, Ricardo Severino, Anis Koubaa, e Eduardo Tovar. Worst-case bound analysis for the time-critical MAC behaviors of IEEE 802.15.4e. Em *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, páginas 1–9, maio 2017.

[166] Harrison Kurunathan, Ricardo Severino, Anis Koubaa, e Eduardo Tovar. IEEE 802.15.4e in a Nutshell: Survey and Performance Evaluation. *IEEE Communications Surveys Tutorials*, 20(3):1989–2010, 2018. Conference Name: IEEE Communications Surveys Tutorials.

[167] Harrison Kurunathan, Ricardo Severino, Anis Koubaa, e Eduardo Tovar. Symphony: routing aware scheduling for DSME networks. *ACM SIGBED Review*, 16(4):26–31, janeiro 2020.

[168] Seolyoung Lee, Cheol Oh, e Gunwoo Lee. Impact of Automated Truck Platooning on the Performance of Freeway Mixed Traffic Flow. *Journal of Advanced Transportation*, 2021:1–13, janeiro 2021.

[169] Sun Woo Lee, Sok Joon Lee, e Dong Hoon Lee. Attack on Vehicular Platooning and Mitigation Strategy: A Survey. *Applied Mechanics and Materials*, 865:423–428, junho 2017.

[170] C. Lei, E. M. van Eenennaam, W. K. Wolterink, G. Karagiannis, G. Heijenk, e J. Ploeg. Impact of packet loss on CACC string stability performance. Em *2011 11th International Conference on ITS Telecommunications*, páginas 381–386, St. Petersburg, Russia, agosto 2011. IEEE.

[171] Lei Lei, Tong Liu, Kan Zheng, e Lajos Hanzo. Deep Reinforcement Learning Aided Platoon Control Relying on V2X Information. *IEEE Transactions on Vehicular Technology*, 71(6):5811–5826, junho 2022.

[172] Matt Leonard, Joseph Morgan, e Jeremy P Coffelt. Digital Systems Teaching and Research (DSTR) Robot: A Flexible Platform for Education and Applied Research. Em *Proceedings of the 2018 ASEE Gulf-Southwest Section Annual Conference*, página 5, The University of Texas at Austin, 2018. ASEE.

[173] Kai Li, Wei Ni, Yousef Emami, Yiran Shen, Ricardo Severino, David Pereira, e Eduardo Tovar. Design and Implementation of Secret Key Agreement for Platoon-based Vehicular Cyber-physical Systems. *ACM Transactions on Cyber-Physical Systems*, 4(2):1–20, fevereiro 2020.

[174] Meng Li, Zehong Cao, e Zhibin Li. A Reinforcement Learning-Based Vehicle Platoon Control Strategy for Reducing Energy Consumption in Traffic Oscillations. *IEEE Transactions on Neural Networks and Learning Systems*, 32(12):5309–5322, dezembro 2021.

[175] S. E. Li, Y. Zheng, K. Li, Y. Wu, J. K. Hedrick, F. Gao, e H. Zhang. Dynamical Modeling and Distributed Control of Connected and Automated Vehicles: Challenges and Opportunities. *IEEE Intelligent Transportation Systems Magazine*, 9(3):46–58, 2017. Conference Name: IEEE Intelligent Transportation Systems Magazine.

[176] Y. Li, W. Chen, S. Peeta, e Y. Wang. Platoon Control of Connected Multi-Vehicle Systems Under V2X Communications: Design and Experiments. *IEEE Transactions on Intelligent Transportation Systems*, pre-print(1):1–12, 2019.

[177] Yongfu Li, Kezhi Li, Taixiong Zheng, Xiangdong Hu, Huizong Feng, e Yinguo Li. Evaluating the performance of vehicular platoon control under different network topologies of initial states. *Physica A: Statistical Mechanics and its Applications*, 450:359–368, maio 2016.

[178] Z. Li, B. Hu, M. Li, e G. Luo. String Stability Analysis for Vehicle Platooning Under Unreliable Communication Links With Event-Triggered Strategy. *IEEE Transactions on Vehicular Technology*, 68(3):2152–2164, março 2019. Conference Name: IEEE Transactions on Vehicular Technology.

[179] Zhiwu Li, Oussama Karoui, Anis Koubâa, Mohamed Khalgui, Emna Guerfala, Eduardo Tovar, e Naiqi Wu. *System and method for operating a follower vehicle in a vehicle platoon*. Macau University of Science and Technology, 2018.

[180] Jiun-Ren Lin, Timothy Talty, e Ozan K. Tonguz. Feasibility of Safety Applications Based on Intra-Car Wireless Sensor Networks: A Case Study. Em *2011 IEEE Vehicular Technology Conference (VTC Fall)*, páginas 1–5, San Francisco, CA, USA, setembro 2011. IEEE.

[181] Steffen Linsenmayer, Dimos V. Dimarogonas, e Frank Allgöwer. Event-Based Vehicle Coordination Using Nonlinear Unidirectional Controllers. *IEEE Transactions on Control of Network Systems*, 5(4):1575–1584, dezembro 2018. Conference Name: IEEE Transactions on Control of Network Systems.

[182] J. Liu, N. Kato, J. Ma, e N. Kadowaki. Device-to-Device Communication in LTE-Advanced Networks: A Survey. *IEEE Communications Surveys Tutorials*, 17(4):1923–1940, 2015. Conference Name: IEEE Communications Surveys Tutorials.

[183] Tong Liu, Lei Lei, Kan Zheng, e Kuan Zhang. Autonomous Platoon Control With Integrated Deep Reinforcement Learning and Dynamic Programming. *IEEE Internet of Things Journal*, 10(6):5476–5489, março 2023.

[184] I. Llatser, G. Jornod, A. Festag, D. Mansolino, I. Navarro, e A. Martinoli. Simulation of cooperative automated driving by bidirectional coupling of vehicle and network simulators. Em *2017 IEEE Intelligent Vehicles Symposium (IV)*, páginas 1881–1886, Los Angeles, CA, USA, junho 2017. IEEE.

[185] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, e E. WieBner. Microscopic Traffic Simulation using SUMO. Em *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, páginas 2575–2582, Maui, HI, USA, novembro 2018. IEEE.

[186] Duo Lu, Zhichao Li, e Dijiang Huang. Platooning as a service of autonomous vehicles. Em *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, páginas 1–6, Macau, China, junho 2017. IEEE.

[187] Duo Lu, Zhichao Li, Dijiang Huang, Xianglong Lu, Yuli Deng, Ankur Chowdhary, e Bing Li. VC-bots: a vehicular cloud computing testbed with mobile robots. Em *Proceedings of the First International Workshop on Internet of Vehicles and Vehicles of Internet - IoV-VoI '16*, páginas 31–36, Paderborn, Germany, 2016. ACM Press.

[188] Z. Lu, G. Qu, e Z. Liu. A Survey on Recent Advances in Vehicular Network Security, Trust, and Privacy. *IEEE Transactions on Intelligent Transportation Systems*, 20(2):760–776, fevereiro 2019.

[189] Lucas Nulle. Lucas Nülle - Lucas-Nuelle Training Systems for vocational training and didactic, maio 2020.

[190] Fengqiao Luo, Jeffrey Larson, e Todd Munson. Coordinated platooning with multiple speeds. *Transportation Research Part C: Emerging Technologies*, 90:213–225, maio 2018.

[191] D. L. Luu, C. Lupu, L. S. Ismail, e H. Alshareefi. Spacing Control of Cooperative Adaptive Cruise Control Vehicle Platoon. Em *2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, páginas 1–6, Cluj-Napoca, Romania, maio 2020. IEEE.

[192] N. Lyamin, A. Vinel, M. Jonsson, e B. Bellalta. Cooperative Awareness in VANETs: On ETSI EN 302 637-2 Performance. *IEEE Transactions on Vehicular Technology*, 67(1):17–28, janeiro 2018.

[193] Nikita Lyamin, Qichen Deng, e Alexey Vinel. Study of the platooning fuel efficiency under ETSI ITS-G5 communications. Em *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, páginas 551–556, novembro 2016. ISSN: 2153-0017.

[194] Nikita Lyamin, Alexey Vinel, Dieter Smely, e Boris Bellalta. ETSI DCC: Decentralized Congestion Control in C-ITS. *IEEE Communications Magazine*, 56(12):112–118, dezembro 2018.

[195] Maria E. López-Lambas. The Socioeconomic Impact of the Intelligent Vehicles. Em *Intelligent Vehicles*, volume 1 de *Enabling Technologies and Future Developments*, páginas 437–453. Elsevier, Elsevier, 2018.

[196] F. Ma, J. Wang, S. Zhu, S. Y. Gelbal, Y. Yang, B. Aksun-Guvenc, e L. Guvenc. Distributed Control of Cooperative Vehicular Platoon With Nonideal Communication Condition. *IEEE Transactions on Vehicular Technology*, 69(8):8207–8220, agosto 2020. Conference Name: IEEE Transactions on Vehicular Technology.

[197] J. Ma, F. Zhou, Z. Huang, e R. James. Hardware-In-The-Loop Testing of Connected and Automated Vehicle Applications: A Use Case For Cooperative Adaptive Cruise Control. Em *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, páginas 2878–2883, Maui, Hawaii, USA, novembro 2018. IEEE. ISSN: 2153-0017.

[198] Anthony A. Maciejewski, Thomas W. Chen, Zinta S. Byrne, Michael A. De Miranda, Laura B. Sample Mcmeeking, Branislav M. Notaros, Ali Pezeshki, Sourajeet Roy, Andrea M. Leland, Melissa D. Reese, Alma H. Rosales, Thomas J. Siller, Richard F. Toftness, e Olivera Notaros. A Holistic Approach to Transforming Undergraduate Electrical Engineering Education. *IEEE Access*, 5:8148–8161, 2017.

[199] Abubakar U. Makarfi, Khaled M. Rabie, Omprakash Kaiwartya, Xingwang Li, e Rupak Kharel. Physical Layer Security in Vehicular Networks with Reconfigurable Intelligent Surfaces. Em *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, páginas 1–6, Online, maio 2020. IEEE. ISSN: 2577-2465.

[200] V. Mannoni, V. Berg, S. Sesia, e E. Perraud. A Comparison of the V2X Communication Systems: ITS-G5 and C-V2X. Em *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, páginas 1–5, Kuala Lumpur, Malaysia, abril 2019. IEEE. ISSN: 2577-2465.

[201] Ann F. McKenna. Educating Engineers: Designing for the Future of the Field. *The Journal of Higher Education*, 81(6):717–719, novembro 2010.

[202] Samer Medawar, Detlef Scholle, e Irfan Sljivo. Cooperative safety critical CPS platooning in SafeCOP. Em *2017 6th Mediterranean Conference on Embedded Computing (MECO)*, páginas 1–5, Bar, Montenegro, junho 2017. IEEE.

[203] J. Mei, K. Zheng, L. Zhao, L. Lei, e X. Wang. Joint Radio Resource Allocation and Control for Vehicle Platooning in LTE-V2V Network. *IEEE Transactions on Vehicular Technology*, 67(12):12218–12230, dezembro 2018. Conference Name: IEEE Transactions on Vehicular Technology.

[204] Karl Meinke. Learning-Based Testing of Cyber-Physical Systems-of-Systems: A Platooning Study. Em Philipp Reinecke e Antinisca Di Marco, organizadores, *Computer Performance Engineering*, volume 10497, páginas 135–151. Springer International Publishing, Cham, 2017. Series Title: Lecture Notes in Computer Science.

[205] Johannes Meyer, Alexander Sendobry, Stefan Kohlbrecher, Uwe Klingauf, e Oskar von Stryk. Comprehensive Simulation of Quadrotor UAVs Using ROS and Gazebo. Em *Simulation, Modeling, and Programming for Autonomous Robots*, volume 7628, páginas 400–411. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[206] Olivier Michel. Webots: Professional Mobile Robot Simulation. *International Journal of Advanced Robotic Systems*, 1(1):5, março 2004.

[207] Richard H. Middleton e Julio H. Braslavsky. String Instability in Classes of Linear Time Invariant Formation Control With Limited Communication Range. *IEEE Transactions on Automatic Control*, 55(7):1519–1530, julho 2010.

[208] Eman Mousavinejad, Fuwen Yang, Qing-Long Han, Xiaohua Ge, e Ljubo Vlacic. Distributed Cyber Attacks Detection and Recovery Mechanism for Vehicle Platooning. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3821–3834, setembro 2020.

[209] NIST. Agile Robotics for Industrial Automation Competition - https://www.nist.gov/el/intelligent-systems-division-73500/agile-robotics-industrial-automation-competition, julho 2020. Last Modified: 2020-07-09T09:19-04:00.

[210] nsnam. ns-3 - https://www.nsnam.org/about/, 2020.

[211] C. Obermaier, R. Riebl, e C. Facchi. Fully Reactive Hardware-in-the-Loop Simulation for VANET Devices. Em *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, páginas 3755–3760, Maui, Hawaii, USA, novembro 2018. IEEE.

[212] Matthew O'Kelly, Varundev Sukhil, Houssam Abbas, Jack Harkins, Chris Kao, Yash Vardhan Pant, Rahul Mangharam, Dipshil Agarwal, Madhur Behl, Paolo Burgio, e Marko Bertogna. F1/10: An Open-Source Autonomous Cyber-Physical Platform. *arXiv:1901.08567 [cs]*, janeiro 2019. arXiv: 1901.08567.

[213] Manabu Omae, Ryoko Fukuda, Takeki Ogitsu, e Wen-Po Chiang. Spacing Control of Cooperative Adaptive Cruise Control for Heavy-Duty Vehicles. *IFAC Proceedings Volumes*, 46(21):58–65, 2013.

[214] Hanafy M. Omar. Enhancing automatic control learning through Arduino-based projects. *European Journal of Engineering Education*, 43(5):652–663, setembro 2018.

[215] Omnet. What is OMNeT++? - https://omnetpp.org/intro/, 2020.

[216] Open Source Robotic Foundation. ROS/Introduction, julho 2018.

[217] Open Source Robotics Foundation. Prius Challenge, 2017.

[218] Open Source Robotics Foundation. Gazebo: Root Simularion Made Easy - http://gazebosim.org/, 2020.

[219] OpenCV. Hough Line Transform, 2019.

[220] OpenCV. Canny Edge Detection, dezembro 2022.

[221] Seungwoo Park, Jaehwan Lee, Ka Lok Man, e Sangoh Park. A SURVEY OF V2X COMMUNICATION TECHNIQUE FOR SUPPORTING PLATOONING. *ICIC Express Letters*, 14(5):6, maio 2020.

[222] Alberto Petrillo, Antonio Pescapé, e Stefania Santini. A collaborative approach for improving the security of vehicular scenarios: The case of platooning. *Computer Communications*, 122:59–75, junho 2018.

[223] C. Pilz, G. Steinbauer, M. Schratter, e D. Watzenig. Development of a Scenario Simulation Platform to Support Autonomous Driving Verification. Em *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, páginas 1–7, Graz, Austria, novembro 2019. IEEE. ISSN: 2378-1297.

[224] Diego G.S. Pivoto, Luiz F.F. de Almeida, Rodrigo da Rosa Righi, Joel J.P.C. Rodrigues, Alexandre Baratella Lugli, e Antonio M. Alberti. Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: A literature review. *Journal of Manufacturing Systems*, 58:176–192, janeiro 2021.

[225] J. Ploeg, D. P. Shukla, N. van de Wouw, e H. Nijmeijer. Controller Synthesis for String Stability of Vehicle Platoons. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):854–865, abril 2014.

[226] D. Plöger, L. Krüger, e A. Timm-Giel. Analysis of Communication Demands of Networked Control Systems for Autonomous Platooning. Em *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, páginas 14–19, Chania, Greece, junho 2018. IEEE.

[227] Paul Pop, Detlef Scholle, Irfan Šljivo, Hans Hansson, Gunnar Widforss, e Malin Rosqvist. Safe cooperating cyber-physical systems using wireless communication: The SafeCOP approach. *Microprocessors and Microsystems*, 53:42–50, agosto 2017.

[228] PyBullet. Bullet Real-Time Physics Simulation, dezembro 2021.

[229] Luisa M. Regueras, Elena Verdu, MarÍa F. Munoz, MarÍa A. Perez, Juan P. de Castro, e MarÍa JesÚs Verdu. Effects of Competitive E-Learning Tools on Higher Education Students: A Case Study. *IEEE Transactions on Education*, 52(2):279–285, maio 2009. Conference Name: IEEE Transactions on Education.

[230] Bruno Ribeiro, Fábio Gonçalves, Alexandre Santos, Maria João Nicolau, Bruno Dias, Joaquim Macedo, e António Costa. Simulation and Testing of a Platooning Management Protocol Implementation. Em *Wired/Wireless Internet Communications*, volume 10372, páginas 174–185. Springer International Publishing, Cham, 2017. Series Title: Lecture Notes in Computer Science.

[231] Richard van Nee e Ramjee Prasad. *OFDM for Wireless Multimedia Communications*. Artech House, Inc. Norwood, MA, USA, USA, 1st edição, 2000.

[232] R. Riebl, H. Günther, C. Facchi, e L. Wolf. Artery: Extending Veins for VANET applications. Em *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, páginas 450–456, Budapest, Hungary, junho 2015. IEEE.

[233] Raphael Riebl, Giovanni Nardini, e Antonio Virdis. Simulating LTE-Enabled Vehicular Communications. Em *Recent Advances in Network Simulation*, páginas 407–423. Springer International Publishing, Cham, 2019.

[234] Raphael Riebl, Christina Obermaier, e Hendrik-Jörn Günther. Artery: Large Scale Simulation Environment for ITS Applications. Em *Recent Advances in Network Simulation*, EAI/Springer Innovations in Communication and Computing, páginas 365–406. Springer International Publishing, Cham, 2019.

[235] Raphael Riebl, Christina Obermaier, Stefan Neumeier, e Christian Facchi. Vanetza: Boosting Research on Inter-Vehicle Communication. Em *Fachgesprach Inter-Vehicle Communication*, páginas 37–40, Fachgespräch, Germany, 2017. IEEE.

[236] Zandra B. Rivera, Marco C. De Simone, e Domenico Guida. Unmanned Ground Vehicle Modelling in Gazebo/ROS-Based Environments. *Machines*, 7(2):42, junho 2019.

[237] Carlo Rizzardo, Sunny Katyara, Miguel Fernandes, e Fei Chen. The Importance and the Limitations of Sim2Real for Robotic Manipulation in Precision Agriculture. *arXiv:2008.03983 [cs]*, agosto 2020. arXiv: 2008.03983.

[238] Robot Eletronics. SRF08 Ultra sonic range finder - https://www.robot-electronics.co.uk/htm/srf08tech.html, 2020.

[239] Michele Rondinone, Julen Maneros, Daniel Krajzewicz, Ramon Bauza, Pasquale Cataldi, Fatma Hrizi, Javier Gozalvez, Vineet Kumar, Matthias Röckl, Lan Lin, Oscar Lazaro, Jérémie Leguay, Jérôme Härri, Sendoa Vaz, Yoann Lopez, Miguel Sepulcre, Michelle Wetterwald, Robbin Blokpoel, e Fabio Cartolano. iTETRIS: A modular simulation platform for the large scale evaluation of cooperative ITS applications. *Simulation Modelling Practice and Theory*, 34:99–125, maio 2013.

[240] Astrid Rupp, Markus Tranninger, Raffael Wallner, Jasmina Zubača, Martin Steinberger, e Martin Horn. Fast and Low-Cost Testing of Advanced Driver Assistance Systems using Small-Scale Vehicles. *IFAC-PapersOnLine*, 52(5):34–39, 2019.

[241] S. Deering e R. Hinden. Request for Comments: 2460, dezembro 1998. https://www.rfc-editor.org/rfc/rfc2460.txt.

[242] SafeCOP. Safe Cooperating Cyber-Physical Systemsusing Wireless Communications, maio 2019.

[243] Björn Schünemann. V2X simulation runtime infrastructure VSimRTI: An assessment tool to design smart traffic management systems. *Computer Networks*, 55(14):3189–3198, outubro 2011.

[244] M. Segata, B. Bloessl, S. Joerer, C. Sommer, M. Gerla, R. Lo Cigno, e F. Dressler. Toward Communication Strategies for Platooning: Simulative and Experimental Evaluation. *IEEE Transactions on Vehicular Technology*, 64(12):5411–5423, dezembro 2015.

[245] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, e R. L. Cigno. Plexe: A platooning extension for Veins. Em *2014 IEEE Vehicular Networking Conference (VNC)*, páginas 53–60, Paderborn, Germany, dezembro 2014. IEEE.

[246] P. Seiler, A. Pant, e K. Hedrick. Disturbance Propagation in Vehicle Strings. *IEEE Transactions on Automatic Control*, 49(10):1835–1841, outubro 2004.

[247] Edgar Serrano Pérez e Fernando Juárez López. An ultra-low cost line follower robot as educational tool for teaching programming and circuit's foundations. *Computer Applications in Engineering Education*, 27(2):288–302, março 2019.

[248] Yunli Shao, Mohd Azrin Mohd Zulkefli, Zongxuan Sun, e Peter Huang. Evaluating connected and autonomous vehicles using a hardware-in-the-loop testbed and a living lab. *Transportation Research Part C: Emerging Technologies*, 102:121–135, maio 2019.

[249] Michael Sherman e Peter Eastman. SimTK - Simbody: Multibody Physics API, dezembro 2021.

[250] Steven Shooter e Mark Mcneill. Interdisciplinary Collaborative Learning in Mechatronics at Bucknell University. *Journal of Engineering Education*, 91(3):339–344, julho 2002.

[251] Fulvio Simonelli, Gennaro Nicola Bifulco, Valerio De Martinis, e Vincenzo Punzo. Human-Like Adaptive Cruise Control Systems through a Learning Machine Approach. Em *Applications of Soft Computing*, páginas 240–249. Springer Berlin Heidelberg, Berlin, Germany, 2009.

[252] Pranav Kumar Singh, Geetesh Saikamal Tabjul, Mohd Imran, Sunit Kumar Nandi, e Sukumar Nandi. Impact of Security Attacks on Cooperative Driving Use Case: CACC Platooning. Em *TENCON 2018 - 2018 IEEE Region 10 Conference*, páginas 0138–0143, Jeju Island, Korea, outubro 2018. IEEE. ISSN: 2159-3450.

[253] K. Sjoberg, P. Andres, T. Buburuzan, e A. Brakemeier. Cooperative Intelligent Transport Systems in Europe: Current Deployment Status and Outlook. *IEEE Vehicular Technology Magazine*, 12(2):89–97, junho 2017. Conference Name: IEEE Vehicular Technology Magazine.

[254] Rhona Smith. Directive 2010/41/EU of the European Parliament and of the Council of 7 July 2010. Em *Core EU Legislation*, páginas 352–355. Macmillan Education UK, London, 2015.

[255] Russ Smith. ODE - Open Dynamics Engine, dezembro 2021.

[256] Society of Motor Manufacturers & Traders. Truck Platooning: the future of road transport, junho 2020.

[257] Christoph Sommer. Veins, the open source vehicular network simulation framework., 2019.

[258] Aakash Soni e Huosheng Hu. Formation Control for a Fleet of Autonomous Ground Vehicles: A Survey. *Robotics*, 7(4):67, novembro 2018.

[259] Markus Sontowski, Stefan Kopsell, Thorsten Strufe, Christian Zimmermann, Andreas Weinand, Hans D. Schotten, e Norbert Bissmeyer. Towards Secure Communication for High-Density Longitudinal Platooning. Em *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, páginas 1–7, Honolulu, Hawaii, USA, setembro 2019. IEEE.

[260] Sparkfun. Infrared Proximity Sensor Long Range - Sharp GP2Y0A02YK0F - SEN-08958 - SparkFun Electronics, 2020.

[261] Sparkfun. SparkFun 9DoF Razor IMU M0 - SEN-14001 - SparkFun Electronics - https://www.sparkfun.com/products/14001, 2020.

[262] Grigore Stamatescu, Dan Popescu, e Iulia Stamatescu. Modeling for deployment techniques for intra-car wireless sensor networks. Em *2014 18th International Conference on System Theory, Control and Computing (ICSTCC)*, páginas 501–505, Sinaia, Romania, outubro 2014. IEEE.

[263] Aaron Staranowicz e Gian Luca Mariottini. A survey and comparison of commercial and open-source robotic simulator software. Em *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '11*, página 1, Heraklion, Crete, Greece, 2011. ACM Press.

[264] StereoLabs. ZED Stereo Camera | Stereolabs - https://www.stereolabs.com/zed/, fevereiro 2020.

[265] Erik G. Strom. On Medium Access and Physical Layer Standards for Cooperative Intelligent Transport Systems in Europe. *Proceedings of the IEEE*, 99(7):1183–1188, julho 2011.

[266] Sonja Stüdli, María M. Seron, e Richard H. Middleton. Vehicular Platoons in cyclic interconnections with constant inter-vehicle spacing. *IFAC-PapersOnLine*, 50(1):2511–2516, julho 2017.

[267] Jie Sun, Zuduo Zheng, e Jian Sun. Stability analysis methods and their applicability to car-following models in conventional and connected environments. *Transportation Research Part B: Methodological*, 109:212–237, março 2018.

[268] Suwasono Suwasono, Dwi Prihanto, Irawan Dwi Wahyono, e Andrew Nafalski. Virtual Laboratory for Line Follower Robot Competition. *International Journal of Electrical and Computer Engineering (IJECE)*, 7(4):2253, agosto 2017.

[269] Zsolt Szendrei, Norbert Varga, e László Bokor. A SUMO-Based Hardware-in-the-Loop V2X Simulation Framework for Testing and Rapid Prototyping of Cooperative Vehicular Applications. Em *Vehicle and Automotive Engineering 2*, páginas 426–440. Springer International Publishing, Cham, 2018.

[270] Araz Taeihagh e Hazel Si Min Lim. Governing autonomous vehicles: emerging responses for safety, liability, privacy, cybersecurity, and industry risks. *Transport Reviews*, 39(1):103–128, janeiro 2019.

[271] Alireza Talebpour, Hani S. Mahmassani, e Fabián E. Bustamante. Modeling Driver Behavior in a Connected Environment: Integrated Microscopic Simulation of Traffic and Mobile Wireless Telecommunication Systems. *Transportation Research Record: Journal of the Transportation Research Board*, 2560(1):75–86, janeiro 2016.

[272] Fengxiao Tang, Yuichi Kawamoto, Nei Kato, e Jiajia Liu. Future Intelligent and Secure Vehicular Network Toward 6G: Machine-Learning Approaches. *Proceedings of the IEEE*, 108(2):292–307, fevereiro 2020.

[273] N. T. Tangirala, A. Abraham, A. Choudhury, P. Vyas, R. Zhang, e J. Dauwels. Analysis of Packet drops and Channel Crowding in Vehicle Platooning using V2X communication. Em *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, páginas 281–286, Bangalore, India, novembro 2018. IEEE.

[274] Tesla. Transitioning to Tesla Vision, maio 2021.

[275] The Business Research Company. Autonomous Cars Global Market Report 2020-30: COVID-19 Growth and Change. Relatório técnico, The Business Research Company, maio 2020.

[276] J. Thunberg, N. Lyamin, K. Sjöberg, e A. Vinel. Vehicle-to-Vehicle Communications for Platooning: Safety Analysis. *IEEE Networking Letters*, 1(4):168–172, dezembro 2019.

[277] Martin Treiber e Arne Kesting. *Traffic Flow Dynamics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[278] Hsin-Mu Tsai, Ozan K. Tonguz, Cem Saraydar, Timothy Talty, Michael Ames, e Andrew Macdonald. Zigbee-based intra-car wireless sensor networks: a case study. *IEEE Wireless Communications*, 14(6):67–77, dezembro 2007.

[279] S. Tsugawa e S. Kato. Energy ITS: another application of vehicular communications. *IEEE Communications Magazine*, 48(11):120–126, novembro 2010.

[280] Tully Foote. Demo of Prius in ROS/GAZEBO - https://github.com/osrf/car_demo, julho 2017.

[281] S. Ucar, S. C. Ergen, e O. Ozkasap. Security vulnerabilities of IEEE 802.11p and visible light communication based platoon. Em *2016 IEEE Vehicular Networking Conference (VNC)*, páginas 1–4, Columbus, Ohio, USA, dezembro 2016. IEEE. ISSN: 2157-9865.

[282] U.S. DepartmentOf Transportation. FMVSS No. 150 - Vehicle-To-Vehicle Communication Technology For Light Vehicles. Relatório técnico FMVSS No. 150, Office of Regulatory Analysis and Evaluation, National Center for Statistics and Analysis, novembro 2016.

[283] V. Devarapalli, R. Wakikawa, A. Petrescu, e P. Thubert. Request for Comments: 3963, janeiro 2005.

[284] A. S. Valente, U. Montanaro, M. Tufo, A. Salvi, e S. Santini. Design of a Platoon Management Strategy and Its Hardware-In-the Loop Validation. Em *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, páginas 1–5, Seoul, Korea, maio 2014. IEEE.

[285] Pablo Valerio. Europe has defined DSRC WiFi as the V2X standard, and now faces 5G vendors revolt – IoT Times, maio 2019.

[286] Valter Costa, Rosaldo Rossetti, e Armando Sousa. Simulator for Teaching Robotics, ROS and Autonomous Driving in a Competitive Mindset. *International Journal of Technology and Human Interaction*, 13(4):19–32, outubro 2017.

[287] Bart van Arem, Cornelie J. G. van Driel, e Ruben Visser. The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):429–436, dezembro 2006.

[288] Vagisha Vartika, Swati Singh, Subhranil Das, Sudhansu Kumar Mishra, e Sitanshu Sekhar Sahu. A Review on Intelligent PID Controllers in Autonomous Vehicle. Em Dusmanta Kr. Mohanta, M. Jaya Bharata Reddy, Deepak Kumar, e Debomita Ghosh, organizadores, *Advances in Smart Grid Automation and Industry 4.0*, volume 693, páginas 391–399. Springer Singapore, Singapore, 2021. Series Title: Lecture Notes in Electrical Engineering.

[289] Enio Vasconcelos Filho, Ricardo Severino, Anis Koubaa, e Eduardo Tovar. An Integrated Lateral and Longitudinal Look Ahead Controller for Cooperative Vehicular Platooning. Em *Intelligent Transport Systems, From Research and Development to the Market Uptake*, volume 364, páginas 142–159. Springer International Publishing, Cham, 2021. Series Title: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering.

[290] Enio Vasconcelos Filho, Yudi, Jones, Abdelkader, Mohamed, Koubaa, Anis, e Tovar, Eduardo. RosDrive: An Open-source ROS-Based Vehicular Simulator for STEM Control Systems Classes Tutorial (accepted for publication). Em *Robot Operating System (ROS)*, volume VII. Springer International Publishing, Cham, 1 edição, junho 2022.

[291] Frederico C. Vaz, David Portugal, André Araújo, Micael S. Couceiro, e Rui P. Rocha. A localization approach for autonomous underwater vehicles: A ROS-Gazebo framework. *arXiv:1811.05836 [cs]*, novembro 2018. arXiv: 1811.05836.

[292] Benjamin Vedder, Jonny Vinter, e Magnus Jonsson. A Low-Cost Model Vehicle Testbed with Accurate Positioning for Autonomous Driving. *Journal of Robotics*, 2018:1–10, novembro 2018.

[293] Bruno Vieira. A simulation approach for increased safety in advanced C-ITS scenarios. Master Thesis CISTER-TR-191206, Polytechnic Institute of Porto (ISEP P.Porto), 2019.

[294] Bruno Vieira, Ricardo Severino, Enio Vasconcelos Filho, Anis Koubaa, e Eduardo Tovar. COPADRIVe - A Realistic Simulation Framework for Cooperative Autonomous Driving Applications. Em *IEEE International Conference on Connected Vehicles and Expo - ICCVE 2019*, páginas 1–6, Graz, Austria, novembro 2019. IEEE.

[295] Bruno Vieira, Ricardo Severino, Anis Koubaa, e Eduardo Tovar. Towards a Realistic Simulation Framework for Vehicular Platooning Applications. Em *2019 IEEE 22nd International Symposium on Real-Time Distributed Computing (ISORC)*, páginas 93–94, Valencia, Spain, maio 2019. IEEE.

[296] A. Vinel, L. Lan, e N. Lyamin. Vehicle-to-vehicle communication in C-ACC/platooning scenarios. *IEEE Communications Magazine*, 53(8):192–197, agosto 2015.

[297] A. Vinel, N. Lyamin, e P. Isachenkov. Modeling of V2V Communications for C-ITS Safety Applications: A CPS Perspective. *IEEE Communications Letters*, 22(8):1600–1603, agosto 2018.

[298] Vladimir Vukadinovic, Krzysztof Bakowski, Patrick Marsch, Ian Dexter Garcia, Hua Xu, Michal Sybis, Pawel Sroka, Krzysztof Wesolowski, David Lister, e Ilaria Thibault. 3GPP C-V2X and IEEE 802.11p for Vehicle-to-Vehicle communications in highway platooning scenarios. *Ad Hoc Networks*, 74:17–29, maio 2018.

[299] J. Wan, D. Zhang, S. Zhao, L. T. Yang, e J. Lloret. Context-aware vehicular cyber-physical systems with cloud support: architecture, challenges, and solutions. *IEEE Communications Magazine*, 52(8):106–113, agosto 2014.

[300] J. Wang, X. Luo, W. Wong, e X. Guan. Specified-Time Vehicular Platoon Control With Flexible Safe Distance Constraint. *IEEE Transactions on Vehicular Technology*, 68(11):10489–10503, novembro 2019.

[301] Jian Wang, Yameng Shao, Yuming Ge, e Rundong Yu. A Survey of Vehicle to Everything (V2X) Testing. *Sensors*, 19(2):334, janeiro 2019.

[302] L. Wang, R. Iida, e A. M. Wyglinski. Vehicular Network Simulation Environment via Discrete Event System Modeling. *IEEE Access*, 7:87246–87264, 2019.

[303] Yi Wang, Yi Zhou, Wei Li, Gaochao Wang, Lin Ren, e Ruirui Huang. Design of a Cooperative Vehicular Platoon System Based on Zynq/SoC Architecture. Em *Communications and Networking*, volume 210, páginas 335–344. Springer International Publishing, Cham, 2018. Series Title: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering.

[304] Z. Wang, G. Wu, e M. J. Barth. A Review on Cooperative Adaptive Cruise Control (CACC) Systems: Architectures, Controls, and Applications. Em *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, páginas 2884–2891, Maui, Hawaii, USA, novembro 2018. IEEE. ISSN: 2153-0017.

[305] WEG. Learning Workbenches for Training | Teaching Equipment | Teaching Equipment | Electric Panels | WEG - Products, maio 2020. Library Catalog: www.weg.net.

[306] Axel Wegener, Michał Piórkowski, Maxim Raya, Horst Hellbrück, Stefan Fischer, e Jean-Pierre Hubaux. TraCI: an interface for coupling road traffic and network simulators. Em *Proceedings of the 11th communications and networking simulation symposium on - CNS '08*, página 155, Ottawa, Canada, 2008. ACM Press.

[307] S. Wei, Y. Zou, X. Zhang, T. Zhang, e X. Li. An Integrated Longitudinal and Lateral Vehicle Following Control System With Radar and Vehicle-to-Vehicle Communication. *IEEE Transactions on Vehicular Technology*, 68(2):1116–1127, fevereiro 2019.

[308] Shixi Wen. Event-triggered cooperative control of vehicle platoons in vehicular ad hoc networks. *Information Sciences*, 1:13, 2018.

[309] R.E. Wilson e J.A. Ward. Car-following models: fifty years of linear stability analysis – a mathematical perspective. *Transportation Planning and Technology*, 34(1):3–18, fevereiro 2011.

[310] Mintesnot Woldeamanuel e Dang Nguyen. Perceived benefits and concerns of autonomous vehicles: An exploratory study of millennials' sentiments of an emerging market. *Research in Transportation Economics*, 71:44–53, novembro 2018.

[311] Paul Wooderson e David Ward. Cybersecurity Testing and Validation. SAE Technical Paper 2017-01-1655, SAE International, Warrendale, PA, março 2017. ISSN: 0148-7191, 2688-3627.

[312] Q. Wu, H. Ge, P. Fan, J. Wang, Q. Fan, e Z. Li. Time-dependent Performance Analysis of the 802.11p-based Platooning Communications Under Disturbance. *IEEE Transactions on Vehicular Technology*, pre-print:1–1, 2020.

[313] Nian Xia e Chu-Sing Yang. Vehicular Communications: Standards and Challenges. Em Sheng-Lung Peng, Guan-Ling Lee, Reinhard Klette, e Ching-Hsien Hsu, organizadores, *Internet of Vehicles. Technologies and Services for Smart Cities*, volume 10689, páginas 1–12. Springer International Publishing, Cham, 2017. Series Title: Lecture Notes in Computer Science.

[314] Lingyun Xiao e Feng Gao. Practical String Stability of Platoon of Adaptive Cruise Control Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1184–1194, dezembro 2011.

[315] Yu Xie, Michael Fang, e Kimberlee Shauman. STEM Education. *Annual Review of Sociology*, 41(1):331–357, agosto 2015.

[316] Lingwei Xu, Xu Yu, Han Wang, Xinli Dong, Yun Liu, Wenzhong Lin, Xinjie Wang, e Jingjing Wang. Physical Layer Security Performance of Mobile Vehicular Networks. *Mobile Networks and Applications*, 25(2):643–649, abril 2020.

[317] Zhigang Xu, Mingliang Wang, Fengzhi Zhang, Sheng Jin, Jin Zhang, e Xiangmo Zhao. PaTAVTT: A Hardware-in-the-Loop Scaled Platform for Testing Autonomous Vehicle Trajectory Tracking. *Journal of Advanced Transportation*, 2017:1–11, 2017.

[318] Junru Yang, Xingliang Liu, Shidong Liu, Duanfeng Chu, Liping Lu, e Chaozhong Wu. Longitudinal Tracking Control of Vehicle Platooning Using DDPG-based PID. Em *2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, páginas 656–661, Hangzhou, China, dezembro 2020. IEEE.

[319] Shi-Ming Yu, Sai-Nan Wu, Yun-Bo Zhao, e De-Feng He. Delayed Feedback MPC Algorithms of Vehicle Platoons Subject to Constraints on Measurement Range and Driving Behaviors. *Asian Journal of Control*, 20(6):2260–2270, novembro 2018.

[320] T. Yu, S. Zhang, S. Cao, e S. Xu. Performance Evaluation for LTE-V based Vehicle-to-Vehicle Platooning Communication. Em *2018 24th Asia-Pacific Conference on Communications (APCC)*, páginas 15–20, Ningbo, China, novembro 2018. IEEE. ISSN: 2163-0771.

[321] Wenhao Yu, Yifan Zhang, Tinghua Ai, Qingfeng Guan, Zhanlong Chen, e Haixia Li. Road network generalization considering traffic flow patterns. *International Journal of Geographical Information Science*, 34(1):119–149, janeiro 2020.

[322] S. Zeadally, J. Guerrero, e J. Contreras. A tutorial survey on vehicle-to-vehicle communications. *Telecommunication Systems*, 73(3):469–489, março 2020.

[323] Dan Zhang, Ye-Ping Shen, Si-Quan Zhou, Xi-Wang Dong, e Li Yu. Distributed Secure Platoon Control of Connected Vehicles Subject to DoS Attack: Theory and Application. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Pre-print:1–10, 2020.

[324] H. Zhang, J. Liu, Z. Wang, H. Yan, e C. Zhang. Distributed Adaptive Event-Triggered Control and Stability Analysis for Vehicular Platoon. *IEEE Transactions on Intelligent Transportation Systems*, Pre-print:1–12, 2020. Conference Name: IEEE Transactions on Intelligent Transportation Systems.

[325] Kailong Zhang, Xiaowu Li, Ce Xie, Yujia Wang, Liuyang Li, Chao Fei, Arnaud de La Fortelle, e Zongtao Duan. A Scenario-Reconfigurable Simulator for Verifying Service-Oriented Cooperation Mechanisms and Policies of Connected Intelligent Vehicles:. *International Journal of Software Innovation*, 7(1):44–62, janeiro 2019.

[326] Linlin Zhang, Feng Chen, Xiaoxiang Ma, e Xiaodong Pan. Fuel Economy in Truck Platooning: A Literature Overview and Directions for Future Research. *Journal of Advanced Transportation*, 2020:1–10, janeiro 2020.

[327] R. Zhang, A. Abraham, S. Dasgupta, e J. Dauwels. Constrained Model Predictive Control using Kinematic Model of Vehicle Platooning in VISSIM Simulator. Em *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, páginas 721–726, Singapore, novembro 2018. IEEE.

[328] W. Zhang, S. Fu, Z. Cao, Z. Jiang, S. Zhang, e S. Xu. An SDR-in-the-Loop Carla Simulator for C-V2X-Based Autonomous Driving. Em *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, páginas 1270–1271, Online, julho 2020. IEEE.

[329] Yang Zhang e Guohong Cao. V-PADA: Vehicle-Platoon-Aware Data Access in VANETs. *IEEE Transactions on Vehicular Technology*, 60(5):2326–2339, 2011.

[330] C. Zhao, L. Cai, e P. Cheng. Stability Analysis of Vehicle Platooning With Limited Communication Range and Random Packet Losses. *IEEE Internet of Things Journal*, 8(1):262–277, janeiro 2021. Conference Name: IEEE Internet of Things Journal.

[331] C. Zhao, X. Duan, L. Cai, e P. Cheng. Vehicle Platooning With Non-Ideal Communication Networks. *IEEE Transactions on Vehicular Technology*, 70(1):18–32, janeiro 2021. Conference Name: IEEE Transactions on Vehicular Technology.

[332] L. Zhao, X. Li, B. Gu, Z. Zhou, S. Mumtaz, V. Frascolla, H. Gacanin, M. I. Ashraf, J. Rodriguez, M. Yang, e S. Al-Rubaye. Vehicular Communications: Standardization and Open Issues. *IEEE Communications Standards Magazine*, 2(4):74–80, dezembro 2018. Conference Name: IEEE Communications Standards Magazine.

[333] Yingbo Zhao, Paolo Minero, e Vijay Gupta. On disturbance propagation in leader–follower systems with limited leader information. *Automatica*, 50(2):591–598, fevereiro 2014.

[334] Y. Zheng, S. E. Li, K. Li, F. Borrelli, e J. K. Hedrick. Distributed Model Predictive Control for Heterogeneous Vehicle Platoons Under Unidirectional Topologies. *IEEE Transactions on Control Systems Technology*, 25(3):899–910, maio 2017.

[335] Zuduo Zheng. Freeway Traffic Oscillations: Microscopic Analysis of Formations and Propagations using Wavelet Transform. *Procedia Social and Behavioral Sciences*, 1:15, 2011.

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

[337] Meixin Zhu, Xuesong Wang, e Yinhai Wang. Human-like autonomous car-following model with deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 97:348–368, dezembro 2018.

[338] P. Zhu, K. Zhu, e L. Zhang. Security Analysis of LTE-V2X and A Platooning Case Study. Em *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, páginas 532–537, Online, julho 2020. IEEE.

[339] S. Zhu, D. Goswami, e H. Li. Evaluation Platform of Platoon Control Algorithms in Complex Communication Scenarios. Em *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, páginas 1–5, Kuala Lumpur, Malaysia, abril 2019. IEEE.

[340] Y. Zhu, J. Wu, e H. Su. V2V-Based Cooperative Control of Uncertain, Disturbed and Constrained Nonlinear CAVs Platoon. *IEEE Transactions on Intelligent Transportation Systems*, Pre-print:1–11, 2020.

[341] Sinan Öncü, Nathan van de Wouw, W. P. Maurice H. Heemels, e Henk Nijmeijer. String stability of interconnected vehicles under communication constraints. Em *IEEE Conference on*

Decision and Control (CDC),2012, páginas 2459–2464, Maui, HI, USA, dezembro 2012. ISSN: 0743-1546.