



VEHICULAR 2022

The Eleventh International Conference on Advances in Vehicular Systems,
Technologies and Applications

ISBN: 978-1-61208-974-4

May 22nd –26th, 2022

Venice, Italy

VEHICULAR 2022 Editors

Reiner Kriesten, Karlsruhe University of Applied Sciences, Germany

VEHICULAR 2022

Forward

The Eleventh International Conference on Advances in Vehicular Systems, Technologies and Applications (VEHICULAR 2022) continued a series of events considering the state-of-the-art technologies for information dissemination in vehicle-to-vehicle and vehicle-to-infrastructure and focusing on advances in vehicular systems, technologies and applications.

Mobility brought new dimensions to communication and networking systems, making possible new applications and services in vehicular systems. Wireless networking and communication between vehicles and with the infrastructure have specific characteristics from other conventional wireless networking systems and applications (rapidly changing topology, specific road direction of vehicle movements, etc.). These led to specific constraints and optimizations techniques; for example, power efficiency is not as important for vehicle communications as it is for traditional ad hoc networking. Additionally, vehicle applications demand strict communications performance requirements that are not present in conventional wireless networks. Services can range from time-critical safety services, traffic management, to infotainment and local advertising services. They are introducing critical and subliminal information. Subliminally delivered information, unobtrusive techniques for driver's state detection, and mitigation or regulation interfaces enlarge the spectrum of challenges in vehicular systems.

We take here the opportunity to warmly thank all the members of the VEHICULAR 2022 technical program committee, as well as all the reviewers. The creation of such a high-quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to VEHICULAR 2022. We truly believe that, thanks to all these efforts, the final conference program consisted of top-quality contributions. We also thank the members of the VEHICULAR 2022 organizing committee for their help in handling the logistics of this event.

VEHICULAR 2022 Chairs

VEHICULAR 2022 Steering Committee

Khalil El-Khatib, University of Ontario Institute of Technology – Oshawa, Canada

Éric Renault, ESIEE Paris, France

Xiaohong Peng, Birmingham City University, UK

Yuping He, University of Ontario Institute of Technology, Canada

VEHICULAR 2022 Publicity Chairs

Mar Parra, Universitat Politècnica de València (UPV), Spain

Hannah Russell, Universitat Politècnica de València (UPV), Spain

VEHICULAR 2022 Committee

VEHICULAR 2022 Steering Committee

Khalil El-Khatib, University of Ontario Institute of Technology – Oshawa, Canada
Éric Renault, ESIEE Paris, France
Xiaohong Peng, Birmingham City University, UK
Yuping He, University of Ontario Institute of Technology, Canada

VEHICULAR 2022 Publicity Chairs

Mar Parra, Universitat Politècnica de València (UPV), Spain
Hannah Russell, Universitat Politècnica de València (UPV), Spain

VEHICULAR 2022 Technical Program Committee

Nor Fadzilah Abdullah, National University of Malaysia (UKM), Malaysia
Mohammed Al-Ansi, University Malaysia Perlis (UniMAP), Malaysia
Sufyan T. Faraj Al-Janabi, University of Anbar, Ramadi, Iraq
Mustafa S. Al-Jumaily, University of Tennessee, Knoxville, USA
Ali Alfoudi, Al-Qadisiyah University, Iraq
Ala'a Al-Momani, Ulm University, Germany
Bhaskar Anand, Indian Institute of Technology Hyderabad, India
Adel Aneiba, Birmingham City University, UK
Muhammad Asim, Chung-Ang University, Seoul, South Korea
Hakan Aydin, Karadeniz Technical University, Turkey
Eduard Babulak, Fort Hays State University, USA
Manlio Bacco, CNR-ISTI, Italy
Andrea Baiocchi, University of Roma "Sapienza", Italy
Ali Balador, RISE Research Institute of Sweden, Sweden
Paolo Barsocchi, ISTI (Institute of Information Science and Technologies) | Italian National Research Council (C.N.R.), Pisa, Italy
Paulo C. Bartolomeu, University of Aveiro, Portugal
Marcel Baunach, Graz University of Technology, Austria
Rahim (Ray) Benekohal, University of Illinois at Urbana-Champaign, USA
Sylvia Bhattacharya, Kennesaw State University, USA
Eugen Borcoci, University Politehnica of Bucharest, Romania
Alexandros-Apostolos A. Boulogeorgos, University of Piraeus, Greece
Christos Bouras, University of Patras, Greece
Roberto Caldelli, CNIT, Florence, Italy
Rodrigo Capobianco Guido, São Paulo State University (UNESP), Brazil
Antonio Carcaterra, Sapienza University of Rome, Italy
Pedro Cardoso, Universidade do Algarve, Portugal
Marcos F. Caetano, University of Brasilia, Brazil
Florent Carlier, Centre de Reserche en Education de Nantes / Le Mans Université, France
Juan Carlos Ruiz, Universitat Politecnica de Valencia, Spain

Rodrigo Castelan Carlson, Federal University of Santa Catarina, Brazil
Francois Chan, Royal Military College, Canada
Claude Chaudet, Webster University Geneva, Switzerland
Rui Chen, Xidian University, China
Gihwan Cho, Jeonbuk University, Korea
Adam Cohen, University of California, Berkeley, USA
Gianpiero Costantino, Institute of Informatics and Telematics (IIT) | National Research Council (CNR), Italy
Yousef-Awwad Daraghmi, Palestine Technical University-Kadoorie, Palestine
David de Andrés, Universitat Politècnica de València, Spain
Fawad Ud Din, McGill University, Canada
Liza Dixon, Independent Researcher, Germany
Zoran Duric, George Mason University, USA
Péter Ekler, Budapest University of Technology and Economics, Hungary
Suzi Iryanti Fadilah, Universiti Sains Malaysia, University
Mariano Falcitelli, Photonic Networks & Technologies National Laboratory of CNIT, Italy
Abraham O. Fapojuwo, University of Calgary, Canada
Gustavo Fernandez Dominguez, Center for Digital Safety & Security | AIT Austrian Institute of Technology, Austria
Irene Fiesoli, University of Florence, Italy
Miguel Franklin de Castro, Federal University of Ceará, Brazil
Tomonari Furukawa, University of Virginia, USA
Varun Garg, UMass Lowell, USA
Pedro Pablo Garrido Abenza, Universidad Miguel Hernandez de Elche, Spain
Malek Ghanes, Centrale Nantes, France
Apostolos Gkamas, University Ecclesiastical Academy of Vella of Ioannina, Greece
Sezer Goren, Yeditepe University, Turkey
Alberto Gotta, National Research Council - Institute of Information Science and Technologies “A. Faedo” (CNR-ISTI), Italy
Heinrich Gotzig, Valeo, Germany
Javier Gozalvez, Universidad Miguel Hernandez de Elche, Spain
Abel Guilhermino da Silva Filho, Federal University of Pernambuco - UFPE, Brazil
Rami Hamdi, Hamad Bin Khalifa University, Qatar Foundation, Qatar
Kyungtae (KT) Han, Toyota North America, USA
Petr Hanáček, Brno University of Technology, Czech Republic
Hong Hande, Huawei Technologies, Singapore
Yuping He, University of Ontario Institute of Technology, Canada
Gonçalo Homem de Almeida Correia, TU Delft, Netherlands
Javier Ibanez-Guzman, Renault S.A., France
Hocine Imine, IFSTTAR/LEPSIS, France
Uzair Javaid, National University of Singapore, Singapore
Dush Nalin Jayakody, Tomsk Polytechnic University, Russia
Terje Jensen, Telenor, Norway
Yiming Ji, Georgia Southern University, USA
Felipe Jiménez Alonso, Technical University of Madrid, Spain
Magnus Jonsson, Halmstad University, Sweden
Filbert Juwono, Curtin University, Malaysia
Gorkem Kar, Bahcesehir University, Turkey

Frank Kargl, Institute of Distributed Systems | Ulm University, Germany
Sokratis K. Katsikas, Norwegian University of Science and Technology, Norway
Tetsuya Kawanishi, Waseda University, Japan
John Kenney, Toyota InfoTech Labs, USA
Norazlina Khamis, Universiti Malaysia Sabah, Malaysia
Asil Koc, McGill University, Montreal, Canada
Lisimachos Kondi, University of Ioannina, Greece
Spyros Kontogiannis, University of Ioannina, Greece
Anastasios Kouvelas, ETH Zürich, Switzerland
Zdzislaw Kowalczyk, Gdansk University of Technology, Poland
Francine Krief, Bordeaux INP, France
Reiner Kriesten, University of Applied Sciences Karlsruhe, Germany
Ryo Kurachi, Nagoya University, Japan
Gyu Myoung Lee, Liverpool John Moores University, UK
Sebastian Lindner, Hamburg University of Technology, Germany
Lianggui Liu, Zhejiang Shuren University, China
Tomasz Mach, Samsung R&D Institute, UK
Dalila B. Megherbi, University of Massachusetts, USA
Zoubir Mammeri, IRIT - Paul Sabatier University, France
Sunil Manvi, REVA University, India
Mirco Marchetti, University of Modena and Reggio Emilia, Italy
P. Takis Mathiopoulos, University of Athens, Greece
Ioannis Mavromatis, University of Bristol, UK
Rashid Mehmood, King Abdul Aziz University, Saudi Arabia
José Manuel Menéndez, Universidad Politécnica de Madrid, Spain
Maria Luisa Merani, University of Modena and Reggio Emilia, Italy
Lyudmila Mihaylova, The University of Sheffield, UK
Gerardo Mino Aguilar, Benemérita Universidad Autónoma De Puebla, Mexico
Naveen Mohan, KTH, Sweden
Bruno Monsuez, ENSTA ParisTech, France
Luís Moutinho, Escola Superior de Gestão e Tecnologia de Águeda (ESTGA) - University of Aveiro /
Instituto de Telecomunicações, Portugal
Vasco N. G. J. Soares, Instituto de Telecomunicações / Instituto Politécnico de Castelo Branco, Portugal
Jose Eugenio Naranjo Hernandez, Universidad Politécnica de Madrid | Instituto Universitario de
Investigación del Automóvil (INSIA), Spain
Ridha Nasri, Orange Labs, France
Keivan Navaie, Lancaster University, UK
Patrik Österberg, Mid Sweden University, Sweden
Antonio M. Pascoal, Institute for Systems and Robotics - IST | Univ. Lisbon, Portugal
Al-Sakib Khan Pathan, Southeast University, Bangladesh
Xiaohong Peng, Birmingham City University, UK
Paulo J. G. Pereirinha, Polytechnic of Coimbra | IPC/ISEC | INESC Coimbra, Portugal
Fernando Pereñíguez García, University Defense Center | Spanish Air Force Academy, Murcia, Spain
Valerio Persico, University of Napoli Federico II, Italy
Paulo Pinto, Universidade Nova de Lisboa, Portugal
Srinivas Pulugurtha, The University of North Carolina at Charlotte, USA
Hesham Rakha, Virginia Tech, USA
Mohd Fadlee A Rasid, Universiti Putra Malaysia, Malaysia

Éric Renault, ESIEE Paris, France
Martin Ring, Robert Bosch GmbH, Germany
Geraldo P. Rocha Filho, University of Brasília, Brazil
Justin P. Rohrer, Naval Postgraduate School, USA
Aymeric Rousseau, Argonne National Laboratory, USA
Javier Rubio-Loyola, CINVESTAV, Mexico
João Rufino, Instituto de Telecomunicações - Pólo Aveiro, Portugal
José Santa Lozano, Technical University of Cartagena (UPCT), Spain
José Santa, Technical University of Cartagena, Spain
Nico Saputro, Parahyangan Catholic University, Bandung, Indonesia
Marios Savvides, Carnegie Mellon University, USA
Erwin Schoitsch, AIT Austrian Institute of Technology GmbH, Austria
Alireza Shahrabi, Glasgow Caledonian University, Scotland, UK
Rajan Shankaran, Macquarie University, Australia
Prinkle Sharma, University of Massachusetts Dartmouth, USA
Shih-Lung Shaw, University of Tennessee Knoxville, USA
Dana Simian, Lucian Blaga University of Sibiu, Romania
Pranav Kumar Singh, Central Institute of Technology Kokrajhar, India
Marko Sonkki, Ericsson, Germany
Chokri Souani, Higher Institute of Applied Sciences and Technology of Sousse, Tunisia
Essam Sourour, Prince Sattam Bin Abdul-Aziz University (PSAU), Saudi Arabia
Mujdat Soyturk, Marmara University, Turkey
Anand Srivastava, IIIT Delhi, India
Dario Stabili, University of Modena and Reggio Emilia, Italy
Pawan Subedi, The University of Alabama, Tuscaloosa, USA
Qasim Sultan, Chung-Ang University, Seoul, South Korea
Qingquan Sun, California State University San Bernardino, USA
Akimasa Suzuki, Iwate Prefectural University, Japan
Philipp Svoboda, TU Wien, Austria
Wai Yuen Szeto, The University of Hong Kong, Hong Kong
Getaneh Berie Tarekegn, National Taipei University of Technology, Taiwan
Angelo Trotta, University of Bologna, Italy
Bugra Turan, Koc University, Istanbul, Turkey
Markus Ullmann, Federal Office for Information Security / University of Applied Sciences Bonn-Rhine-Sieg, Germany
Klaus David, University of Kassel, Germany
Costin Untaroiu, Virginia Tech, USA
Massimo Villari, Università di Messina, Italy
Ankur Vora, Intel, SanJose, USA
Hong Wang, Oak Ridge National Laboratory | US Department of Energy, USA
You-Chiun Wang, National Sun Yat-sen University, Taiwan
Duminda Wijesekera, George Mason University, USA
Ramin Yahyapour, Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG), Germany
Shingchern D. You, National Taipei University of Technology, Taiwan
Salim M Zaki, Dijlah University, Baghdad, Iraq
David Zage, Intel Corporation, USA
Christos Zaroliagis, University of Patras, Greece

Sherali Zeadally, University of Kentucky, USA

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission to reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Transmission Range Influence on Secure Routing in VANETs <i>Afef Slama and Ilhem Lengliz</i>	1
Near-Optimal Coordination of Vehicles at an Intersection Plaza Using Bezier Curves <i>Elham Ahmadi, Rodrigo Castelan Carlson, Werner Kraus Junior, and Ehsan Taheri</i>	6
CAN Demonstrator with Intrusion Detection System <i>Heiko Polster and Dirk Labudde</i>	13
A Smart Control Strategy for a Battery Thermal Management System: Design, Validation and Implementation <i>Mikel Arrinda, Gorka Vertiz, Christophe Morel, Nicolas Hascoet, Pierre Woltmann, and Alois Sonnleitner</i>	19
Event-triggered Robust Output Feedback Controller for a Networked Roll Control System <i>Fernando Viadero-Monasterio, Manuel Jimenez-Salas, Miguel Melendez-Useros, Javier Garcia-Guzman, Beatriz Lopez-Boada, and Maria Jesus Lopez-Boada</i>	25
A Multi-UAS Simulator for High Density Air Traffic Scenarios <i>David Martin-Lammerding, Jose Javier Astrain, and Alberto Cordoba</i>	32
Simulation-based Testing of Service Drones in U-Space Environments <i>Moritz Hoser and Kai-Daniel Buchter</i>	38
Challenges for Periodic Technical Inspections of Intelligent Cars <i>Mona Gierl, Felix Muller, Reiner Kriesten, Philipp Nenninger, and Eric Sax</i>	41

Transmission Range Influence on Secure Routing in VANETs

Afef Slama
 HANA Laboratory
 University of Manouba
 Manouba, Tunisia
 e-mail: slama.afef@hotmail.fr

Ilhem Lengliz
 Computer Science Department
 Military Academy
 Nabeul, Tunisia
 e-mail: ilhem.lengliz@gmail.com

Abstract—With the rapid development in smart vehicles, the security and privacy issues of the Vehicular Ad-hoc Network (VANET) have drawn noteworthy regard. Indeed, every secure routing protocol must suit its operation to meet VANETs requirements and to yield a better security level. Incorporating Route Life Time (RLT) policy to Dynamic Source Routing (DSR) routing protocol is one of these adaptations. This policy intends to improve the global route lifetime. Trust Cryptographic Secure Routing (TCSR) protocol is one more proposition for secure routing found on the selection of the most trustworthy node all along with the route establishment. In this paper, we propose a comparative study of DSR-RLT and TCSR routing protocols on a highway to evaluate their performances in terms of transmission range variation. The simulation results show that TCSR exceeds DSR-RLT in terms of the packet loss ratio, average network throughput, and average delay.

Keywords-VANET; secure routing; RLT; DSR-RLT; TCSR.

I. INTRODUCTION

VANETs refer to the Intelligent Transportation System (ITS) where vehicles are intelligent objects communicating (sending and receiving data) between each other in a smart manner [1]. Their purpose is to assist road users with appropriate services like safety, infotainment, and traffic management, by incorporating information and communication technology into vehicles and transport infrastructure. The transmission of messages in an open-access environment like VANETs leads to the most critical and challenging security issues [2]. As a result, the design of an effective secure routing protocol for VANET is crucial. So, the major threat is to design a robust and efficient secure routing algorithm that is very adaptable to frequent changes in the topology of fast-moving vehicles [3].

Diverse routing protocols have been proposed for VANETs to address the nodes' powerful mobility. Unfortunately, since most of these routing protocols use nodes succession during the route foundation among the source and the destination, the nature of communications characterized by a short duration may provide frequent disconnections. To handle this challenge, we provide in this work a comparison between two protocols we have proposed: the DSR-RLT protocol proposed in [4] and the TCSR protocol [5]. These protocols are found on the increase of the routing process's vigor toward regular

common disconnections. The rest of this paper is organized as follows. Section II explores the DSR-RLT protocol. Section III gives a summary of the TCSR proposal. Section IV presents a comparison of the two protocols. The conclusion closes the article.

II. DSR-RLT

In an Intelligent Transportation System (ITS), each vehicle can be the sender, the receiver, or the router of every broadcasted message in the VANET. It is fundamental to secure the routing information since this information can be modified by malicious nodes. One of these secure routing protocols is DSR-RLT which is an enhancement of the native DSR routing protocol [6] using the Route Life Time (RLT) policy proposed in [7]. This policy seeks the optimal choice of the next hop based on the node's speed and the inter-node distances for a given approximation of the optimal number of hops in a VANET. When integrated into a routing operation, this policy tries to find an optimal choice of the next hop (relay node) in order to maximize the associated link lifetime and, hence, the overall route lifetime. Indeed, when invoked for a route building, the DSR-RLT protocol begins to look for the most favorable node positioning so as to cope with the RLT policy requirements such that the formed route for data transmissions will have the longest life time among all possible routes. In the same manner, the DSR-RLT protocol acts as the DSR protocol in establishing a route for a data packet to be sent on the VANET.

The preliminary evaluation of DSR-RLT protocol we carried out in [4] has shown that it achieves a higher network throughput in a realistic environment, especially on a crowded highway.

III. TCSR PROTOCOL

The trust metric has been proposed in various works addressing the secure routing in VANETs [8]-[10]. In this context, we have designed the Trust Cryptographic Secure Routing protocol (TCSR). Its operation takes place in two phases, as shown in Fig. 1 below. The first phase aims to create a high trust-surrounding level for each node in the VANET. It initiates calculating the trust level (TL_v) of each node in a dynamic and distributed model. Thereby, every vehicle is capable to assign a TL_v to every vehicle in its vicinity. Indeed, the evaluation of the behavior of a node is defined upon the interchanged packets. Therefore, depending

on the result of the overseeing process, the TL_v of each vehicle can rise, decline or stay fixed.

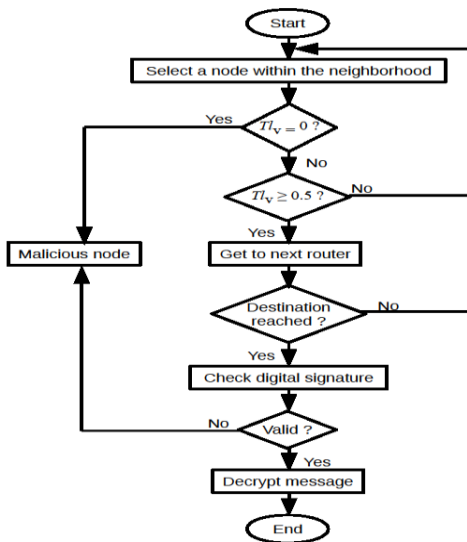


Figure 1. Flowchart of the TCSR operation process.

The second phase aims to enhance the security level of the TCSR model using asymmetric cryptography. Thus, the source information is ciphered using the public key of the sender to generate a digital signature. Therefore, the receiver authenticates the sender before decrypting the received message.

To compute the TL_v , the TCSR operation process is based on the reuse of the Additive Increase/Multiplicative-Decrease (AIMD) [11] technique along with the 3 DUP PKL (PacKet Losses) principle derived from the DUP ACK TCP congestion control mechanism. As illustrated in Fig. 2 below, at first, TCSR confesses that each vehicle in the transmission range (T_r) has a basic TL_0 in $[0, 1]$ that may vary in time or over the routing process.

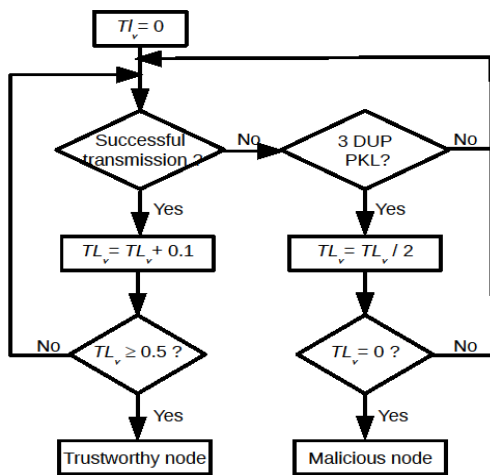


Figure 2. Flowchart of the node trust level monitoring.

The TCSR proposal allows each node to communicate with the others using a series of plausibility checks enabling them to compute the node score before selecting the one having the highest score. Consequently, the safest node is able to broadcast every signed message. In regard to RLT policy, it grants the choice of the route with the longest life time among all possible routes during the routing process of DSR-RLT.

IV. COMPARISON

It is interesting to evaluate the performances of the trust metric of the TCSR protocol on a highway and to compare its performances with those of DSR-RLT in order to verify the impact of the evolution on these two protocols.

A. Simulation model and parameters

For this study, we designed the VANETs scenario using Simulation of Urban Mobility (SUMO) [12]. We then converted the resulting SUMO trace file to be the data file used in NS-3.27. The objective of this simulation is to study the impact of the variation of the transmission range on a secure VANET routing protocol in order to evaluate its performance under different transmission ranges with variable data rates.

For the purpose of this study, we defined a VANET model with the parameters listed in Table 1 below. Two hundred nodes (vehicles) with a speed of 110 km/h were tested in the scenario to determine the impact of the network density on the TCSR and DSR-RLT secure routing process.

TABLE I. SIMULATION PARAMETERS OF A HIGHWAY

Parameter	Value
MAC layer	MAC IEEE 802.11p
Node buffer size	50 packets
Propagation model	Two Ray Ground
Network bandwidth	6 Mbps
Packet length	100, 200 & 512 Kb
Communication range	100 - 700 m
Highway length	6 km
Number of lanes	6 (3 in each direction)
Time of simulation	1800 sec

The performance indicators we selected to evaluate the two protocols in different VANETs scenarios are as follows: the Packet Loss Ratio (PLR), the average network throughput, the delay and the total energy consumed, which are the most relevant parameters commonly used to evaluate any given routing protocol in VANETs.

- Packet Loss Ratio (PLR): it is the loss rate of message delivery among vehicles within the same range of communication using single-hop messaging.

- Average Network Throughput: is the total payload over the entire session divided by the total time. Total time is calculated by taking the difference in timestamps between the first and last packet.
- Average Delay: it represents the period of time spent to route a packet from the source to the destination. That is the ratio of the number of sending bits in the packet to the throughput.
- Total Energy Consumed: it measures the total energy consumed by nodes during the routing process.

B. Simulations results

The purpose of this section is to examine the behavior of the TCSR and DSR-RLT protocols according to the variation of the transmission range and the traffic load.

Packet Loss Ratio (PLR): Fig. 3 describes the behavior of both protocols as a result of varying transmission range and transmitted packet size. We show that for a transmission range strictly less than or equal to 500 m, the value of PLR is inversely proportional to the value of the range for both protocols. It becomes proportional for a range strictly greater than 500 m, always for both protocols. This result is logical if we know that increasing the range of transmission with the maintenance of the number of vehicles reduces the number of jumps and thus ensures better connectivity that results in higher signal strength. On the other hand, when the transmission range exceeds 500 m, the conflict flow increases at the MAC layer resulting in a higher interference rate. Nevertheless, the CSMA/CA rules limit communication to many nodes avoiding collisions, which limit the reuse of bandwidth.

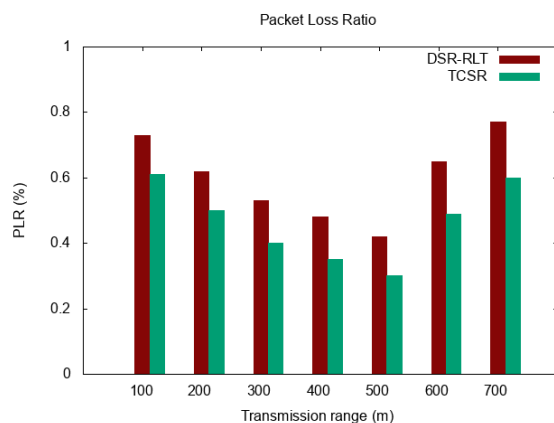


Figure 3. PLR vs Transmission range.

Fig. 3 shows the adaptability of the two protocols to more realistic road scenarios by adjusting the transmission range. However, it reveals better performance in terms of PLR for the TCSR protocol. This result is predictable because TCSR is essentially based on the value of the confidence level Th_v of each vehicle. Thus, all relay nodes that it chooses during the routing process have the highest levels of trust. As a result, the number of internal attacks is reduced, which guarantees the successful transfer of the packets of the data signed by the CA to minimize external attacks. On a

congested highway, TCSR differs from DSR-RLT in that it adjusts the confidence level of network nodes quickly with plausibility check series initiated. The longer the transmission range, the higher the number of vehicles traveling at high speeds, which increases the response time.

In addition, the lower performance of the DSR-RLT protocol is explained by the fact that the source routing keeps the complete path between the source and the destination in the header of the data packet. Besides, the use of the RLT policy which seeks the optimal choice of the next hop according to the speed of the node and the inter-node distances induces a loss of time during the phase of the construction of the road. The optimal choice of the relay node maximizes the lifetime of the link and therefore the overall life of the route but does not address the problem of internal and external attacks. As a result, the transmitted messages may be modified during the routing process which causes the loss of data.

Average Network Throughput: Fig. 4 shows the influence of the change in transmission range on the average network throughput following the deployment of the TCSR and DSR-RLT protocols. The PLR influences in the sense that the decrease of the PLR increases the flow. Thus, and as illustrated in Fig. 4, it increases for ranges less than or equal to 500 m and decreases for ranges strictly greater than 500 m.

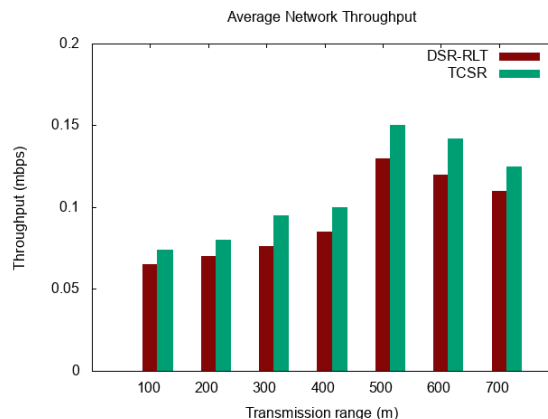


Figure 4. Average Network Throughput vs Transmission range.

We note that the average network throughput measured with the TCSR protocol decreases for the 600 m and 700 m transmission ranges, remains higher than the one provided by DSR-RLT. Indeed, the use of the AIMD mechanism for calculating the trust level of the relay nodes in addition to the digital signatures ensures better stability of the route during the routing of the data. It turns out that the detection of internal and external attacks throughout the routing increases network throughput.

Average Delay: Fig. 5 shows the behavior of both protocols as a result of varying transmission range and transmitted packet size. We show that for a transmission range strictly less than or equal to 500 m, the value of Average Delay is inversely proportional to the value of the range for both protocols. It becomes proportional for a range strictly greater than 500 m, always for both protocols. This

result is logical if we know that increasing the transmission range while maintaining the number of vehicles decreases the number of hops and thus guarantees greater connectivity that results in higher signal strength. On the other hand, when the transmission range exceeds 500 m, the conflict flow increases at the MAC layer resulting in a higher interference rate. Nevertheless, the CSMA/CA rules limit communication to many nodes avoiding collisions which limits the reuse of bandwidth.

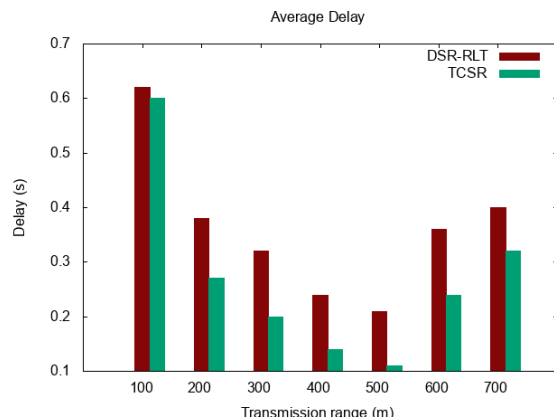


Figure 5. Average Delay vs Transmission range.

Fig. 5 shows the adaptability of the two protocols to more realistic road scenarios by adjusting the transmission range. However, it reveals better performance in terms of Average Delay for the TCSR protocol. This result is predictable because TCSR is fundamentally based on the value of the trust level T_v of each vehicle. Thus, all relay nodes that it chooses during the routing process have the highest levels of trust. As a result, the number of internal attacks is reduced, which guarantees the successful transfer of the packets of the data signed by the CA to minimize external attacks. On a congested highway, TCSR differs from DSR-RLT in that it adjusts the confidence level of network nodes quickly with plausibility check series initiated. The longer the transmission range, the higher the number of vehicles traveling at high speeds, which increases the response time.

In addition, the lower performance of the DSR-RLT protocol is explained by the fact that the source routing keeps the complete path between the source and the destination in the header of the data packet. Besides, the use of the RLT policy which seeks the optimal choice of the next hop according to the speed of the node and the inter-node distances induces a loss of time during the phase of the construction of the road. The optimal choice of the relay node maximizes the lifetime of the link and therefore the overall life of the route but does not address the problem of internal and external attacks. As a result, the transmitted messages may be modified during the routing process which causes the loss of data.

Total Consumed Energy: Fig. 6 shows the influence of the change in transmission range on the amount of Total Consumed Energy following deployment of the TCSR and DSR-RLT protocols. Thus, and as illustrated in Fig. 6, TCSR has the uppermost amount of consumed energy which is also

expected since the RLT policy reduces the number of control packets generated to establish a route between a source and a destination. While the TCSR employs a trust metric and a cryptography strategy known for their complexity. But we should recall that this may not affect the network overall status given that vehicles in VANETs are equipped with OBUs and batteries.

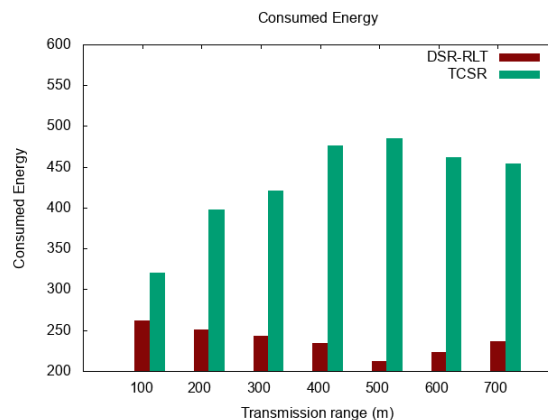


Figure 6. Total Consumed Energy vs Transmission range.

The TCSR protocol is suitable for managing a fast and continuous network topology. Indeed, this protocol makes it possible to deliver more packet than the protocol DSR-RLT, it also reaches better performances in term of flow for a T_r equal to 500 m.

V. CONCLUSION

In this paper, we presented a comparison between the TCSR protocol that uses the trust metric and the DSR-RLT protocol based on RLT policy. We chose to compare their respective performances on a congested highway for different transmission ranges. We found that the TCSR protocol is better adapted to scalability due to its performance in terms of PLR, average network throughput and average delay for transmission range values up to 500 m. However, it indicates a poor effect in terms of consumed energy compared to DSR-RLT for all transmission ranges. For transmission range values strictly greater than 500 m, a study should be developed based on the variation of simulation parameters such as bandwidth and data packet size.

REFERENCES

- [1] F. Cunha, L.Villas, A. Boukerche, G. Maia, A. Viana, R.A.F. Mini, and A.A.F. Loureiro, "Data communication in vanets: survey, applications and challenges", *Ad Hoc Networks journal*, vol. 44, pp 90 - 103, 2016.
- [2] S. S. Manvi and S. Tangade, "A survey on authentication schemes in VANETs for secured communication", *Vehicular Communications*, vol. 9, pp. 19-30, July 2017.
- [3] Y. Hammouche and S. Merniz, "Vanet cross layer routing", 2019 International Conference of Computer Science and Renewable Energies (ICCSRE), July 2019.
- [4] I. Lengliz and A. Slama, "Enhancing VANETs' Routing Operation with the Route Lifetime Policy", *International*

- Journal of Computer Applications, vol. 164, pp. 35-40, April 2017.
- [5] A. Slama, I. Lengliz and A. Belghith, "TCSR: an AIMD Trust-based Protocol for Secure Routing in VANET", The 2018 International Conference on Smart Applications, Communications and Networking, November 2018.
 - [6] D. Johnson, Y. Hu, D. Maltz, RFC 4728, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4", February 2007.
 - [7] D. Kumar, A. A. Kherani, E. Altman. "Route Life time based Interactive Routing in Intervehicle Mobile Ad Hoc Networks", Research Report, INRIA, France, September 2005.
 - [8] T. Gazdar, A. Rachedi, A. Benslimane, and A. Belghith, "A distributed advanced analytical trust model for vanets", 2012 IEEE Global Communications Conference (GLOBECOM), December 2012.
 - [9] A. Chinnasamy, S.Prakash and P.Selvakumari, "Enhance trust based routing techniques against sinkhole attack in AODV based VANET", International Journal of Computer Applications, vol. 65, issue 15, March 2013.
 - [10] T. Gazdar, A. Belghith, and H. Abutar, "An Enhanced and Distributed Trust Computing Protocol for VANETs", IEEE Access, vol. 6, pp. 380-392, October 2017.
 - [11] RFC5681, TCP Congestion Control.
 - [12] <http://sumo.sourceforge.net>. Accessed April 2022.

Near-Optimal Coordination of Vehicles at an Intersection Plaza Using Bézier Curves

Elham Ahmadi
Rodrigo Castelan Carlson
Werner Kraus Junior
Federal University of Santa Catarina
Florianópolis, SC, Brazil
email: elham.ahmadi@posgrad.ufsc.br
email: {rodrigo.carlson,werner.kraus}@ufsc.br

Ehsan Taheri
Department of Aerospace Engineering
Auburn University
Auburn, AL, USA
email: etaheri@auburn.edu

Abstract—We investigate the problem of optimal coordination of Connected Vehicles under Automated Driving (CVAD) at intersections. We aim for more driving flexibility for the CVAD, with the possibility of fully utilizing the intersection space, while strictly avoiding collisions. We propose the Intersection Trajectories Optimal control Problem (ITOP), in which an intersection is a space without movement-related horizontal markings or structural restrictions, except for the intersection boundaries, which we call a *plaza*. By using the Bézier curves and discretization, we convert the ITOP to a Non-Linear Program (NLP) that generates near-optimal collision-free trajectories. Numerical results demonstrate that the proposed approach generates feasible trajectories and is suitable for the solution of the ITOP.

Index Terms—*intersection Plaza; Bézier curves; optimal control; non-linear programming.*

I. INTRODUCTION

In spite of the changes taking place in traffic systems due to the emergence of Connected Vehicles under Automated Driving (CVAD), the paradigm established decades ago for the organization of road infrastructure remains roughly the same. Most of the vehicle coordination strategies and intersection models proposed in this new context [1] [2] rely on the concept of vehicle movements, thus restricting the possible or allowed paths within the intersection. This configures a waste of scarce intersection space and a loss of efficiency.

The proposed approaches for modeling intersections can be summarized in three categories [1] [3]:

- Cells: the intersection is divided into cells while time windows in each cell are allocated to vehicles so that there are no collisions (Fig. 1a). See, e.g., [4] [5].
- Paths: a limited set of paths are allocated to vehicles in such a way that vehicles on conflicting paths do not collide (Fig. 1b). See, e.g., [6].
- Conflicting regions/points: only the points or regions where conflicts between paths occur are discretized and time windows are allocated to vehicles to pass through these points (Fig. 1c). See, e.g., [7]–[9].

Moreover, most of these strategies simplify the behavior within the intersection to consider a constant speed and some even preclude turning movements.

In essence, these models deal with how to allocate time windows of the scarce intersection space to different vehicles. However, despite the higher capacity obtained by the elimination of the traffic light cycle and by the smaller headways between vehicles enabled by CVAD, the capacity of the intersection ends up limited by the relationship between paths constrained to pre-established vehicular movements.

Better use of the infrastructure is possible if the vehicles are allowed to make full use of the intersection space via the definition of their trajectories for any possible path. The research on this subject is limited. In [10], a cooperative motion-planning method was proposed for Connected and Automated Vehicles (CAVs) crossing an intersection. The solution of a centralized optimal control problem provides optimized trajectories offline based on pre-defined formations to which the CAVs are guided online before entering the intersection. The high-dimensionality and non-linearity of the model called for the convexification of the collision-avoidance constraints and an algorithm that provides suitable trajectories as initial guesses for the problem to speed up the solution [11]. An approach relying on emergent behaviors modeled vehicle behavior by a hierarchical set of rules, similar to the modeling of flocks [12] [13]. The simplicity of the approach results in low computation complexity but also in high sensitivity to minor changes in the rules. Finally, [14] proposed formal methods to model the behavior at intersections. The nature of the formulation and the employed methods lead, however, to combinatorial a explosion of the states even for small

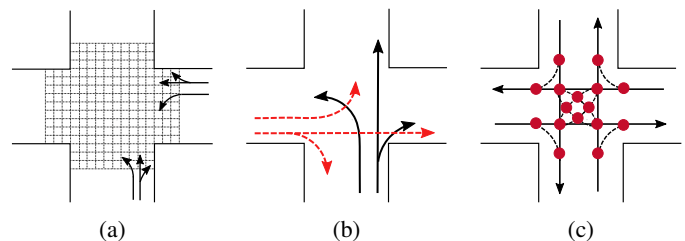


Fig. 1: Intersection modeling: (a) cells; (b) paths; and (c) collision regions/points (red circles) (adapted from [1]).

instances.

We consider an intersection as an empty space free of movement-related horizontal markings or structural restrictions, except for its boundaries. We call this space a *plaza* (see Fig. 2). The objective is to find the optimal trajectories, without predefined paths, that vehicles should follow to minimize one or more criteria, such as intersection delay or fuel consumption, while avoiding collisions. To this end, we state an Intersection Trajectories Optimal control Problem (ITOP) for CVAD. To solve the ITOP, we propose finding general functions to describe the optimal trajectories of the vehicles according to their positions and speeds, and plaza geometry.

In a previous work [15], we explored Finite Fourier Series (FFS) for the generation of trajectories (see also [16]). In this work, Bézier curves [17]–[19] are used along with discretization notions to convert the ITOP into a Non-Linear Program (NLP) with Bézier coefficients as the unknown parameters. This paper is featured by: (i) the vehicles' paths and speeds are no longer fixed; and the intersection is signal-free and movement-free. The main contributions of this paper are:

- The effective collision avoidance constraints;
- A method for the generation of the near-optimal trajectories that utilizes Bézier curves; and
- A compact representation of the Bézier curve that reduces the number of decision variables.

In Section II, the model details are provided and the ITOP is stated. In Section III, the ITOP is formulated as a NLP based on the Bézier curves representation of the states and discretization of the problem. Numerical results are presented in Section IV. Section V concludes the paper.

II. OPTIMAL CONTROL PROBLEM

In this section, the intersection plaza is modeled, the vehicles' state equations, the constraints, and the performance criterion are presented, and the ITOP is stated.

A. Plaza Modeling

A plaza can have varied layouts with respect to the number of intersecting roads and geometry. To introduce our concept, we select a four-leg intersection (Fig. 2). The x and y axes represent the central lines of the plaza on the Cartesian Coordinate System (CCS). Vehicles can travel between any two points of approaching and departing roads and their trajectories, e.g., T_1 and T_2 in the figure, are not bound to pre-specified paths or movements. We model this plaza by its four Intersection Boundaries (IB) shown by the dashed lines in Fig. 2. Each IB_h , $h = 1, \dots, 4$, is modeled by an exponential function:

$$\begin{aligned} y_h &= f_h(x(t)), \\ f_h(x(t)) &= r_{0,h} + r_{1,h} \cdot e^{r_{2,h} \cdot (x(t) + r_{3,h})}, \end{aligned} \quad (1)$$

with parameters $r_{0,h}$, $r_{1,h}$, $r_{2,h}$, and $r_{3,h}$ that shape the function according to the intersection geometry.

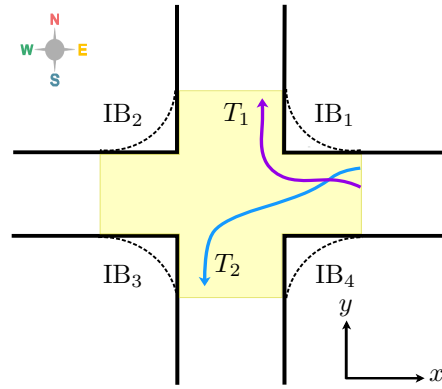


Fig. 2: A typical four-leg intersection as a plaza (colored area) with approximated boundaries (dashed lines).

B. Vehicle's Equations of Motion

There are various models to describe the vehicle's dynamics, from the simple unicycle model [20] to sophisticated car models [21]. For simplicity, in this work we use Equations of Motion (EoM) that model vehicles as point masses:

$$\begin{cases} \ddot{x}_j(t) = a_{x_j}(t), \\ \ddot{y}_j(t) = a_{y_j}(t), \end{cases} \quad (2)$$

with a_{x_j} and a_{y_j} the acceleration of vehicle j in coordinates x and y in the CCS, respectively, and t the continuous time. The total (absolute) acceleration of vehicle j is given by:

$$a_j(t) = \sqrt{a_{x_j}^2(t) + a_{y_j}^2(t)}. \quad (3)$$

The speed increment of the j -th vehicle is defined as:

$$\Delta v_j = \int_0^T a_j(t) dt, \quad (4)$$

in which T stands for the completion time, i.e., the time taken by the vehicles to cross the plaza. Then, given k vehicles at the plaza, the total speed increment is computed as:

$$\Delta v = \Delta v_1 + \dots + \Delta v_k. \quad (5)$$

C. Constraints

1) *Vehicle's kinematic constraints*: To guarantee that the vehicles speeds and accelerations are within admissible range the following constraints are defined:

$$a_j(t) \leq a_{\max}, \quad 0 \leq v_j(t) \leq v_{\max}, \quad j = 1, \dots, k, \quad (6)$$

with a_j being the total acceleration of vehicle j , v_j the speed of vehicle j , and a_{\max} and v_{\max} the maximum total acceleration and maximum speed, respectively.

2) *Vehicle-to-vehicle collision avoidance constraints*: The distance, $d_{ij}(t)$, between every two vehicles i and j must be kept at or above a minimum safe distance, d_s :

$$d_{ij}(t) \geq d_s, \quad i = 1, \dots, k, \quad j = 1, \dots, k, \quad i < j. \quad (7)$$

3) *Plaza boundaries constraints*: Violation of the plaza boundaries by the CVAD must be disallowed. Hence, the intersection's geometric constraints, based on (1), that ensure there are no collisions of CVAD with the boundaries, are:

$$\begin{cases} y_j(t) \leq f_h(x_j(t)), & \text{if } h = 1, 2 \\ y_j(t) \geq f_h(x_j(t)), & \text{if } h = 3, 4 \end{cases} \quad j = 1, \dots, k, \quad (8)$$

with $(x_j(t), y_j(t))$ being the position of vehicle j in the CCS.

D. Performance Index

We choose as the performance index a weighted sum of Δv and T that should be minimized:

$$\mathcal{J} = w_1 \cdot \Delta v + w_2 \cdot T, \quad (9)$$

with $w_1 \geq 0$ and $w_2 \geq 0$ weighting parameters. Minimizing Δv and T are conflicting objectives that affect fuel consumption and comfort (since speed increment is minimized) versus speed and maneuver completion time.

E. Intersection trajectories optimal control problem

By considering the EoM, the constraints, and the performance index, the ITOP can be formulated as:

$$\text{minimize } \mathcal{J} = w_1 \cdot \Delta v + w_2 \cdot T, \quad (10)$$

subject to (6)–(8) with $h = 1, \dots, 4$, and $0 \leq t \leq T$.

By appropriately writing the EoM in state-space representation, the ITOP formulation can be expressed as a non-linear optimal control problem [11]. Since the detailed formulation is not needed for the approach in the next section, it is omitted here. We note, however, that the states are the position and speed of each vehicle j in the CCS $(x_j(t), y_j(t), v_{x_j}(t), v_{y_j}(t))$, and the decision variables are $a_{x_j}(t), a_{y_j}(t)$, and T .

III. OPTIMIZATION PROBLEM

In this section, we propose the Bézier curve method [22] for solving the ITOP for the CVAD at the plaza. In this method, the state variables, i.e., positions and speeds, are interpolated, and control variables, i.e., accelerations and completion time, are considered in the objective function. Then, the Bézier representation of state variables is imposed on the dynamics, and the required accelerations to realize the resulting trajectories are evaluated. Finally, the ITOP is reduced to a system of algebraic equations in the Bézier coefficients and a collision-free trajectory optimization problem is formulated.

A. Bézier Approximations

The Bézier curves have several properties for trajectory optimization that are appropriate for the purpose of this work: (i) the starting and ending points of the curve correspond to the first and final Bézier coefficients, respectively; (ii) the curve completely lies within the convex hull formed by all Bézier coefficients; and (iii) the curves have the advantage of simplicity and curvature continuity.

In this part, a Bézier curve is employed to approximate each position state variable of each vehicle j in each coordinate of the CCS (x_j and y_j) as follows [22]:

$$\mathbf{z}(\tau) = \sum_{l=0}^{n_z} B_{z,l}(\tau) P_{z,l}, \quad (11)$$

with $\mathbf{z} = [x_j(\tau) \ y_j(\tau)]^T$, $0 \leq \tau = t/T \leq 1$ is the scaled time, n_z is the number of Bézier terms (order of the Bézier curve), $P_{z,j}$ are the unknown Bézier coefficients to be determined, and $B_{z,l}(\tau)$ are the Bernstein basis polynomials given by:

$$B_{z,l}(\tau) = \binom{n_z}{l} \tau^l (1-\tau)^{n_z-l}, \quad l \in \{0, 1, \dots, n_z\}. \quad (12)$$

The speed state variables of each vehicle j in each coordinate of the CCS (v_{x_j} and v_{y_j}) are the first derivatives of (11) with respect to the scaled time (τ):

$$\mathbf{z}'(\tau) = \sum_{l=0}^{n_z} B'_{z,l}(\tau) P_{z,l}, \quad (13)$$

with

$$B'_{z,l}(\tau) = \begin{cases} -n_z(1-\tau)^{n_z-1}, & \text{if } l = 0, \\ \frac{n_z! \tau^{l-1} (1-\tau)^{n_z-l}}{(l-1)!(n_z-l)!} - \frac{n_z! \tau^l (1-\tau)^{n_z-l-1}}{l!(n_z-l-1)!}, & \text{if } l \in [1, n_z-1], \\ n_z \tau^{n_z-1}, & \text{if } l = n_z. \end{cases} \quad (14)$$

1) *Boundary conditions*: For each vehicle, we know the Boundary Conditions (BCs), i.e., the initial and final positions and speeds in the coordinate system. The BCs with respect to the scaled time for each vehicle are:

$$\mathbf{z}(0) = \mathbf{z}_i, \quad \mathbf{z}(1) = \mathbf{z}_f, \quad \mathbf{z}'(0) = T\dot{\mathbf{z}}_i, \quad \mathbf{z}'(1) = T\dot{\mathbf{z}}_f. \quad (15)$$

The labels ‘i’ and ‘f’ refer to ‘initial’ and ‘final’, respectively, the prime denotes the derivative with respect to the scaled time and the dot the derivative with respect to the continuous time. These relations are obtained through the chain rule [16].

2) *Using the BCs for expressing some coefficients*: By manipulating algebraically (11)–(15), it is straightforward to derive four Bézier coefficients of (11) as:

$$\begin{aligned} P_{z,0} &= \mathbf{z}_i, & P_{z,1} &= \mathbf{z}_i + \frac{T\dot{\mathbf{z}}_i}{n_z}, \\ P_{z,n_z-1} &= \mathbf{z}_f - \frac{T\dot{\mathbf{z}}_f}{n_z}, & P_{z,n_z} &= \mathbf{z}_f, \end{aligned} \quad (16)$$

reducing the number of unknown Bézier coefficients, thus speeding up the optimization. Substituting these coefficients in (11) and organizing the resulting expression gives:

$$\mathbf{z}(\tau) = F_z + \sum_{l=2}^{n_z-2} B_{z,l}(\tau) P_{z,l}, \quad (17)$$

with

$$F_z = B_{z,0} P_{z,0} + B_{z,1} P_{z,1} + B_{z,n_z-1} P_{z,n_z-1} + B_{z,n_z} P_{z,n_z}. \quad (18)$$

The corresponding first and second derivatives with respect to the scaled time, $\mathbf{z}'(\tau)$ and $\mathbf{z}''(\tau)$, can be readily obtained.

3) *Evaluation points*: In order to solve for the unknown Bézier coefficients, the EoM are evaluated at m points, called Discretization Points (DPs). We consider m DPs with equal time intervals within the scaled time ($\tau_i - \tau_{i-1} = 1/(m - 1), i = 2, \dots, m$):

$$\tau_1 = 0 < \tau_2 < \dots < \tau_{m-1} < \tau_m = 1. \quad (19)$$

The constraints in Section II-C are satisfied only at each DP. Thus, to avoid violations in-between DPs, we must choose a large enough safe distance and/or sufficiently dense DPs.

4) *Compact matrix form representation*: Since the EoM are evaluated at the DPs, a compact matrix form representation for the position state variables and its derivatives (speed state variables and accelerations) already incorporating the coefficients from the BCs can be used. We can write the position state variables at the m DPs as vectors of its values:

$$[\mathbf{z}]_{m \times 1} = [B_{\mathbf{z}}]_{m \times (n_{\mathbf{z}}-3)} [X_{\mathbf{z}}]_{(n_{\mathbf{z}}-3) \times 1} + [F_{\mathbf{z}}]_{m \times 1}, \quad (20)$$

with $[F_{\mathbf{z}}]$ being a constant vector depending on $n_{\mathbf{z}}$ and on the BCs obtained using (18), $[B_{\mathbf{z}}]$ being a matrix of coefficients given by:

$$[B_{\mathbf{z}}]_{m \times (n_{\mathbf{z}}-3)} = [B_{\mathbf{z},2} \ B_{\mathbf{z},3} \ \dots \ B_{\mathbf{z},n_{\mathbf{z}}-2}]^T, \quad (21)$$

and $[X_{\mathbf{z}}]$ being the vector of unknown Bézier coefficients:

$$[X_{\mathbf{z}}]_{(n_{\mathbf{z}}-3) \times 1} = [P_{\mathbf{z},2} \ P_{\mathbf{z},3} \ \dots \ P_{\mathbf{z},n_{\mathbf{z}}-2}]^T. \quad (22)$$

Matrices $[B_{\mathbf{z}}]$ and $[F_{\mathbf{z}}]$ are computed offline and $[X_{\mathbf{z}}]$ results from the optimization. The compact forms of the first and second derivatives of (20) have a similar structure.

Then, the total acceleration of vehicle j along its trajectory can be represented in a compact matrix form as well by replacing (20) and its derivatives in (2) and (3):

$$[a_j]_{m \times 1} = \sqrt{\sum_{\forall \mathbf{z}} [a_{\mathbf{z}}]_{m \times 1}^2} \leq [a_{\max}]_{m \times 1}. \quad (23)$$

Note that $a_{\mathbf{z}} = \ddot{\mathbf{z}}$, therefore, we need the relations between $\ddot{\mathbf{z}}$ and \mathbf{z}'' via the chain rule (see Section III-A1).

B. Nonlinear Programming Formulation

Given the compact matrix form (20) and corresponding derivatives, we can formulate a NLP with the unknown Bézier coefficients $[X_{\mathbf{z}}]$ and the completion time T as decision variables:

$$\begin{aligned} & \min_{[X_{\mathbf{z}}] \forall \mathbf{z} \forall j, T} \mathcal{J} \\ & \text{s.t. } [a_j(t)] \leq [a_{\max}], \\ & \quad 0 \leq [v_j(t)] \leq [v_{\max}], \\ & \quad [d_{ij}(t)] \geq [d_s], \\ & \quad [y_j(t)] \leq [f_h(x_j(t))], \text{ if } h = 1, 2 \\ & \quad [y_j(t)] \geq [f_h(x_j(t))], \text{ if } h = 3, 4, \end{aligned} \quad (24)$$

with $i = 1, \dots, k, j = 1, \dots, k$, and $i < j$. We note that (20) and its derivatives are embedded in formulation (24) through the substitution in (2)–(4).

TABLE I: SETTINGS FOR VEHICLES/ INTERSECTIONS

Parameter	Description	Value
k	Number of CVAD	3
a_{\max}	Maximum acceleration (m/s^2)	2
v_{\max}	Maximum speed (m/s)	10
d_s	Safe distance (m)	1
W_{road}	Road width (m)	11
L_{road}	Road length (m)	90

C. Initialization of Decision Variables

The initialization of the unknown Bézier coefficients can be expressed in a compact form as:

$$[X_{\mathbf{z}}]_{(n_{\mathbf{z}}-3) \times 1} = \left([B_{\mathbf{z}}]_{n_{\mathbf{z}} \times (n_{\mathbf{z}}-3)} \right)^{-1} \left([\mathbf{z}_a]_{n_{\mathbf{z}} \times 1} - [F_{\mathbf{z}}]_{n_{\mathbf{z}} \times 1} \right), \quad (25)$$

with n_a being the number of DPs for the approximation and $[\mathbf{z}_a]$ the approximated position state variables. A cubic Bézier curve can be used to approximate $[\mathbf{z}_a]$ using the BCs [16][19].

The initialization of the completion time T_a can be approximated by arbitrarily selecting the time taken by a vehicle to cross in a straight direction from its origin to its destination with maximum total acceleration as:

$$T_a = \sqrt{\frac{2S}{a_{\max}}}, \quad (26)$$

where S is the distance between the origin and destination of the selected vehicle.

IV. NUMERICAL RESULTS

In this section, we evaluate the solution of the ITOP via the Bézier curve method with MATLAB 2018b. The NLP is solved using the `fmincon` solver of the optimization toolbox. Each $\Delta v_j(t)$ in (4) is computed by numerical integration of the corresponding total acceleration over time T through the built-in function `trapz`. Moreover, we compare the results obtained with the Bézier method with the approach based on the FFS [15].

A. Scenario setup

We investigate a scenario with three CVAD at the plaza. The goal is to show that the proposed Bézier method is able to generate near-optimal collision-free trajectories for these three vehicles. CVAD₁ travels from north to east, CVAD₂ travels from south to west, and CVAD₃ goes straight from east to west. The center of the plaza is the origin of the CCS. The initial positions (x_{ij}, y_{ij}) of the three CVAD are $(-2, 40)$ m, $(2, -40)$ m, and $(43, 8)$ m, respectively, and the final positions (x_{fj}, y_{fj}) are, respectively, $(45, -4)$ m, $(-45, 1)$ m, and $(-45, 8)$ m. The initial speeds (v_{ixj}, v_{iyj}) are $(1, -5)$ m/s, $(-1, 5)$ m/s, and $(-7, 0)$ m/s, respectively, and the final speeds (v_{fxj}, v_{fyj}) are, respectively, $(6, -2)$ m/s, $(-6, 2)$ m/s, and $(-8, 0)$ m/s. The required parameters for the formulation are reported in Tables I and II.

The results discussed next and summarized in Table III were obtained by the solution of the NLP (24) with $w_1 = 5, w_2 = 2, m = 30$ and $n_{\mathbf{z}} = 8$ for the Bézier method and with $w_1 = 4, w_2 = 2, m = 30$ and $n_{\mathbf{z}} = 6$ for the FFS method in [15].

TABLE II: SCALING PARAMETERS

Parameter	IB ₁	IB ₂	IB ₃	IB ₄
r_0	11	11	-11	-11
r_1	1	1	-1	-1
r_2	-1	1	1	-1
r_3	-11	11	11	-11

TABLE III: NUMERICAL RESULTS OF BÉZIER AND FFS

Method	Δv (m/s)	T (s)	\mathcal{J}	T_c (s)
Bézier	14.1	11.5	93.5	4.3
FFS	18.2	12.3	97.4	7.9

B. Quantitative Results

As shown in Table III, the computation time (T_c) of the Bézier method is lower than the one of the FFS method due to the smaller number of decision variables in the first method. Moreover, smaller total speed increment and completion time were obtained with the Bézier method.

We experimented with different combinations of values for m , w_1 , w_2 and n_z that were considered by trial and error (details not shown). Varying the values of w_1 and w_2 had more influence on the total speed increment than the completion time. When the number of DPs (m) is increased, there is an increase in computation time. Despite the corresponding increase in total speed increment, completion times also increase, suggesting that worse local minima are found for higher values of m , i.e., trajectories in longer paths result. Finally, increasing n_z increases the computation time without sensible improvements in the other measures. Small values of m and n_z may result in better values of Δv , T , and computation time. However, the trajectories might not be smooth and may also lead to infeasible instances of the NLP problem, as also observed with the FFS method [15].

C. Analysis of the Trajectories

The optimized trajectories generated by the solution of the NLP based on the Bézier and FFS methods are illustrated in Fig. 3. The colored disks corresponding to the cool colormap indicate DPs and time of the Bézier method (T_B), and the colored disks corresponding to the warm colormap indicate DPs and time of the FFS method (T_F). The gray squares are the initial positions for each CVAD. The solid thick black lines show the boundary of each IB whose approximations are presented by black dashed lines. For this particular scenario, the trajectories of both methods deviate from what would be expected in a movement-based method and it is clear that the CVAD follow free trajectories. Noteworthy, the followed paths of the Bézier method seem to approach the paths of minimum distance compared to the FFS method.

Fig. 4 shows the distances between every two CVAD generated with the Bézier method. It can be observed that, at all times, a minimum safe distance (d_s) is maintained between all CVAD. Accordingly, the vehicles distances remain above d_s by a large margin for this scenario. Similar results were observed for the FFS case (not shown) [15].

Figs. 5(a), (c), and (e) show the total (absolute) acceleration (a_j), and the acceleration (\hat{a}_j) for each of the three CVAD,

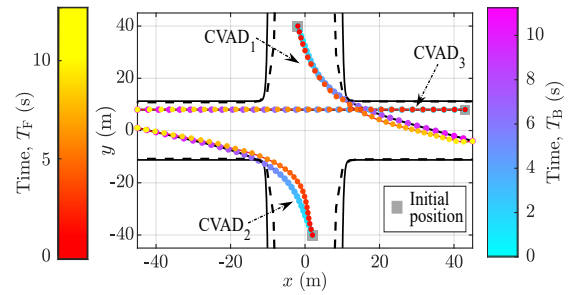


Fig. 3: Trajectories of the CVAD using Bézier method (T_B) and FFS method (T_F).

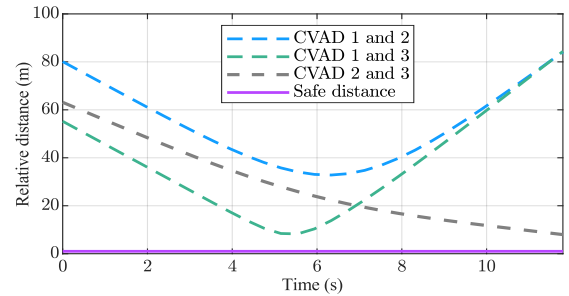


Fig. 4: Distance between every two CVAD using the Bézier method and $d_s = 1$ m.

with $j = 1, 2, 3$, for both Bézier and FFS methods. Figs. 5(b), (d), and (f) show the speed (v_j) for the same three vehicles. The total acceleration and speed profiles of each vehicle are far below the maximum total acceleration and maximum speed values, respectively, satisfying the constraints. In addition, the profiles of both methods are smooth, as expected due to the minimization of Δv . However, the acceleration and speed profiles of the trajectories generated with the FFS method exhibit more variation compared to the ones with the Bézier method. Accordingly, we can conclude that the Bézier method is capable of providing more comfortable vehicle movement with less computation time.

To evaluate the efficacy of the collision avoidance constraint, we present in Fig. 6 the numerical results of three CVAD based on a larger value of safe distance, $d_s = 7$ m. We observe in Fig. 6a a slightly different behavior of CVAD₁, which keeps more distance from the corner of the IB (compare to Fig. 3) due to the activation of collision avoidance constraints. It can be seen in Fig. 6b that the collision avoidance constraint avoids the collision between CVAD₁ and CVAD₃ at around $t = 5$ s, and subsequently, the distance between the vehicles (green dashed line) does not go below the safe distance line (solid purple line).

V. CONCLUSION

This paper introduced the ITOP with free use of the intersection space, called a plaza, by CVAD. There are no vehicle movement-related constraints except for the intersection boundaries, which were properly modeled. The Bézier curve method and discretization were used to transform the

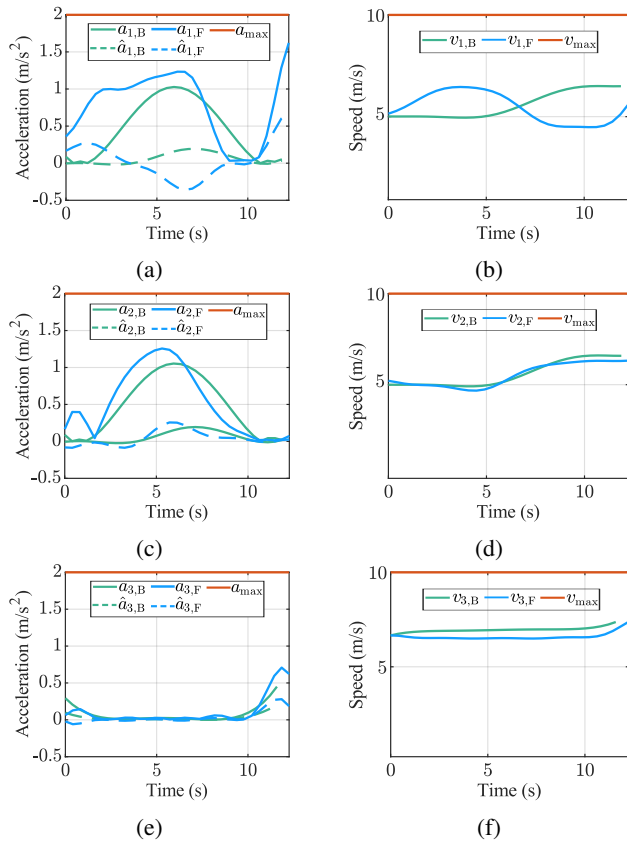


Fig. 5: (a), (c), and (e) Acceleration profiles and (b), (d), (f) speed profiles of j -th CVAD, with $j = 1, 2, 3$.

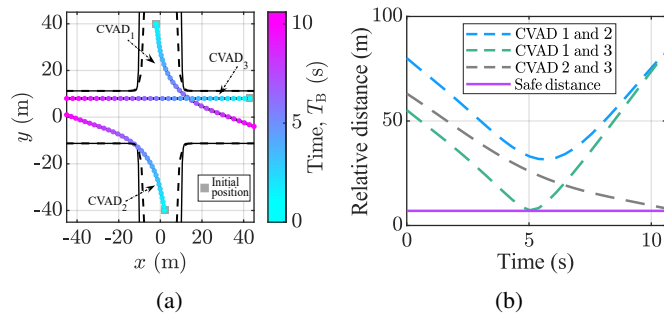


Fig. 6: (a) Trajectories of the CVAD using Bézier method; and (b) distance between them with $d_s = 7$ m.

ITOP into a nonlinear program problem. The method is able to generate near-optimal collision-free trajectories for the CVAD coordination at the plaza. Finally, the results of the proposed method were compared to the FFS method, showing slightly better results.

Future work should consider more elaborate vehicle dynamics, vehicle dimensions, different completion times for the CVAD, and free final states. Other objectives should be evaluated and additional constraints should be added, e.g., bounds on jerk values. The continuous arrival of vehicles should be handled along with a comparison with movement-based approaches based on typical traffic performance metrics,

such as traffic delay and intersection capacity.

ACKNOWLEDGMENT

We acknowledge partial financial support from the CAPES Foundation, Ministry of Education of Brazil, and the National Council for Scientific and Technological Development (CNPq).

REFERENCES

- [1] L. Chen and C. Englund, "Cooperative Intersection Management: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 570–586, 2016.
- [2] J. Rios-Torres and A. A. Malikopoulos, "A Survey on the Coordination of Connected and Automated Vehicles at Intersections and Merging at Highway On-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1066–1077, 2017.
- [3] E. R. Müller, "Optimal Arrival Time Scheduling of Automated Vehicles at Intersections," Doctoral dissertation, Post-graduate Program in Automation and Systems Engineering, Federal University of Santa Catarina, Florianópolis, SC, Brazil, 2018.
- [4] K. Dresner and P. Stone, "A Multiagent Approach to Autonomous Intersection Management," *Journal of Artificial Intelligence Research*, vol. 31, pp. 591–656, 2008.
- [5] H. Schepperle and K. Böhm, "Valuation-aware Traffic Control: The Notion and the Issues," in *Multi-agent Systems for Traffic and Transportation Engineering*. IGI Global, 2009, pp. 218–239.
- [6] J. Lee and B. Park, "Development and Evaluation of a Cooperative Vehicle Intersection Control Algorithm under the Connected Vehicles Environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 81–90, 2012.
- [7] F. Zhu and S. V. Ukkusuri, "A Linear Programming Formulation for Autonomous Intersection Control within a Dynamic Traffic Assignment and Connected Vehicle Environment," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 363–378, 2015.
- [8] E. R. Müller, R. C. Carlson, and W. Kraus, "Intersection Control for Automated Vehicles with MILP," *IFAC-PapersOnLine*, vol. 49, no. 3, pp. 37–42, 2016.
- [9] M. W. Levin and D. Rey, "Conflict-point Formulation of Intersection Control for Autonomous Vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 85, pp. 528–547, 2017.
- [10] B. Li, Y. Zhang, Y. Zhang, N. Jia, and Y. Ge, "Near-optimal Online Motion Planning of Connected and Automated Vehicles at a Signal-free and Lane-free Intersection," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1432–1437.
- [11] B. Li, Y. Zhang, N. Jia, and X. Peng, "Autonomous Intersection Management over Continuous Space: A Microscopic and Precise Solution via Computational Optimal Control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 17 071–17 076, 2020.
- [12] D. Roca, D. Nemirovsky, M. Nemirovsky, R. Milito, and M. Valero, "Emergent Behaviors in the Internet of Things: The Ultimate Ultra-large-scale System," *IEEE Micro*, vol. 36, no. 6, pp. 36–44, 2016.
- [13] D. Roca, R. Milito, M. Nemirovsky, and M. Valero, "Advances in the Hierarchical Emergent Behaviors (HEB) Approach to Autonomous Vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 12, no. 4, pp. 57–65, 2020.
- [14] G. J. Ninos Neto, "Analysis of the Management of Automated Vehicles with Free Exploration of the Intersection Area (in Portuguese)," Master thesis, Post-graduate Program in Automation and Systems Engineering, Federal University of Santa Catarina, Florianópolis, SC, Brazil, 2021.
- [15] E. Ahmadi, R. C. Carlson, W. Kraus Junior, and E. Taheri, "Near-optimal Coordination of Vehicles at an Intersection Plaza using Finite Fourier Series," in *35^o Congresso Nacional de Pesquisa e Ensino em Transportes*. Brazil: ANPET, 2021.
- [16] E. Taheri and O. Abdelkhalik, "Initial Three-dimensional Low-thrust Trajectory Design," *Advances in Space Research*, vol. 57, no. 3, pp. 889–903, 2016.
- [17] R. Lattarulo and et al., "Urban Motion Planning Framework based on n-Bézier Curves Considering Comfort and Safety," *Journal of Advanced Transportation*, vol. 2018, 2018.
- [18] M. Schwung and J. Lunze, "Networked Event-Based Collision Avoidance of Mobile Objects with Trajectory Planning based on Bézier Curves," *European Journal of Control*, vol. 58, pp. 327–339, 2021.

- [19] Z. Fan, M. Huo, S. Xu, J. Zhao, and N. Qi, "Fast Cooperative Trajectory Optimization for Close-Range Satellite Formation Using Bezier Shape-Based Method," *IEEE Access*, vol. 8, pp. 30918–30927, 2020.
- [20] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer, 2008.
- [21] A. G. Ulsoy, H. Peng, and M. Cakmakci, *Automotive Control Systems*. Cambridge: Cambridge University Press, 2012.
- [22] R. T. Farouki, *Pythagorean—hodograph Curves*. Springer, 2008.

CAN Demonstrator with Intrusion Detection System

Simulation Environment for Carrying Out and Analysing Hacker Attacks with a CAN Bus Demonstrator

Dirk Labudde and Heiko Polster

Forensic Science Investigation Lab

Hochschule Mittweida – University of Applied Sciences Mittweida, Germany
Fraunhofer Institute for Secure Information Technology, Germany

Email: dirk.labudde@hs-mittweida.de, heiko.polster@hs-mittweida.de

Abstract — The paper describes a digital simulation environment to reproduce and test attacks on Controller Area Network (CAN) bus systems with a CAN bus demonstrator. It describes how the CAN bus demonstrator is structured and used to analyze and evaluate hacker attacks on automotive bus systems. Security researchers repeatedly find vulnerabilities in various software components of networked vehicles. Since investigations on the real system seem too costly, various attacks on a CAN bus system are to be tested and analyzed with the help of a CAN bus demonstrator. The system consists of a CAN bus demonstrator in the form of a model car and the development environment Vector CANoe. The development environment is connected to the model via two NodeMCU-ESP32 development boards, which are configured as WLAN CAN gateways, and a VN1610 CAN-USB interface. The CANoe tool is used to control the CAN bus demonstrator, simulate CAN bus attacks and analyze them using the intrusion detection system. The simulation environment shows that the CAN demonstrator is an effective method for imparting theoretical knowledge and solving practical tasks in the field of car forensic training.

Keywords - cyberattacks; car forensics; can bus; demonstrator; intrusion detection system.

I. INTRODUCTION

The digital transformation has transformed the power vehicle from an isolated means of transport to a system that communicates incessantly with its environment. Thanks to the networking of vehicle functions (drive, steering, braking, comfort and infotainment) by bus systems and the communication of the vehicle with its surroundings via mobile communications, emergency call (eCall) or Car2X, vehicles today have become more efficient, more comfortable and safer. Today, a modern vehicle consists of approximately 100 million lines of program code [1], distributed over at least 50 installed Electronic Control Units (ECU). The statistics from [2] show a forecasted share of connected cars in the USA, China and the European Union up to the year 2035. Some 72% of motor vehicles in the European Union will be connected by 2035.

Reference [3] shows that by 2023 it is estimated that there will be worldwide 775 million vehicles connected via in-vehicle telematics or a smartphone app. This rapidly increasing number of connected vehicles, coupled with the constant expansion of new interfaces with the environment, opens the possibility for unauthorized persons to compromise the system of the connected vehicle.

Safety researchers repeatedly find weaknesses in the numerous software components of networked vehicles. For example, most software errors in mobile communications connections and in tracking or emergency call systems that can be detected allow infiltration into infotainment systems to subsequently compromise the control electronics. In some cases, however, it is necessary for the functionality of the comfort applications in the vehicle to send vehicle data to the infotainment system. Here, it is important to prevent the flow of data in the other direction to avoid a compromise of the control electronics [4] - [6].

This work deals with an environment to reproduce and test attacks on Controller Area Network (CAN) bus systems using the Vector CANoe tool [18] and a CAN bus demonstrator. It describes how a CAN bus demonstrator is structured and shows how the demonstrator hardware can be controlled using the Vector CANoe tool. In addition, an implementation of an intrusion detection system is shown, with which CAN bus attacks generated in the CANoe tool can be analyzed.

Due to cost reasons and the dangers of tampering with vehicle systems during real use, a CAN bus demonstrator offers a useful alternative for performing and analyzing hacking attacks on automotive bus systems. The demonstrator provides a safe way to analyze what-if scenarios.

The content of the paper is structured as follows. Section II describes the hardware components and the structure of the CAN bus demonstrator. Section III deals with the software-side control of the CAN bus demonstrator using the CANoe tool from Vector. Section IV describes the implementation of an intrusion detection system. Finally, Section V summarizes the results achieved.

II. DEVELOPMENT OF THE CAN BUS DEMONSTRATOR

The mechanical basis of the CAN bus demonstrator is Joy-It's Robot Car Kit 4WD (Figure 1), which is equipped with four electric motors [8].

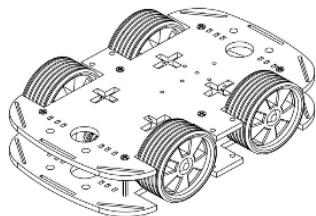


Figure 1. Schematic illustration Robot Car Kit 4WD by Joy-IT

The electronics of the CAN bus demonstrator are based on the classic basic structure of on-board automotive electronics, which consists of a central gateway and classic vehicle bus systems such as Powertrain CAN, Comfort CAN, Diagnostic CAN, etc. (Figure 2).

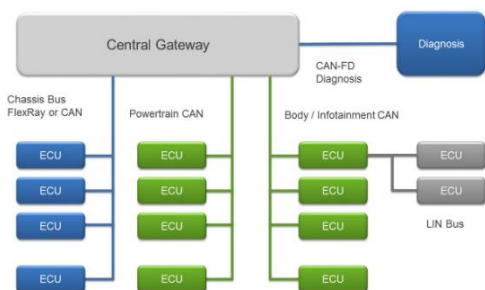


Figure 2. Basic structure of the on-board electronics of a motor vehicle - [7]

For the CAN bus demonstrator, a CAN bus system was designed (Figure 3), which includes a central gateway, a motor control unit, a control unit for lighting and a control unit for other electronic components (distance measurement, battery monitoring and horn).

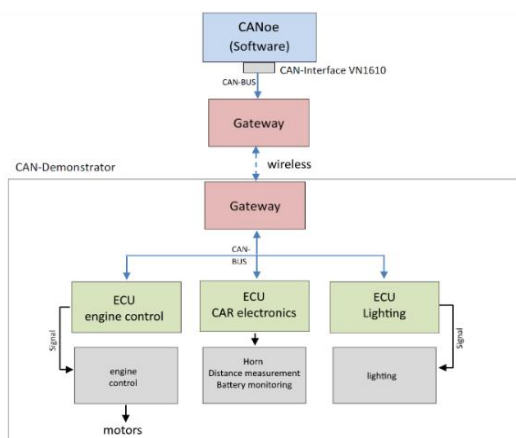


Figure 3. Block diagram of the CAN bus demonstrator with connection to the Vector CANoe tool

The CAN Bus Demonstrator is connected to the Vector CANoe tool via a wireless connection and a Vector CAN interface VN1610.

Motor electronic control unit

In addition to an Arduino Nano [9], the engine driver TB6612 [10] from Adafruit is used to control the engines (Figure 4).

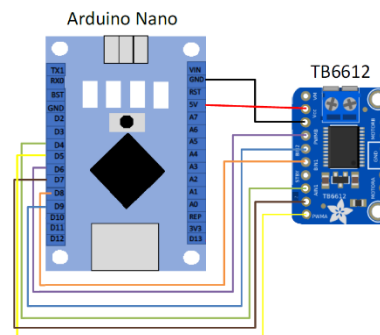


Figure 4. Sketch of the physical connection between the Arduino Nano and motor driver for motor control – adapted from [11]

This breakout board receives the corresponding control information to control the engines via a physical connection via the ATmega328P-AU implemented on the Arduino Nano. The speed regulation is realized by means of pulse width modulation.

Car electronic control unit

The demonstrator has a total of 10 Light-Emitting Diodes (LED). At the front are four LEDs, two each for the turn signals (1, 4) and headlights (2, 3) which can be adjusted in brightness by means of Pulse Width Modulation (PWM) depending on the functionality (dipped beam or main beam). Another four LEDs are located at the back. Two for the indicators (5, 10), two for the tail lamp (6, 9) and one each for the rear-end lamp (8) and the rear fog lamp (7). Equivalent to the headlights, the taillights are controlled by PWM depending on the function (dipped beam or brake beam) in brightness.

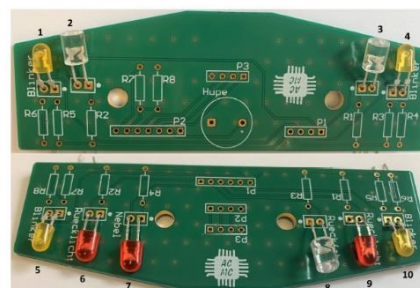


Figure 5. Overview of the implemented lighting elements

The LEDs are supplied by the corresponding Arduino Nano and are regulated via pre-resistors. Printed Circuit Boards (PCB) at the front and rear are used as a mounting solution (Figure 5).

The 2-pin speaker connector SUMMER AL-60P12 [12] is used to imitate the horn. This is regulated by the electronic control unit (Arduino Nano). Power is also supplied by the Arduino Nano.

The distance measurement is realized by means of ultrasonic sensors HC-SR04 [13]. Measurements are only taken in the direction of travel. From a distance of 100 cm to an object, CAN messages containing the distance are sent to the CAN bus at intervals of one second. If the distance decreases, the transmission interval increases. Beginning at a distance of 50 cm to the object, the message is sent every 30 seconds and at 30 cm every 200 ms. At standstill, no distance measurement takes place. The two ultrasonic sensors were soldered to the designed circuit boards at the front and rear. They belong to the electronic control unit that also supplies them with power.

To monitor the charge state of the accumulator in the CAN demonstrator, the voltage is connected to the electronics ECU (Arduino Nano) via an analog-to-digital converter. When a certain threshold is reached, a message is sent, the demonstrator starts to honk and stops. Because the maximum of the battery voltage is 8.4 V, but the Arduino Nano only has an operating voltage of 3.3 V, a voltage divider is used.

Gateway electronic control unit

The central gateway in the demonstrator is a NodeMCU-ESP32 [14]. For optimal use of the CAN, a high-speed CAN transceiver VP230 [15] is connected to the NodeMCU-ESP32. Since the VP230 cannot be conveniently attached to the NodeMCU-ESP32 and a handy solution had to be found especially for the external gateway, a compact board14 was designed for the connection of the mentioned gateway to the CAN analysis software (Figure 6), which in addition to a slot for the NodeMCU-ESP32 (also) includes an RS-232 interface Connection point according to CAN in automation Draft Standard 102 (CiA/DS 102) [16] for the connection of the CAN interface VN1610.

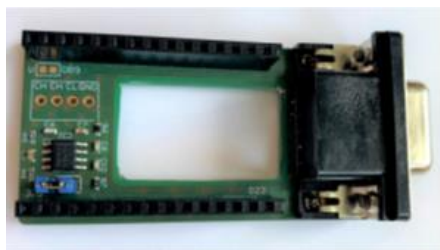


Figure 6. Circuit board of the VP230 on the client side

A small board was also designed for the transceiver in the demonstrator (Figure 7), which contains the VP230 and a three-pin terminal for CAN-High, CAN-Low and ground.



Figure 7. Circuit board of the VP230 on the server side

Power supply

For the ideal power supply, an optimal power circuit diagram was developed, considering the consumption values of the individual components and a sensible arrangement (Figure 8). Modules connected directly to the accumulator have an integrated voltage regulator.

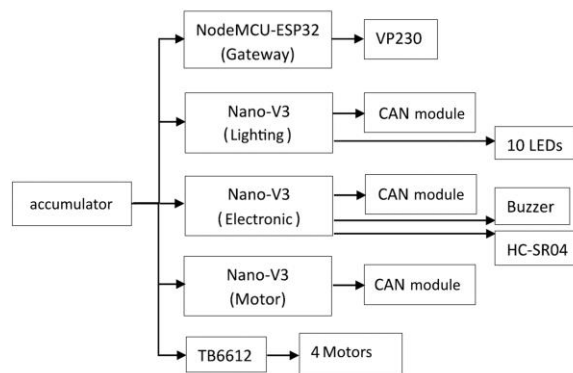


Figure 8. Overview of the power supply in the CAN demonstrator

To regulate the power supply and to be able to switch the CAN demonstrator on and off, a switch was integrated into the experimental setup.

Final assembly of the CAN bus demonstrator

In the final step, all previously described hardware components were attached to the Robot Car Kit. Figure 9 shows the final test setup.

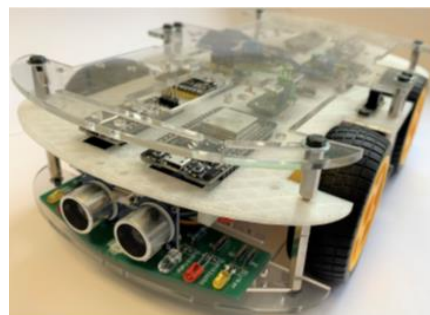


Figure 9. Final structure of the CAN demonstrator

III. CONTROL OF THE CAN BUS DEMONSTRATOR

When developing new control units for motor vehicles, it is possible to simulate CAN bus systems using professional simulation environments. The tool CANoe of the company Vector is used for this purpose. This tool allows virtual ECUs to be tested in real CAN bus environments via a hardware interface, e. g., Vector VN1610. Within the environment, all valid CAN messages for the respective CAN system are stored in a database. These messages can be sent and received in the simulation using interactive generators or virtual ECUs. The virtual ECUs are equipped with their special functionality using the Communication Access Programming Language (CAPL). This type of programming is event oriented. The control of the environment is realized with panels in which corresponding controls such as buttons, switches, visual displays etc., are to be deployed. These controls can be linked to system variables or message content. Extensive analysis tools are available within the environment.

A CANoe project was created for controlling the CAN bus demonstrator (Figure 10).

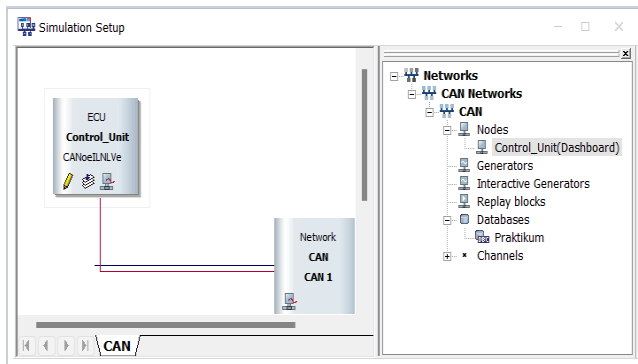


Figure 10. Simulation setup of the CANoe project

The setup of the CANoe project is limited to a virtual ECU. This is linked to a CAPL file in which the functions for the output of the CAN messages are implemented. CAPL is a procedural, C-like programming language developed by Vector. Execution is controlled by program blocks by events. CAPL programs are developed and compiled with their own browser. All objects contained in the database (messages, signals, environment variables) and system variables can be accessed. In addition, CAPL offers a variety of predefined functions. The CAN messages are sent from the virtual ECU when the dashboard changes the values of the checkboxes and controls and their associated system variables.

The basis of the CANoe project is the data base in which the CAN messages and signals required to control the CAN bus demonstrator are defined.

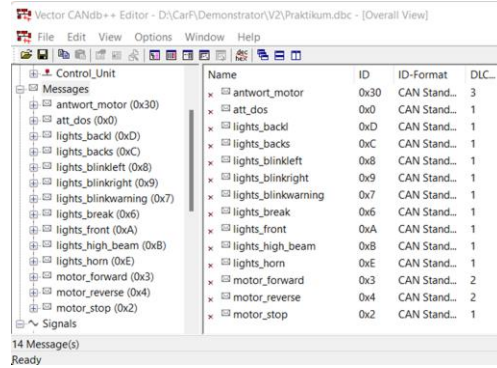


Figure 11. CAN messages of the database

Figure 11 shows the database with the 14 implemented CAN messages for the CANoe project. These messages have been assigned the necessary signals (Figure 12).

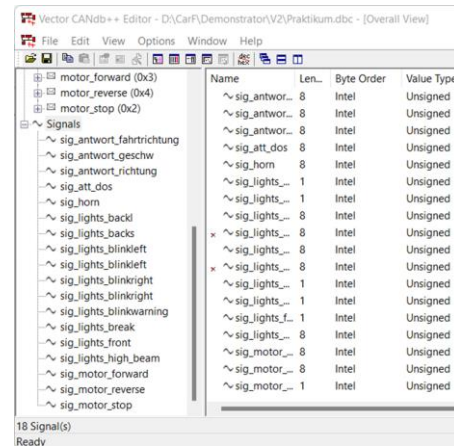


Figure 12. Signals of the Can messages

A dashboard was created using the panel designer built into the CANoe (Figure 13).



Figure 13. Dashboard for controlling the CAN bus demonstrator

Driving direction and lighting can be controlled via the checkboxes. In addition, the CAN bus demonstrator can be moved forward and backward via the virtual accelerator pedal. The speed is displayed on the dashboard in the speedometer.

IV. IMPLEMENTATION OF THE INTRUSION DETECTION SYSTEM

Intrusion detection systems are often used to detect and report an attack on a system. However, there is currently no way in vehicles to digitally trace and prove an attack on the CAN bus in digital forensics. Bresch and Salman [17] developed an intrusion detection system that can be integrated into the CANoe and therefore also be tested for the demonstrator. They propose two anomaly-based algorithms based on message cycle time analysis and plausibility analysis of messages (e. g., speed messages). Since the messages for the acceleration of the demonstrator are like those in the Bresch and Salman [17] experiment, the functions of the intrusion detection system only need to be minimally adjusted. In the case of attacks, messages are sent whose payload either contains a too high change of speed transmitted to the actual state or whose frequency deviates from that of normal operation. The Intrusion Detection System covers precisely these two scenarios. Figure 14 shows the basic program sequence of the Intrusion Detection System.

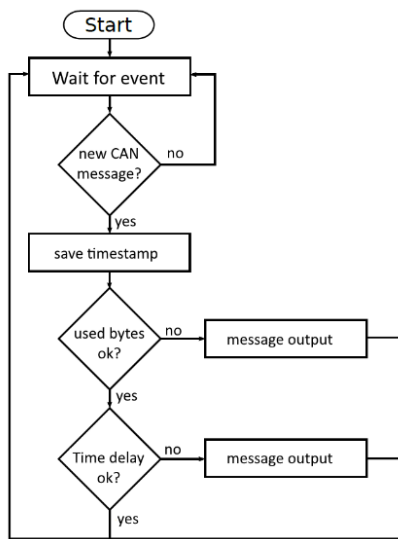


Figure 14. Sketch of the program flow of the intrusion detection system

To find out the threshold value for the message frequency, messages were sent via the panel, see Section III, to simulate an undisturbed ride. The interval between messages never fell below 300 ms/was never under 300 ms, making it a good threshold for the time between successive messages. The Intrusion Detection System is implemented in the CAPL file of the virtual ECU.

A panel was created to carry out the CAN bus attacks (Figure 15).



Figure 15. Panel for controlling the CAN bus attacks

The buttons in the panel can be used to trigger the CAN attacks Denial of Service (DoS) implemented in the CANoe project, creeping increase and decrease of speed, maximum and minimum speed, as well as the stopping of the engines.

The DoS attack is implemented in such a way that the highest priority message with the CAN identifier 0x00 is sent 20 milliseconds apart (Figure 16).

```

on sysvar att::dos{
  message 0x0 msg;
  int DoSTime = 20;
  DoSBreak = 1;
  setTimer(DoS,DoSTime);
  while(DoSBreak==1) output(msg);}
    
```

Figure 16. CAPL function of the DoS attack

The slow increase or decrease of speed is realized with a system variable and two timers (Figure 17).

The first step is to evaluate the system variable *motor_stealthy_max*. Subsequently, the direction of travel is evaluated, and a corresponding timer is started depending on it.

```

on sysvar att::motor_stealthy_max{
  activespeed = speed; int time = 80;
  if(reverse) setTimer(timer2, time);
  else setTimer(timer1, time); }
on timer timer1{
  message motor_forward msg;
  if(speed < 255){
    speed ++;
    setTimer(timer1, time); }
  msg.byte(0) = speed;
  output(msg); }
on timer timer2 {
  message motor_reverse msg;
  if(speed < 255){
    speed ++;
    setTimer(timer2, time); }
  msg.byte(0) = speed;
  output(msg); }
    
```

Figure 17. CAPL functions for realizing the creeping increase in speed

After the set 80 milliseconds have elapsed, the variable for speed and the transmission of the corresponding CAN message is increased. Furthermore, the speed is increased to the maximum value.

The IDS detect when the interval between sending the CAN message for speed change via a *TIME RATE CHANGE DETECTION* falls short and issues alerts in the CANoe tool console (Figure 18).

```
on message *{
/*TIME RATE CHANGE DETECTION, Bresch*/
possibleTimeRateAttack = 0;
zeitdiff = timeDiff(prevMessgTimeRate, this);
if (zeitdiff < cycle_time_threshold){
possibleTimeRateAttack = 1;
consecutive++;
if(consecutive > 1)write("ATTACK_TIME RATE CHANGE!!");}
```

Figure 18. CAPL function for *TIME RATE CHANGE DETECTION* according to [17]

In addition to the *TIME RATE CHANGE DETECTION*, there is a *SPEED PLAUSIBILITY DETECTION* (Figure 19).

```
/*SPEED PLAUSIBILITY DETECTION, Bresch*/
carspeed_monitor = this.byte(0) - previous_speed;
previous_speed = this.byte(0);

if (abs(carspeed_monitor) >= speed_diff_threshold){
write("ATTACK_SPEED PLAUSIBILITY!!"); }
```

Figure 19. CAPL evaluation of *SPEED PLAUSIBILITY DETECTION* according to [17]

This information is also displayed in the console of the CANoe tool.

V. CONCLUSION

In this paper, we showed that a real CAN bus demonstrator can be recreated using the hardware presented. It is possible to carry out cyber-attacks on vehicle CAN bus systems using the CANoe environment. In addition to the implementation of the CAN bus attacks, the realization of a rule-based intrusion detection system to filter out malicious messages in the data stream of the CAN bus was demonstrated. This structure was already tested by students and received positive feedback throughout. In particular, solving practical tasks on the real CAN bus demonstrator was positively mentioned and motivated the participants to find creative solutions. In the future, it is conceivable to equip the demonstrator with further functionalities to realize further attack vectors on automotive systems.

REFERENCES

[1] Y. Simon, "100 million lines of code: What the Golf 8 can do" [Online]. Available from: <https://www.kfz-betrieb.vogel.de/100-millionen-codezeilen-das-kann-der-golf-8-a-824084/> 2022.05.19

[2] Statista, "Connected cars as a share of the total car parc in China in 2021, with a forecast for 2025, 2030, and 2035" [Online]. Available from: <https://www.statista.com/statistics/1276410/connected-car-fleet-as-a-share-of-total-car-parc-in-china/> 2022.05.19

[3] S. Smith, "In-Vehicle Commerce Opportunities Drive Total Connected Cars to Exceed 775 Million by 2023" [Online]. Available from: <https://www.juniperresearch.com/press/in-vehicle-commerce-opportunities-exceed-775mn> 2022.05.19

[4] K. Koscher et al., "Experimental Security Analysis of a Modern Automobile" [Online]. Available from: <https://ieeexplore.ieee.org/document/5504804> 2022.05.19

[5] K. Kochetkova, "Shock at the wheel: your Jeep can be hacked while driving down the road" [Online]. Available from: <https://www.kaspersky.com/blog/remote-car-hack/9395/> 2022.05.19

[6] C. Thompson, "A hacker made a \$30 gadget that can unlock many cars that have keyless entry" [Online]. Available from: <https://www.businessinsider.com/samy-kamkar-keyless-entry-car-hack-2015-8> 2022.05.19

[7] R. Lieder, "Gateway processor evolution in automotive networks" [Online]. Available from: https://www.cancia.org/fileadmin/resources/documents/conferences/2017_lieder.pdf 2022.05.19

[8] Robot Car Kit 4WD, [Online]. Available from: <https://joy-it.net/en/products/Robot03> 2022.05.19

[9] Arduino Nano, [Online]. Available from: <https://store.arduino.cc/products/arduino-nano> 2022.05.19

[10] Adafruit TB6612 1.2AMotor Driver Breakout Board, [Online]. Available from: <https://www.exp-tech.de/en/modules/motor-controllers/dc-motors/6501/adafruit-tb6612-1.2amotor-driver-breakout-board> 2022.05.19

[11] TB6612 1.2A DC/Stepper Motor Driver, [Online]. Available from: http://wiki.t-o-f.info/uploads/Arduino/Adafruit_TB6612_1motor_bb.png 2022.05.19

[12] DISTRELEC GROUP AG: RND Components Electromagnetic Buzzer, [Online]. Available from: https://www.distrelec.de/Web/Downloads/_t/ds/RND%20430-00009_eng_tds.pdf 2022.05.19

[13] JOY-IT: Joy-IT Ultrasonic Distance Sensor, [Online]. Available from: <https://cdn-reichel.de/documents/datenblatt/A300/SEN-US01-DATASHEET.pdf> 2022.05.19

[14] JOY-IT: NodeMCU ESP32, [Online]. Available from: <https://joy-it.net/en/products/SBC-NodeMCU-ESP32> 2022.05.19

[15] SN65HVD23x 3.3-V CAN Bus Transceivers, [Online]. Available from: https://www.ti.com/lit/ds/symlink/sn65hvd230.pdf?ts=1646221051596&ref_url=https%253A%252F%2022.05.19

[16] CiA Draft Standard 102 Version 2.0, [Online]. Available from: <http://affon.narod.ru/CAN/DS102.pdf> 2022.05.19

[17] M. Bresch and N. Salman, "Design and implementation of an intrusion detection system (IDS) for in-vehicle networks" Department of Computer Science and Engineering (Chalmers), Chalmers University of Technology, Göteborg, 2017, [Online]. Available from: <https://publications.lib.chalmers.se/records/fulltext/251871/251871.pdf> 2022.05.19

[18] Vector, "Testing ECUs and Networks With CANoe" [Online]. Available from: <https://www.vector.com/int/en/products/products-a-z/software/canoe/> 2022.05.19

A Smart Control Strategy for a Battery Thermal Management System: Design, Validation and Implementation

Mikel Arrinda, Gorka Vertiz
CIDETEC, Basque Research and Technology Alliance
(BRTA)
Donostia, Spain
e-mail: marrinda@cidetec.es; gvertiz@cidetec.es

Christophe Morel, Nicolas Hascoët
Commissariat à l'énergie atomique et aux énergies
alternatives (CEA)
Grenoble, France
e-mail: christophe.morel@cea.fr; nicolas.hascoet@cea.fr

Pierre Woltmann
AUDI AG
Ingolstadt, Germany
e-mail: Pierre.Woltmann@audi.de

Alois Sonnleitner
Miba eMobility GmbH
Laakirchen, Austria
e-mail: Alois.Sonnleitner@MIBA.COM

Abstract— A Battery Thermal Management System (BTMS) controller with smart features is designed, validated through simulations, and implemented at lab level. The bedrock of the developed controller consists of four Proportional-Integral-Derivative (PID) controllers that manage independently the four actuators of the evaluated thermal system. The additional smart features respond to four different control goals: driving profile adaptation, energy consumption minimization, aging minimization, and fast-charge efficiency maximization. The designed controller's stability and applicability is validated through simulations thanks to the provided battery pack model, heat exchanger model and thermal actuators models. Finally, the controller has been implemented on the battery pack of the AUDI e-tron at lab level. The results confirm that the designed controller is suitable to be used on a real vehicle.

Keywords— control; battery thermal management system; electric vehicle; experimental.

I. INTRODUCTION

Decarbonization and emission reduction of road transport are the main drivers for the electrification of vehicles. The envisaged European CO₂ fleet emission targets for 2025 [1] and 2030 already require a massive market introduction of partially electrified vehicles or full electric vehicles. Furthermore, local or regional air quality regulations, such as potential zero emission zones, will drive demand for these vehicles. However, there are still some obstacles for user acceptance of those electric vehicles [2]: high cost, slow charging, limited range, unperceived added value and concerns of limited mobility. In order to remove these obstacles, it is necessary to increase the efficiency of the components of the battery pack system.

One of the most critical components in an electric vehicle is the Battery pack Thermal Management System (BTMS). A BTMS is the component that adjusts and balances the battery temperature to an appropriate range [3]. There are different BTMS-related technologies, but basically all the technologies can be categorized into two topics: BTMS design and BTMS control strategies.

The BTMS control strategy topic is addressed in this paper. The BTMS control strategies are the algorithms that manage the thermal actuators of the BTMS. They aim at controlling the temperature range and distribution inside the battery pack [3]. These control strategies have a direct effect on the performance rate of the thermal system [4]. There are many techniques available, but the most common and robust ones are the Proportional-Integral-Derivative (PID) and the state diagram control strategy.

The PID control is a control loop mechanism employing feedback that applies a correction based on proportional, integral, and derivative term of the difference between the desired set point and the measured one. The PID is widely used in industrial applications, and it is a suitable algorithm to control the BTMS [5]. It provides a continuous control based on the observed changes in the controlled variable and can bring the system to a stable state. However, this kind of control is far from being an optimal control method for the BTMS since it cannot be adjusted to different scenarios and/or objectives [6].

The state diagram control strategy is an expert control that defines the decision-making responses based on the system's state diagram [7]. These states, which have been defined based on the observations of the real system, define the way the actuators are managed. This type of controller is also called macro-control or rule-based control. It is easily implementable and provides high rate of adaptability to a variety of scenarios and can respond to different objectives. Nonetheless, it does not pay attention to the dynamics of the controller variable, and it is uncommon to bring the system to a stable state [8].

Consequently, this paper proposes a controller that brings the controlled variable to a stable state while being adaptable to different scenarios and objectives. The main element of the controller is the PID, that can reach a stable state. Above it, smart features have been added based on state diagram concepts. These smart features have the goal of improving the efficiency of the heating and cooling system of the battery pack.

The aim of the paper is to design, validate and implement at lab level this smart control that (0) has a continuous control that can bring the system to a stable state; (1) lengthens the life of the vehicle by minimizing the aging and adjusting the control strategy based on the driving profile; and (2) lengthens the use range by minimizing the total energy consumption and maximizing the charge events. This paper is structured as follows: the use case is introduced in Section II. The design is described in detail in Section III. The simulations used to validate the designed controller are explained in Section IV. The test at lab level is shown in Section V. The discussion is done in Section VI. The conclusions are drawn in Section VII.

II. USE CASE

The developed control strategy is designed for a commercial electric vehicle. The evaluated battery pack is the one inside the 2019 AUDI e-tron, see Figure 1.a. However, the thermal system used on the use case is not the commercial one, instead, it is the one developed in i-HeCoBatt European project, grant agreement No. 824300 [9].

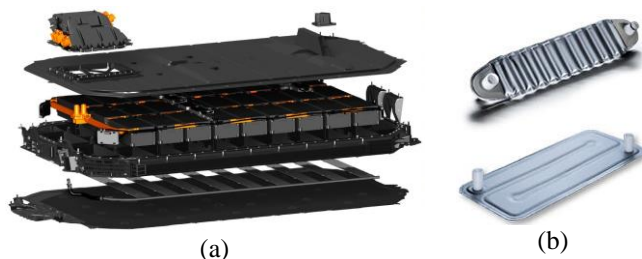


Figure 1. Evaluated use case: (a) battery pack [10]; (b) FLEXcooler [11].

The heat exchanger has been built using MIBA's FLEXcooler® technology, see Figure 1.b. Additionally, the heating process is done by resistances that are in direct contact with the batteries instead of heating the liquid of the heat exchanger.

III. DESIGN

The designed TMS strategy is a mix of PID controllers and state diagram controller's concepts. The result is a smart control strategy that brings the system to a stable state while adapting smartly to different scenarios and objectives, see Figure 3.

Firstly, three main states are defined following the state diagram controller design: cool zone, optimal operation zone and hot zone. These zones represent the actuation temperature ranges of the designed controller. At the cool zone, heating processes will be taken. At the hot zone, cooling processes will be taken. Finally, at the optimal operation zone, there will not be any active control. These control zones are divided by two temperature thresholds. The hot zone's temperature threshold is set up to 35°C. The cool zone's temperature threshold is set up to 10°C.

The base of the designed controller that provides the active control is the PID controller. Specifically, there are 4 PIDs. Three of the PIDs are used to control independently

the three independent heating resistance areas that are in direct contact with the batteries. The fourth PID controls the chiller used to cool down the liquid that goes through the heat exchanger. The PIDs are saturated to avoid an unnecessary overlapping of processes. An anti-windup function is added to the integral component of all the PID controllers to avoid undesired responses of the controller.

The different PIDs have been parametrized using Ziegler-Nichols method [12]. The method consists of getting the transfer function of the system under a step perturbation, see Figure 2.

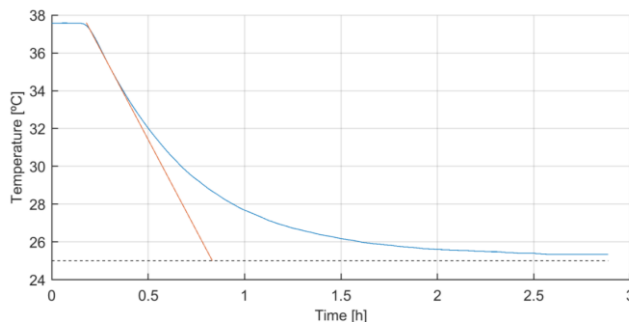


Figure 2. Applied Ziegler-Nichols method (red line) for the heat exchanger PID to the system response under a step perturbation (blue line).

The parameters of that transfer function are used to calculate the parameters of the PID controller, see Table I.

TABLE I. PID TUNING VALUES.

Parameter	Equation	Heat Exchanger A-sample
P (Kp)	$1,2 \times T / (L \times yf)$	10.5945
I (ki)	$Kp / (2 \times L)$	0.0227
D (kd)	$Kp \times 0,5 \times L$	1238.3
N (td)	>100	105

Once we build up the basis of the controller (the main state diagram with the parametrized four PID controllers), additional states are defined. These states provide the smart features of the controller and fulfil the objectives of the controller.

Firstly, the states used to lengthen the life of the vehicle by minimizing the aging and adjusting the control strategy based on the driving profile are implemented. The obtained features on those states are the following:

- Optional sport mode. The aggressivity of the controller is modified at the driver's will. This is done by increasing the P value of the PID controller. For this work, the P value is increased 10 times if an aggressive driving is requested. The adjustment of the controller to the driver characteristics decreases the aging.
- Preconditioning when parked. The heating when parked is set to the state of having the contact switched on. The controller is informed via a Boolean signal. The signal can come from different sources, such as mobile phones or PCs. It reduces the aging that comes from cold or hot starts (depending on the room temperature).

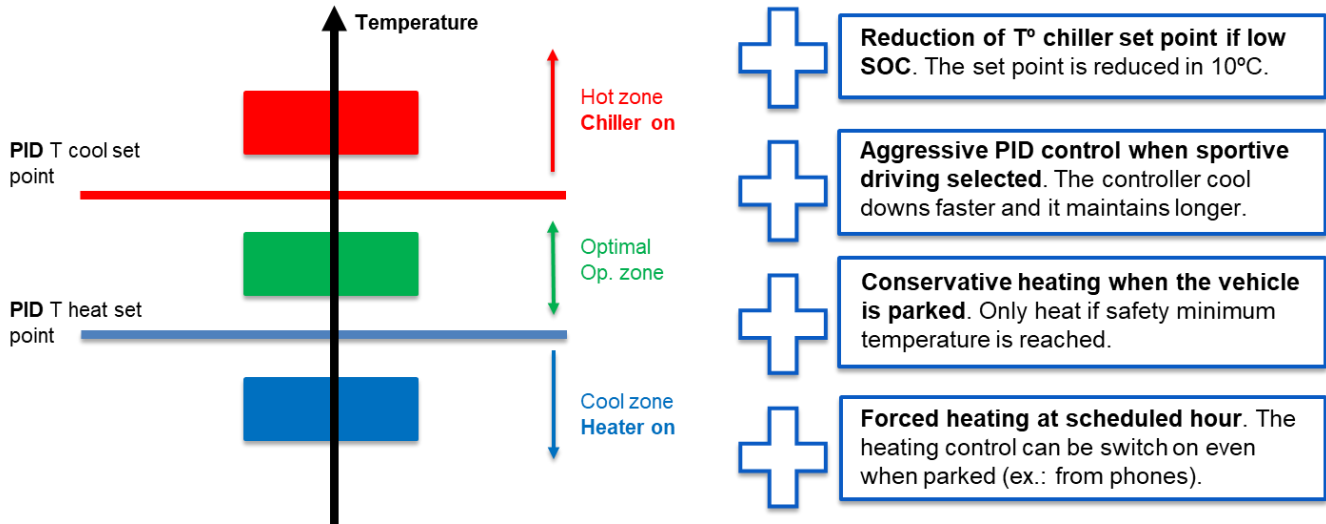


Figure 3. Design of the adaptive and smart control strategy.

Secondly, the features used to lengthen the use range minimizing the total energy consumption and maximizing the charge events are added. These features are the following ones:

- Conservative cooling-heating when parked (the contact of the vehicle is switched off). The temperature thresholds of the controller are modified to increase the optimal operation zone. The thresholds are modified to 45°C and to -20°C. Unnecessary energy consumption is avoided when parked.
- Reduction of temperature set point if conditions are met. The detection of an imminent charge event leads to act accordingly. The controller is informed via a Boolean signal. The detection can come from different sources, such as GPS signals or machine learning predictive algorithms. The upper temperature threshold is reduced to 15°C. The efficiency of charge events is maximized thanks to reducing the limitations imposed by the temperature.

IV. VALIDATION

The validation process of the designed controller is undergone through simulations using a modeling library containing the battery pack model, heat exchanger model and the model of the thermal actuators provided in i-HeCoBatt project “unpublished” [13], see Figure 6.

The simulation environment is built up to have a current profile, the room temperature, the contact state, the driving profile selection, the preconditioning signal, and the sudden fast-charge event signal as inputs and have the thermal response of the battery pack, the thermal response of the liquid of the heat exchanger, the response of the controller and the energy consumption as outputs.

This simulation environment is used to validate the different features of the controller through 4 simulations:

- The first simulation is used to validate the overall stability of the controller, see Figure 6.a. For that,

the whole battery pack model (heat exchanger and actuators included) is simulated at a standard driving mode with a mild vehicle driving cycle that has a fast-charge event at the end of the cycle, see current profile in Figure 3. The contact is assumed to be all the time on. The room temperature is set to be constant at 35°C. The additional signals are set off.

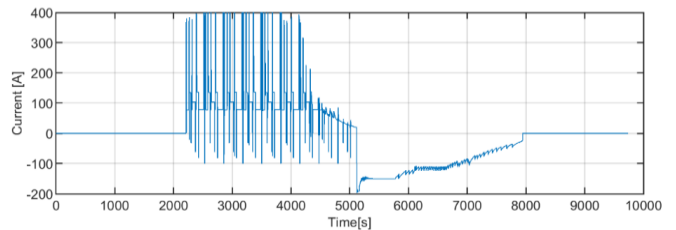


Figure 4. Realistic drive cycle current profile.

- The second simulation is used to validate the overall stability of the Sport mode used for aggressive drivers, see Figure 6.b. The inputs of the simulation are the same as in the first simulation except for the sport mode that is switched on.
- The third simulation is used to validate the efficacy of the added feature that maximizes the charge event, see Figure 6.c. The inputs are the same as in the second simulation except for the sudden fast-charge signal, see Figure 5.

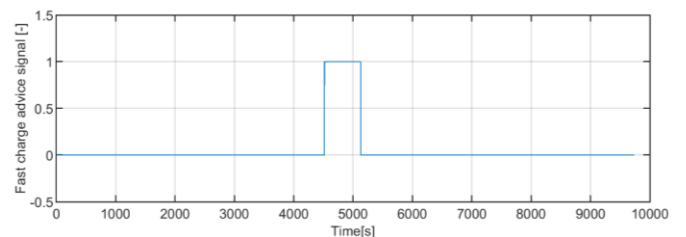


Figure 5. Sudden fast-charge event signal used in the third simulation.

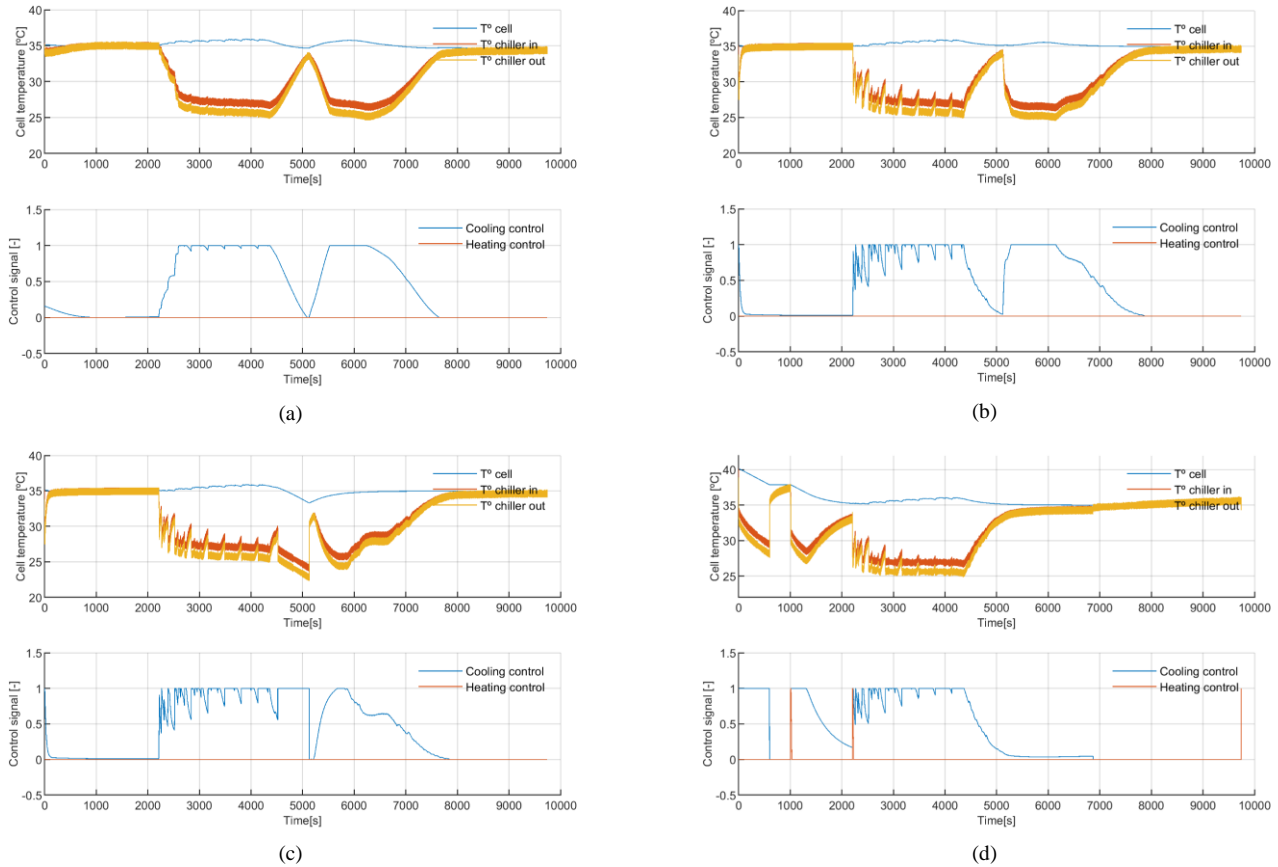


Figure 6. Results from (a) the first simulation, (b) the second simulation, (c) the third simulation and (d) the fourth simulation. There are two figures for each simulation: the upper figure is the thermal response of the battery and the coolant temperature; the lower figure is the response of the controller.

- The fourth simulation validates the preconditioning features, see Figure 6.d. The inputs used in the second simulation are used in the fourth simulation, but with some modifications. The fast charge is eliminated from the current profile, the room temperature is increased to 40°C, the contact signal is switched off in some ranges and the preconditioning signal is switched on 1h before the start of the driving cycle. The modified preconditioning and the contact signals are shown in Figure 7.

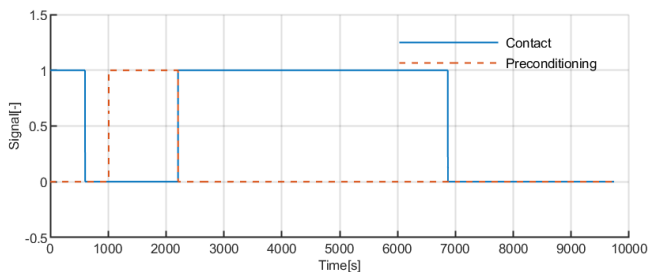


Figure 7. Contact and preconditioning signals in the fourth simulation.

V. IMPLEMENTATION

A test at lab level has been run to confirm the applicability of the designed and validated controller.

The set-up has been done in CEA’s laboratory, see Figure 8.a. The battery pack is positioned inside a climatic chamber where the heat exchanger is connected to an external thermal regulator to ensure the cooling and heating strategy during the test. The BP wiring box is connected to the electric regulation devices of the test bench (Current range: +/- 0-800 A and Voltage range: 60-1000V). In order to avoid convection heat transfer on the lower face of the BP, a plasterboard is added between the battery pack and the metallic frame that supports the battery pack.

Pressure and temperature sensors are positioned on the inlet and the outlet pipes of the heat exchanger. We have added 24 external temperature sensors on the battery pack surfaces. A small extra pressure drop is created by the non-rigid 0.5m pipe to ensure connection between the external chiller pipes and the heat exchanger. The pressure sensors are from Danfoss (Type MBS 1700) with precision of 0.5% of full scale (0-8 bars). The temperature sensors are PT100 that were calibrated from 0 to 100°C.

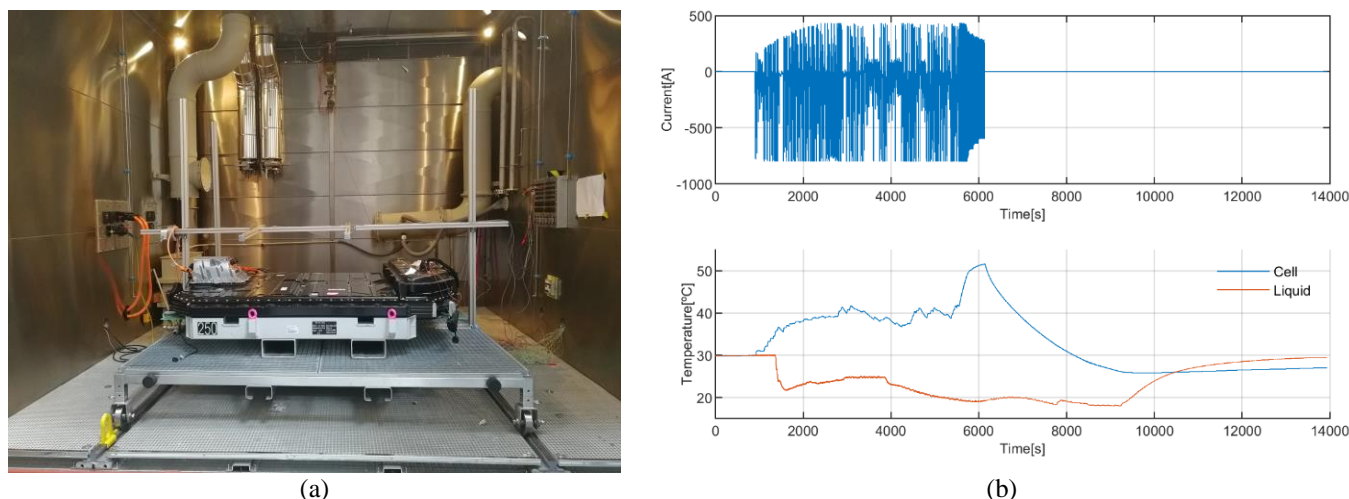


Figure 8. Tests at lab level of the developed controller. (a) the test bench with the tested battery pack; (b) results of the test at lab level: upper graph the applied current profile, the lower graph the thermal response of the battery and the coolant.

The inputs on the test at lab level are the following ones:

- The current profile of an exigent driving cycle, see upper graph of Figure 8.b.
- The room temperature is set up to a constant 30°C through a climatic chamber.
- The driving adjustment is done to a standard-conservative driving profile (sport mode off).
- The contact is assumed to be active all the test (contact switched on).
- The preconditioning is not applied (preconditioning signal switched off).

There is no fast-charge event and therefore, the signal that announces the imminence of this event is not applied (fast-charge event signal switched off).

VI. DISCUSSION

The results of the simulation have successfully validated the designed control strategy. The first simulation shows how the cooling process is activated once the battery surpasses a certain temperature and how the controller is able to adjust the actuator in a continuous and smooth way, providing at the end a stable state.

The second simulation shows how the sport mode has a faster reaction to the temperature changes of the system, providing a faster response of the controller. The controller is able to adjust its response to a more aggressive driving pattern, hence minimizing the effect this aggressive driving pattern has in the aging of the battery.

The third simulation shows how the reduction of the upper temperature threshold through the fast-charge event notifying signal reduces considerably the temperature of the battery before that event (the battery temperature is reduced 3°C), reaching lower temperatures at the peak of the fast-charge event (1°C lower).

The fourth simulation shows that the preconditioning and energy saving features work properly.

In conclusion, the performance of the controller itself as well as the added smart features have been validated through simulations.

Lastly, the validated controller has been applied in a real battery pack with a demanding current profile. The results show how the controller starts working once the optimal operation zone is left. We can see how the controller is able to provide an appropriate smooth control even though the temperatures the battery reach are much higher than the ones observed in the simulations. This means that the controller is suitable to be used in real applications.

VII. CONCLUSIONS

This paper proposes a design of a smart BTMS control strategy that has a continuous and smooth control over the actuators while (1) lengthening the life of the vehicle by minimizing the aging and adjusting the control strategy based on the driving profile and (2) lengthening the use range by minimizing the total energy consumption and maximizing the charge events.

All the smart features of the controller have been validated through simulations and the stability and the applicability of the proposal have been observed through simulations and through a real test of AUDI's battery pack at lab level.

To sum up, the design is simple and implementable; the validation has shown its viability; and the tests done at lab level have shown its implementability.

ACKNOWLEDGMENT

This work was supported by the European Union's Horizon 2020 Programme for research and innovation under Grant Agreement No. 824300, i-HeCoBatt project [9].



REFERENCES

- [1] P. Mock, "CO2 emission standards for passenger cars and light-commercial vehicles in the European Union," 2018. [Online]. Available: [http://www.europarl.europa.eu/news/en/press-room/20181218IPR22101/curbing-co2-\(accessed Mar. 24, 2022\)](http://www.europarl.europa.eu/news/en/press-room/20181218IPR22101/curbing-co2-(accessed Mar. 24, 2022))
- [2] ERTRAC, EPoSS, and ETIP SNET, "European Roadmap Electrification of Road Transport Status: final for publication," 2017.
- [3] H. Dai et al., "Advanced battery management strategies for a sustainable energy future: Multilayer design concepts and research trends," *Renewable and Sustainable Energy Reviews*, vol. 138, no. August, p. 110480, 2021, doi: 10.1016/j.rser.2020.110480.
- [4] H. Liu, M. Wen, H. Yang, Z. Yue, and M. Yao, "A Review of Thermal Management System and Control Strategy for Automotive Engines," *Journal of Energy Engineering*, vol. 147, no. 2, p. 03121001, Apr. 2021, doi: 10.1061/(asce)ey.1943-7897.0000743.
- [5] J. Cen and F. Jiang, "Li-ion power battery temperature control by a battery thermal management and vehicle cabin air conditioning integrated system," *Energy for Sustainable Development*, vol. 57, pp. 141–148, 2020, doi: 10.1016/j.esd.2020.06.004.
- [6] H. Liu, M. Wen, H. Yang, Z. Yue, and M. Yao, "A Review of Thermal Management System and Control Strategy for Automotive Engines," *Journal of Energy Engineering*, vol. 147, no. 2, p. 03121001, Apr. 2021, doi: 10.1061/(asce)ey.1943-7897.0000743.
- [7] S. Yang et al. "A review of lithium-ion battery thermal management system strategies and the evaluate criteria," *International Journal of Electrochemical Science*, vol. 14, no. 7, pp. 6077–6107, 2019, doi: 10.20964/2019.07.06.
- [8] M. Arrinda, G. Vertiz, D. Snachez, A. Makibar and H. Macicior, "Surrogate Model of the Optimum Global Battery Pack Thermal Management System Control," *Energies*. Vol 15(5), no 1695, 2022, doi: 10.3390/en15051695
- [9] CIDETEC, "Home - i-HeCoBatt." <https://ihcobatt.eu/> (accessed Feb. 10, 2022).
- [10] "Company | audi.com." <https://www.audi.com/en/company.html> (accessed Feb. 22, 2022).
- [11] "Miba: Battery Components." <https://www.miba.com/en/innovation/battery-components> (accessed Feb. 21, 2022).
- [12] D. Valério and J. S. da Costa, "Tuning of fractional PID controllers with Ziegler–Nichols-type rules," *Signal Processing*, vol. 86, no. 10, pp. 2771–2784, Oct. 2006, doi: 10.1016/J.SIGPRO.2006.02.020.
- [13] C. Morel et al., "Numerical study of two Heat Exchangers for the cooling of a Battery Pack for an Electric Vehicle," *Open Research Europe*, 2022.

Event-triggered Robust Output Feedback Controller for a Networked Roll Control System

Fernando Viadero-Monasterio*, M. Jiménez-Salas†, M. Meléndez-Useros‡,
J. Garcia-Guzman§, B. L. Boada¶ and M. J. L. Boada||

*†‡§¶Mechanical Engineering Department

*†‡¶||Institute for Automotive Vehicle Safety

§Computer Science and Software Engineering Department

Universidad Carlos III de Madrid, Leganés, Spain

Email: * fviadero@ing.uc3m.es, † manuejim@ing.uc3m.es, ‡ mmelende@ing.uc3m.es,

§ jgarciag@inf.uc3m.es, ¶ bboada@ing.uc3m.es, || mjboada@ing.uc3m.es

Abstract—Development of control systems that help to reduce the danger in risky driving situations while also improving stability and comfort are one of the main research focus nowadays. The use of low-cost sensors integrated in these systems is important for their commercialisation. Also, more complex architectures require a larger number of sensors and connections. The delays generated by these sensors, in addition to those of the network, must be addressed to provide robust control. The novelty of this paper is a robust H_∞ Output Feedback Controller based on an active suspension system for roll control that takes into account the effect of the delays in the network. An event-triggering condition is included in order to not saturate the network.

Keywords— H_∞ control; event-triggering; networked control systems; roll stability; active suspension

I. INTRODUCTION

The development of advanced vehicle safety systems is a major research focus in the road transport sector today. These systems can be divided into active and passive systems. Passive systems attempt to minimise the damage that vehicle occupants may suffer in an accident from a mitigating perspective. Active systems act to prevent a potential accident by taking control of vehicle dynamics. The latter requires sensors to monitor the dynamic state of the vehicle, as well as controllers that provide a fast and reliable response.

One of the main causes of fatal accidents in road transport is rollover. Danger of rollover becomes even greater in vehicles with a higher centre of gravity, such as SUVs, vans or trucks. For this reason, previous work has focused on designing rollover controllers to achieve a certain vehicle behaviour. Vehicle rollover stability can be improved through different control strategies, such as four wheel steering, differential braking, active suspension, or active stabilizer. In [1], a robust rollover risk suppression controller is proposed by reducing lateral acceleration through a steer-by-wire system. In [2], an integrated controller based on active steering and braking is proposed, taking the expected path of the driver to achieve better anti-rollover effect into account. A differential braking rollover mitigation control strategy through a polytopic description of the vehicle is presented in [3]. In [4], an H_∞ controller of roll stability is proposed, taking into account the effect of delays and noise of low cost sensors. Overall, the

forementioned studies were good, however a control signal was generated even when not necessary, as no rules were defined in order not to update it when the system does not change substantially.

Accurate knowledge of the variables involved in rollover dynamics is key to designing reliable controllers that help improve vehicle stability and save lives. Some of them, such as roll angle and sideslip angle, cannot be measured directly by sensors installed in series production vehicles due to their high cost. Previous work analysed the use of low-cost sensors to estimate roll angle using different methods [5] [6] which, while demonstrating good performance, could present relevant noise and delays related to the low-cost nature of the architecture.

In real control systems, the plant, controller, sensors and actuators are usually located far from each other. Therefore, signals must be sent through a communication network. Networked Control Systems (NCSs) bear with delays generated in transmission process which affect the system. In autonomous or intelligent vehicles, the architecture network can increase its complexity significantly and higher bandwidth is required in order to not saturate the network. This leads to an increment of the delay between sensors and controllers which could lead to an unstable system behaviour. Therefore, an event-triggering condition can be included in the controller design so that the amount of data transmitted over time is reduced. This condition, which is evaluated at every sampled time step, determines if the state of the system has changed significantly enough. Only in that case, the observed variables from the sensors are transmitted through the network. The event-triggering method has proven to be an useful way to avoid saturating the network by not sending data packets through it unless is strictly necessary [7] [8].

In order to compensate the delays in the network and the noise in sensor signals, a robust control design is required to improve the performance of the system. Robust LMI-based H_∞ control is a good option for this kind of problems, as proven in previous works [9] [10].

The novelty of this paper is the development of a robust H_∞ Output Feedback Controller for a NCSs that takes into account the effect of the delays over the network. An event-triggering

condition mechanism is included in order to not saturate the network. Lyapunov-Krasovskii functional approach and LMI restrictions are used to guarantee system stability and H_∞ criteria. The controller is based on an active suspension system, taking the anti-roll moment as control input and the roll rate as measured data.

The document is structured as follows. Section II presents the mathematical formulation of the problem, introducing the vehicle model used for the design of the controller and the control algorithm. Section III includes the simulation results where the proposed system is compared with the vehicle without control. Section IV states the conclusion of this study.

II. PROBLEM FORMULATION

This section describes the problem of event-triggered H_∞ Output Feedback Controller for roll stability. The diagram of the network communication and control sequence is depicted in Figure 1. Every component is described in the following subsections.

A. Vehicle model

The vehicle model used in the design of the H_∞ Output Feedback Controller describes the roll vehicle motion as seen in Figure 2, where ϕ is the roll angle and a_y is the lateral acceleration.

$$\dot{x}(t) = Ax(t) + B_u \tilde{u}(t) + B_{a_y} a_y(t) + B_{\phi_r} \phi_r(t) + B_d d_s(t) \quad (1)$$

$$y(t) = C_1 x(t) \quad (2)$$

$$z(t) = C_2 x(t) \quad (3)$$

with $x = [\phi, \dot{\phi}]^T$ as the state vector defined by the roll angle ϕ and roll rate $\dot{\phi}$, $y(t) = \phi$ is the observed measurement from the system, $z(t)$ is the control output to be minimized though the event-triggered control input $u = M_x$, where M_x is the anti-roll moment provided by actuators located at every wheel suspension. a_y is the lateral acceleration of the vehicle, ϕ_r is the road bank angle and d_s is the unknown vector disturbance. a_y , ϕ_r and d_s are an input disturbance to the system, which can be represented as $\omega(t) = [a_y(t) \ \phi_r(t) \ d_s(t)]$, with an input

disturbance matrix $B_\omega = [B_{a_y} \ B_{\phi_r} \ B_d]$, leading the system presented in (1) to

$$\dot{x}(t) = Ax(t) + B_u \tilde{u}(t) + B_\omega \omega(t) \quad (4)$$

$$y(t) = C_1 x(t) \quad (5)$$

$$z(t) = C_2 x(t) \quad (6)$$

B. H_∞ Output Feedback Controller

The control input is defined as follows

$$u(t) = Ky(t) \quad (7)$$

where K is the control gain matrix to be determined so that, the closed-loop H_∞ performance of the closed-loop system (1) is guaranteed, which is satisfied if the following inequality remains true

$$\|z^T(t)z(t)\|_2 < \gamma^2 \|\omega^T(t)\omega(t)\|_2 \quad (8)$$

C. Event-triggering mechanism

In order to reduce the Transmission Rate (TR), the following event-triggered rule is defined so that the control output is sent to the actuators only when necessary

$$e^T(t)K^T\Omega Ke(t) \leq \varepsilon^2 \tilde{u}^T(t)\Omega \tilde{u}(t) \quad (9)$$

where $e(t)$ is the difference between the current vehicle measurement and the measurement from the last triggering instant, $\tilde{u}(t)$ is the last transmitted control input, ε is a user defined threshold and $\Omega > 0$ is a positive definite matrix to be designed

D. Network communication

The observed measurement $y(t)$ from the vehicle is sampled every h milliseconds, then the event-triggering mechanism decides whether to neglect this information or to update the control input that the actuators must supply. Every time the event-triggering mechanism sends new information through the communication network, this presents a maximum delay of τ_M milliseconds. When the system state varies significantly, actuators will react to stabilize it after a maximum period of $\tau = h + \tau_M$ milliseconds.

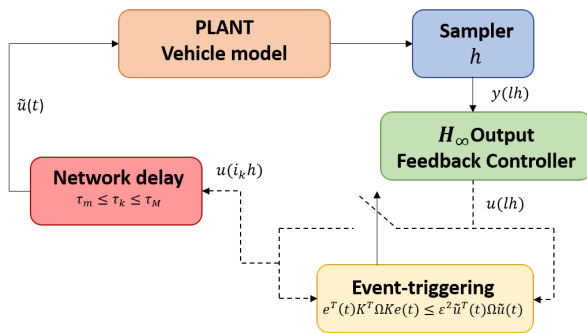


Fig. 1. Diagram of the networked control system.

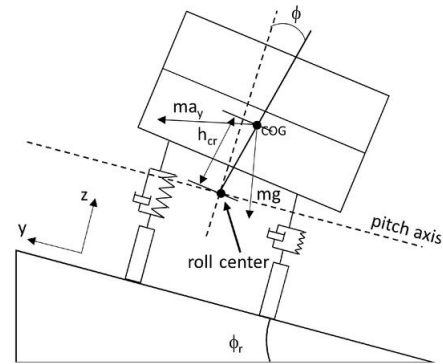


Fig. 2. Vehicle model used in the controller.

E. Lyapunov stability

In order to guarantee the system stability, the following Lyapunov-Krasovskii functional is chosen

$$V(t) = V_1(t) + V_2(t) + V_3(t) \quad (10)$$

where

$$\begin{aligned} V_1(t) &= x^T(t)Px(t) \\ V_2(t) &= \int_{t-\tau}^t \dot{x}^T(s)C_1^T K^T SKC_1 x(s)ds \\ V_3(t) &= \int_{t-\tau}^t (s - (t - \tau))\dot{x}^T(s)C_1^T K^T SRC_1 \dot{x}(s)ds \end{aligned} \quad (11)$$

$$(12)$$

with symmetrical positive matrices $P > 0$, $R > 0$, $S > 0$ to be determined such that

$$\dot{V}(t) < 0 \quad (13)$$

III. SIMULATION RESULTS AND DISCUSSION

The control gain K is obtained through the minimization problem

$$\min \gamma^2 \quad (14)$$

$$\text{subject to } P > 0, R > 0, S > 0, \Omega > 0 \quad (15)$$

A feasible solution was found using the MATLAB LMI Toolbox, with the matrices

$$\begin{aligned} \gamma^2 &= 16.96 \\ P &= \begin{bmatrix} 14.306391 & 0.607416 \\ 0.607416 & 0.464377 \end{bmatrix} \\ R &= 3.4 * 10^{-9} \\ S &= 1.25 * 10^{-10} \\ K &= -13552.53 \\ \Omega &= 1 \\ \epsilon^2 &= 0.1 \end{aligned}$$

The sampling period defined for the observer in this study is $h = 20$ ms and the maximum and minimum delays over the network are assumed as $\tau_M = 20$ ms and $\tau_m = 10$ ms respectively, which are common times for data sampling and communication delays [4] [7]. It is not expensive to use data logging devices with higher sampling rates, however, a sampling frequency of 50 Hz is enough for this study.

For the validation of the proposed controller, the CarSim software is used together with Simulink, where the following experiments were set

- 1) A double line change at a speed of 100 km/h
- 2) A double line change at a speed of 120 km/h, in order to study the performance in a more severe test

The results for the first test are depicted through Figures 3-10. Figures 3 and 5 show the roll rate and angle obtained from this simulation, where it can be seen that the proposed controller achieves a better performance, as the roll rate and angle are lowered through time. These results are summarized

in Table I. Figures 4 and 6 compare the Power Spectral Density (PSD) of the roll rate and angle respectively, where the proposed controller enhances the lateral behaviour, as the PSD is lowered for all frequencies. The anti-roll moment provided by the actuators is depicted in Figure 7. The Event Triggering instants are depicted in Figure 10; the event-triggering mechanism achieves a TR of 29.14% for this experiment. The results for the second test are depicted through Figures 11-18 and summarized in Table II. Once again, the controlled system achieves a better behaviour than the system without control. The TR is 25.33% for this case, which proves that the network usage is reduced.

Normalized load transfer (NLT) was calculated in addition to the above mentioned results. This variable is an accurate measure for evaluating roll stability control systems. It guarantees that the vehicle will not roll over while its value for both axles is within the ± 1 range. The normalized load transfer can be calculated as:

$$NLT_f = \frac{\Delta F_{zf}}{F_{zf}}, NLT_r = \frac{\Delta F_{zr}}{F_{zr}} \quad (16)$$

where F_{zf} and F_{zr} are the total load on the front and rear axle, respectively. Values of NLT for experiments 1 and 2 are depicted through Figures 8-9 and Figures 16-17, respectively. NLT remains within the determined range in all experiments and a decrease of the NLT is observed for both experiments at the front axle, which demonstrates the effectiveness of the controller.

Overall, the vehicle's dynamic response shows a smoother behaviour, ensuring safety and comfort inside the vehicle. Roll angle and rate have been significantly decreased compared to [4], and the TR has been reduced as we have implemented an event triggering mechanism which is more flexible than the one presented in [7] and facilitates the design of the controller.

IV. CONCLUSIONS

An output-feedback event-triggered H_∞ controller that enhances the lateral behaviour of the vehicle has been introduced in this paper. A Lyapunov-Krasovskii functional is chosen to assure the system stability despite the affection of communication delays. An event-triggering rule is designed in order to reduce the communication resources. The controller is designed under a guaranteed H_∞ performance index γ , which represents the impact of the disturbances on the roll rate and angle. To analyze the performance of the proposed controller, the RMS and MAX values of the roll rate and angle are compared, leading to a reduction of up to 50% of the roll rate and angle in the worst cases. The Event-Triggered mechanism reduces the use of network communication resources usage by up to 70%.

TABLE I
ROLL BEHAVIOUR OF THE VEHICLE UNDER THE EXPERIMENT 1

	RMS Roll Rate (°/s)	MAX Roll Rate (°/s)	RMS Roll Angle (°)	MAX Roll Angle (°)
Passive system	3.57	4.15	1.97	3.31
Active system (proposed)	1.70	3.31	1.14	2.22

TABLE II
ROLL BEHAVIOUR OF THE VEHICLE UNDER THE EXPERIMENT 2

	RMS Roll Rate (°/s)	MAX Roll Rate (°/s)	RMS Roll Angle (°)	MAX Roll Angle (°)
Passive system	4.13	11.05	2.18	3.27
Active system (proposed)	2.21	5.38	1.30	2.26

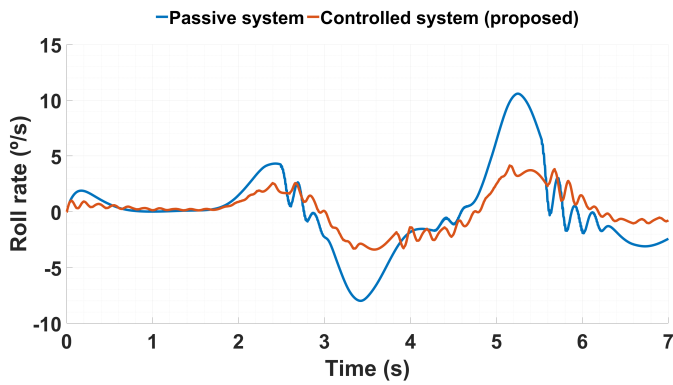


Fig. 3. Roll rate under the experiment 1

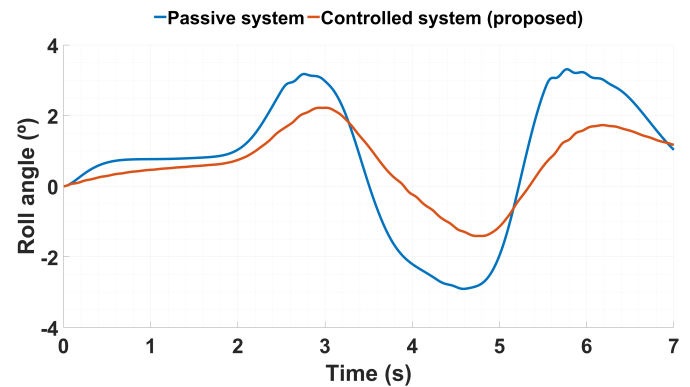


Fig. 5. Roll angle under the experiment 1

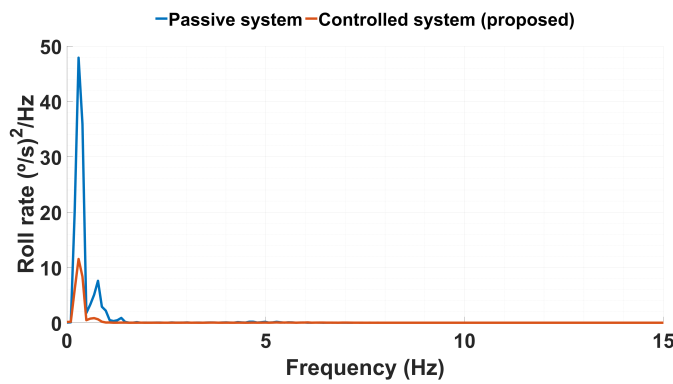


Fig. 4. Roll rate PSD under the experiment 1

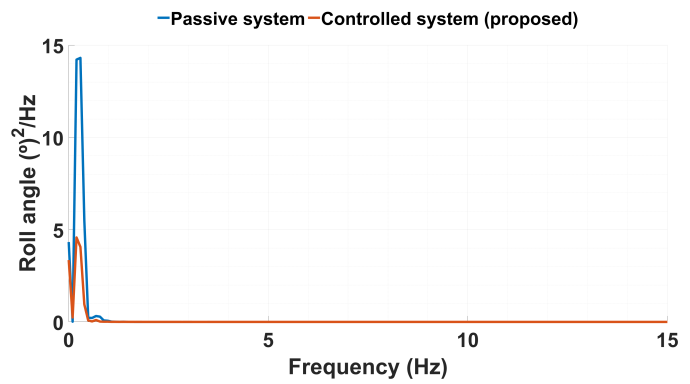


Fig. 6. Roll angle PSD under the experiment 1

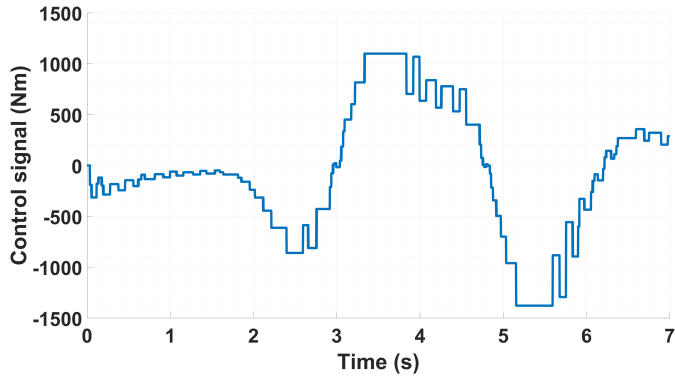


Fig. 7. Control signal under the experiment 1

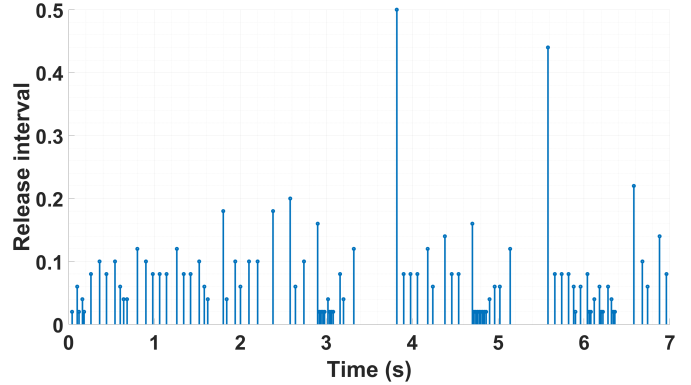


Fig. 10. Event Triggering instants under the experiment 1

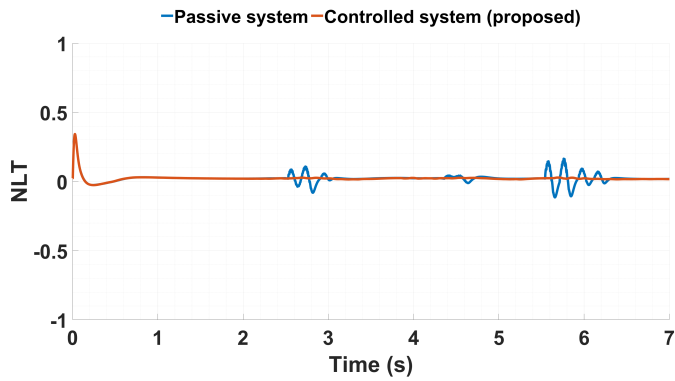


Fig. 8. NLT of front axle under the experiment 1

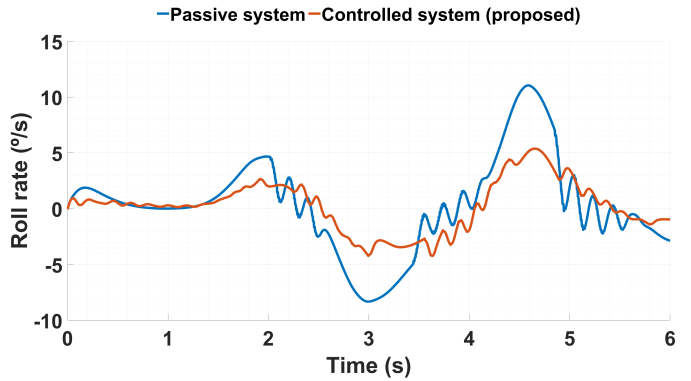


Fig. 11. Roll rate under the experiment 2

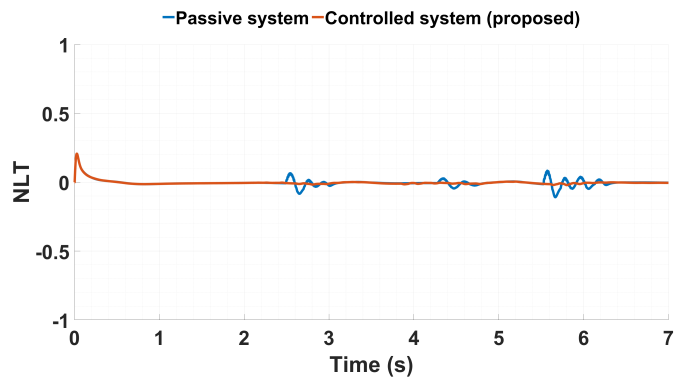


Fig. 9. NLT of rear axle under the experiment 1

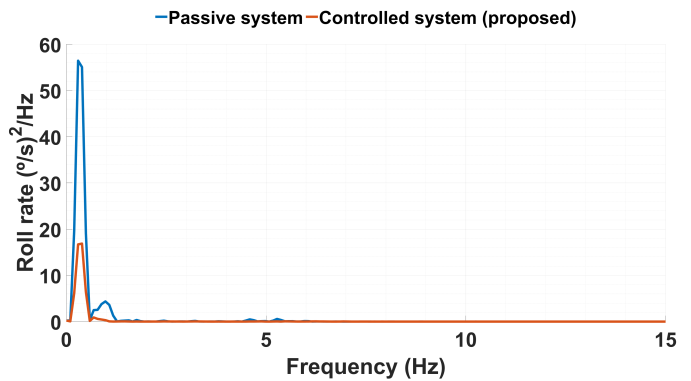


Fig. 12. Roll rate PSD under the experiment 2

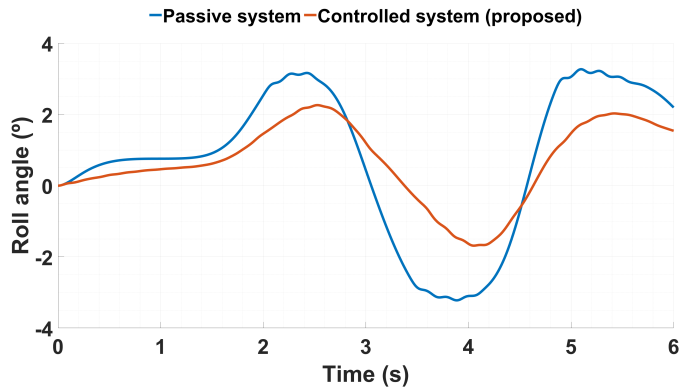


Fig. 13. Roll angle under the experiment 2

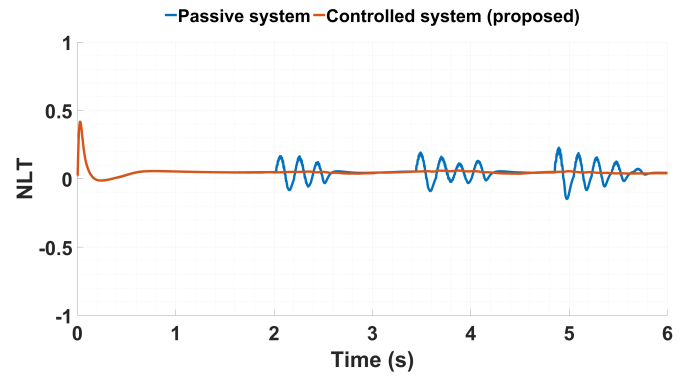


Fig. 16. NLT of front axle under the experiment 2

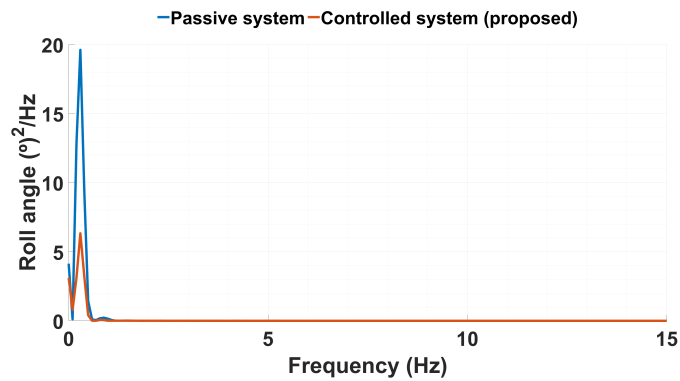


Fig. 14. Roll angle PSD under the experiment 2

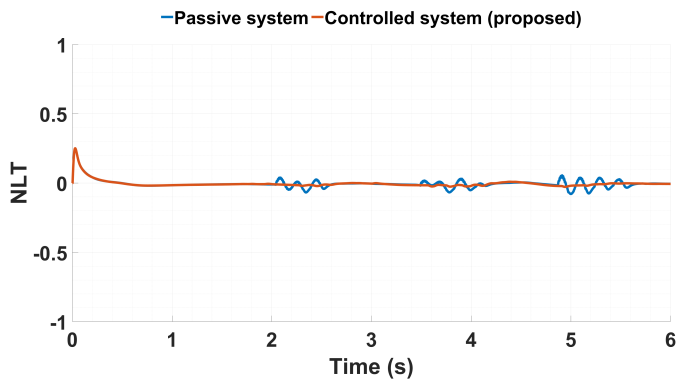


Fig. 17. NLT of rear axle under the experiment 2

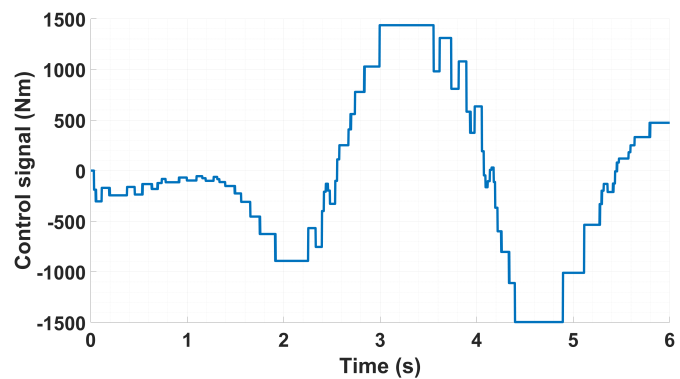


Fig. 15. Control signal under the experiment 2

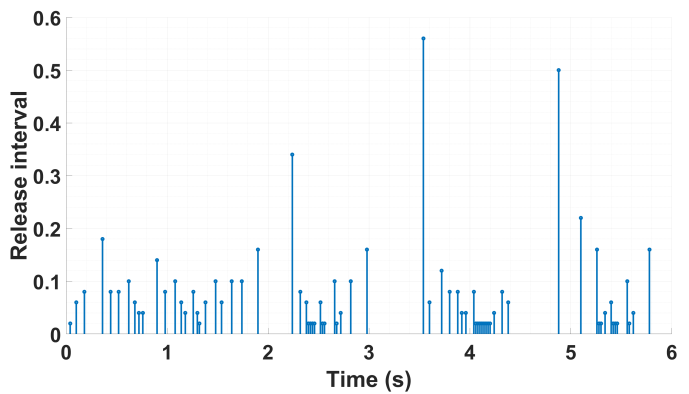


Fig. 18. Event Triggering instants under the experiment 2

ACKNOWLEDGEMENTS

This work was supported by the FEDER/Ministry of Science and Innovation - Agencia Estatal de Investigación (AEI) of the Government of Spain through the project RTI2018-095143-B-C21 and by Centros de Arte, Cultura y Turismo de Lanzarote through the project Cities Timanfaya.

REFERENCES

- [1] K. Shao, J. Zheng, B. Deng, K. Huang, and H. Zhao, "Active steering control for vehicle rollover risk reduction based on slip angle estimation," *IET Cyber-systems and Robotics*, vol. 2, no. 3, pp. 132–139, 2020.
- [2] X. Qian, C. Wang, and W. Zhao, "Rollover prevention and path following control of integrated steering and braking systems," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 234, no. 6, pp. 1644–1659, 2020.
- [3] M. A. Rodríguez Licea and I. Cervantes, "Robust switched predictive braking control for rollover prevention in wheeled vehicles," *Mathematical Problems in Engineering*, vol. 2014, no. 4, pp. 1–12, 2014.
- [4] J. P. Redondo, B. L. Boada, and V. Díaz, "Lmi-based h controller of vehicle roll stability control systems with input and output delays," *Sensors*, vol. 21, no. 23, p. 7850, 2021.
- [5] J. Garcia Guzman, L. Prieto Gonzalez, J. Pajares Redondo, S. Sanz Sanchez, and B. L. Boada, "Design of low-cost vehicle roll angle estimator based on kalman filters and an iot architecture," *Sensors*, vol. 18, no. 6, p. 1800, 2018.
- [6] J. Garcia Guzman, L. Prieto Gonzalez, J. Pajares Redondo, M. M. Montalvo Martinez, and M. J. L Boada, "Real-time vehicle roll angle estimation based on neural networks in iot low-cost devices," *Sensors*, vol. 18, no. 7, p. 2188, 2018.
- [7] F. Viadero-Monasterio, B. Boada, M. Boada, and V. Díaz, "H dynamic output feedback control for a networked control active suspension system under actuator faults," *Mechanical Systems and Signal Processing*, vol. 162, p. 108050, 2022.
- [8] G. Wang, M. Chadli, H. Chen, and Z. Zhou, "Event-triggered control for active vehicle suspension systems with network-induced delays," *Journal of the Franklin Institute*, vol. 356, no. 1, pp. 147–172, 2019.
- [9] W. Lyu and X. Cheng, "A novel adaptive h filtering method with delay compensation for the transfer alignment of strapdown inertial navigation systems," *Sensors*, vol. 17, no. 12, p. 2753, 2017.
- [10] M. J. L. Boada, B. L. Boada, and H. Zhang, "Event-triggering h-based observer combined with nn for simultaneous estimation of vehicle sideslip and roll angles with network-induced delays," *Nonlinear Dynamics*, vol. 103, no. 3, pp. 2733–2752, 2021.

A Multi-UAS Simulator for High Density Air Traffic Scenarios

David Martín-Lammerding

Department of Stats. Comput. Sci. Math
Public University of Navarre (UPNA)
Pamplona, Spain
email:david.martin@unavarra.es

José Javier Astrain

Department of Stats. Comput. Sci. Math
Public University of Navarre (UPNA)
Pamplona, Spain
email:josej.astrain@unavarra.es

Alberto Córdoba

Department of Stats. Comput. Sci. Math
Public University of Navarre (UPNA)
Pamplona, Spain
email:alberto.cordoba@unavarra.es

Abstract—Unmanned Aerial Systems (UASs) have become enormously popular as they improve many applications like structural inspections, homeland security, delivering packages, etc. UASs share the same airspace as manned aircrafts and thus autonomous UAS operations and current airspace management have to be integrated. It is recognized that the use of small UASs and taxi-drones at lower altitudes is now a driving force of economic development, but a safety risk when their numbers increase. UASs will fly and make decisions autonomously using only on-board sensors and processors. As the number of simultaneous UAS flights increase, keeping a safety airspace is a challenge that implies that UAS navigation autonomy must be verified for conflict avoidance in high density air traffic areas. The required scenarios to verify air traffic management algorithms and autonomous capabilities of UASs are complex to deploy and UASs may collide, therefore simulations are of great importance. This paper presents a conflict simulator of multiple UASs with different equipment. The main purpose of the simulator is to verify UAS subsystems, like collision avoidance, in different scenarios with a combination of different equipped UASs and flight plans. To evaluate its effectiveness, we performed an integrated simulation process for a Collision Avoidance Systems (CAS) implementation where Hardware In the Loop (HIL) and software simulations are combined to improve UAS subsystems safety and to reduce the time-to-market.

Keywords—*Simulator; HIL; UAS; autopilot; autonomous; CAS; conflicts; anti-collision; air traffic.*

I. INTRODUCTION

The use of Unmanned Aerial Systems (UASs) in various fields, such as military, policing, firefighting, etc. has evolved in multiple UAS types with different characteristics, such as size, speed and flight range. Each UAS must be equipped with new safety systems, like a Collision Avoidance System (CAS), to fly the shared airspace with other aircraft. CASs detect airplanes in airspace, discover potential collision hazards and perform maneuvers to avoid collisions. Rigorous simulations of UAS traffic benefits application development of any UAS subsystem, like CAS, and the integration of UASs in the common air-space safely. Therefore, there is a real need of a simulator that can combine multiple UASs with different equipment and capabilities.

Accelerating, enhancing the development and testing of CAS and other software elements that provide autonomy to UAS are a key challenge. The logistical effort to deploy multiple conflicting UASs imposes a high cost and complexity.

Evaluating the effectiveness of UAS systems requires an automated process to simulate every new release.

In this paper, we present an UAS simulator, called *SIMU-drone*, that is suitable for simulating high density air traffic operations with UASs. The simulator supports multiple UASs in user-predefined scenarios and in randomly automatic generated scenarios. Each simulation consists of multiple UASs with a Flight Control Unit (FCU) and a Global Navigation Satellite System (GNSS). Simulated UASs may have an autopilot, a CAS, sensors for conflict detection (video-camera, LIDAR, etc.) or a radio equipment to control it remotely by a pilot. The main motivation for developing this simulator lies in the necessity to generate multiple conflict scenarios with combinations of multiple types of UASs and to integrate in the simulations specific UAS subsystems executed on specific hardware.

The rest of the paper is structured as follows. Section II presents the state of the art of UAS simulators, Section III defines the problem statement and Section IV describes our contribution. The simulator is presented in Section V. Section VI is devoted to presenting an experimental simulation. Section VII presents the conclusions and references end the paper.

II. RELATED WORK

Next, we review the main Commercial Off-The-Shelf (COTS) UAS simulators available.

The *SITL simulator* is used for testing an autonomous quadcopter [1]. It is intended for executing a FCU based on ArduPilot in a PC. However, the *SITL simulator* does not allow to simulate multiple UASs simultaneously. Other UAS simulators focus on simulating the physical model of the vehicle. Some vehicle simulators are XPlane [2], Flightgear [3], Gazebo [4], JMavSim [5][6]. For our purposes, any of the previously reviewed simulators requires a costly adaptation to model UAS traffic.

AirSim [7] is a visual simulator of different types of vehicles, including UAS, based on the video game engine Unreal Engine. It is focused on the development of AI algorithms based on deep learning. Simulation time can be reduced but at the cost of losing accuracy. However, response time and multiple UASs simulations are limited as the simulation runs in

a desktop computer and it does not provide predefined conflict scenarios.

UTSim [8] is an UAS conflict simulator based on Unity. Anti-collision algorithms can be integrated in the simulation executed by UTSim, but UAS sub-systems can not be run in HIL mode. There are air traffic management simulators that include UAS traffic, such as BlueSky [9]. This type of simulator is focused on air traffic planning but not on simulating conflicts and collisions. In [10], two simulators are proposed: the TIMed State space Performance Analysis Tool (TIM-SPAT) [11] and the CPN-tools [12]. The former requires a complex configuration. The latter is a limited tool with short simulation configuration capabilities.

In [10], an anti-collision system is simulated using TIMed State space Performance Analysis Tool (TIM-SPAT) [11], developed at the Logistics and Aeronautics Unit of the Autonomous University of Barcelona. But it has a complex configuration and development. Another tool used in [10] is CPN-tools [12], a basic simulator with limitation in the UAS equipment configuration.

A Real-time Multi-UAV Simulator (RMUS) was presented in [13]. It supports multiple UAVs, data collection and control but lacks a flexible configuration of environments.

Multiple UASs simulators exist [14][15] but they are not focused on testing multiple conflicts and their avoidance.

Table I classifies the main UAS simulators using selected capabilities: HIL simulation, multiple UASs simulation, and external implementation/code integration in the simulator.

TABLE I
MAIN UAS SIMULATORS AND TOOLS FEATURES.

	Ref	HIL simulation	Multiple UASs	Data link	External code integration
Airsim	[7]		✓		
Gazebo	[4]		✓		
UTSim	[8]		✓	✓	✓
RMUS	[13]	✓		✓	
SIMUdrone	This	✓	✓	✓	✓

III. PROBLEM STATEMENT

UAS applications deployment require a security verification for the onboard systems to fly in high density air traffic scenarios. Very Low Level airspace (VLL) is the space below 500 ft. above ground level. It is the part of the airspace intended for new UAS applications and it will concentrate most UAS conflicts. A conflict between two UASs occurs when minimum separation, defined as the protection radius, r_p , is lost. Figure 1 shows a conflict between *local UAS* and *remote UAS*. A loss of separation does not always predict a future collision, but it is a key safety indicator.

An UAS flies over some particular locations, called waypoints. A waypoint is a location determined by GNSS. A flight plan consists of an ordered set of waypoints that the onboard autopilot follows. A CAS deployed in an UAS is aimed at maintaining a minimum safe separation between UASs. Once a conflict is detected, a CAS diverts the UAS to a new safe

path, changing the flight plan autonomously. The number of simultaneous conflicts are denoted as n_c .

UAS traffic in VLL airspace consists of UASs with an autopilot, remotely piloted UASs and even autonomous UASs. Therefore, CAS must deal with conflicts with different UAS types, capabilities and in random relative locations.

Most CASs for UASs are distributed so they run in an onboard computer, commonly denoted as an embedded board. However, the payload of the UASs limits the weight and the energy consumption of the embedded board, as well as its computing power.

One of the derived requirements from the above is that the response time of the CAS running in an embedded board should be considered in any simulation as it can reduce the time to react.

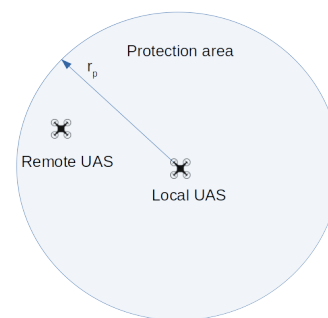


Figure 1. Conflict between a local UAS and a remote UAS.

UAS software is necessarily integrated with hardware and embedded systems for autonomous behavior. An intensive testing with live-fly field experiments are costly and highly time consuming. Therefore, simulations should combine HIL simulations and other simulations based on software models.

IV. CONTRIBUTION

SIMUdrone is a novel multi-UAS simulator for high density air traffic scenarios suitable for urban environments. It provides a unique conflict generation framework with extensive configuration capabilities in order to simulate the variability of situations in urban environments. Integration of external software components and hardware in the simulations are two key features of *SIMUdrone*.

SIMUdrone improves UAS safety and reduces the time to market for UAS subsystems development as it implements an automated integrated continuous simulation for each new UAS subsystem release. The automated integrated continuous simulation combines HIL simulations with simulation based on software model.

SIMUdrone simplifies the verification of UAS subsystems, like CAS, as it can generate extensive traffic combinations to simulate different scenarios. A large dataset of data from every simulation is available to verify its behavior.

V. SIMUDRONE SIMULATOR

SIMUdrone simulates UAS traffic, the effect of the environment surrounding the UAS during the flight, the onboard equipment of the UAS and it can integrate external UAS hardware using the *HIL mode*.

SIMUdrone is implemented in JAVA 8, HTML, CSS and JS. The simulation output consists of multiple configurable log files and a HTML5 conflict animation. A conflict animation simplifies the analysis of a simulation scenario and the results obtained. It can be easily shared with others as it can be viewed in any PC with a browser.

To generate conflicting UAS scenarios, *SIMUdrone* provides custom scenarios, randomly generated with configurable UASs types. UAS equipped with an autopilot can fly following a set of waypoints that can be configured, as well as the flying speed. UASs with a CAS integrated with the autopilot can avoid conflicts by changing the predefined flight plan. UASs remotely piloted can be also included in a simulation. To simulate a remotely piloted UAS, *SIMUdrone* allows to configure an area centered on the pilot where the UAS flies randomly for a time interval to limit its duration.

A. Architecture

SIMUdrone consists of multiple software components that are combined to simulate a complete UAS system and a configurable traffic generator. A complete UAS system can be simulated combining software models with a HIL simulation to test a software component running on a specific piece of hardware.

To simulate a flight environment and provide sensor inputs during simulation, we develop a virtual collaborative transponder and a sensor system that generates conflict data. The CAS log, the autopilot status and the scenario animation are generated and stored in a proper format for each type.

SIMUdrone implements an autopilot and a reference CAS implementation to facilitate the comparison with other implementations. The CAS implemented in *SIMUdrone* is based on the Potential Field (PF) technique. PF is a collision avoidance path calculation method that simulates a force field where the UAS is attracted to its final destination and repulsed from obstacles or conflicts [16], [17].

SIMUdrone implements two modes of operation, the *HIL-mode* and the *integrated-conflict-mode*, as depicted in Figure 2.

The *HIL-mode* is devoted mainly to run a CAS implementation in the planned hardware that will be used as a companion-*pc* on the UAS. The *HIL-mode* is a testbed that provides a safe environment for testing any external systems in real time, like CAS. *SIMUdrone* implements two adaptation layers, a *SIMUdrone* Input Layer (SIL) and a *SIMUdrone* Output Layer (SOL), to integrate external implementations running on a specific piece of hardware. The communication between *SIL*, *SOL* and *SIMUdrone* could be configured by a REST interface or a serial communication using the MAVLink protocol.

A *SIMUdrone* HIL simulation can be deployed in two ways:

The first consists of a single embedded board running the CAS in order to profile its response time and use it in a later

simulation. Therefore, a trusted CAS response time is obtained and it is available for subsequent simulations. This is the initial step before a complete software simulation is performed. The two layers are adaptable and modular to simplify the integration with external hardware. The HIL response time profiling of *SIMUdrone* works as follows: *SIMUdrone* sends conflict data to the input layer and receives the maneuver at the output layer. *SIMUdrone* coordinates the two layers to obtain the response time for each input-output.

The second consists of n_C boards connected to *SIMUdrone*. Each board runs the CAS of a simulated UAS whose FCU and the rest of its equipment are software models instantiated by *SIMUdrone*. Each board runs a CAS instance that receives conflict data and sends avoidance maneuvers through the SIL and SOL layers, respectively.

The *integrated-conflict-mode* is devoted to simulate multiple UASs in a cluster using software models for the UAS equipment, the environment and the communications. The *integrated-conflict-mode* simulates n_C conflicting UASs with different equipment flying in a conflict area. It simulates the scenario in virtual time considering the real response time of the UAS equipment, the latency of communications and the delay of any onboard processing.

External implementations of UAS subsystems can be integrated in the simulator as components or libraries linked to the simulator code. However, UAS subsystems response time may not be the same when running in a cluster. For this case, an initial HIL simulation is required to provide data to configure and tune a response time model.

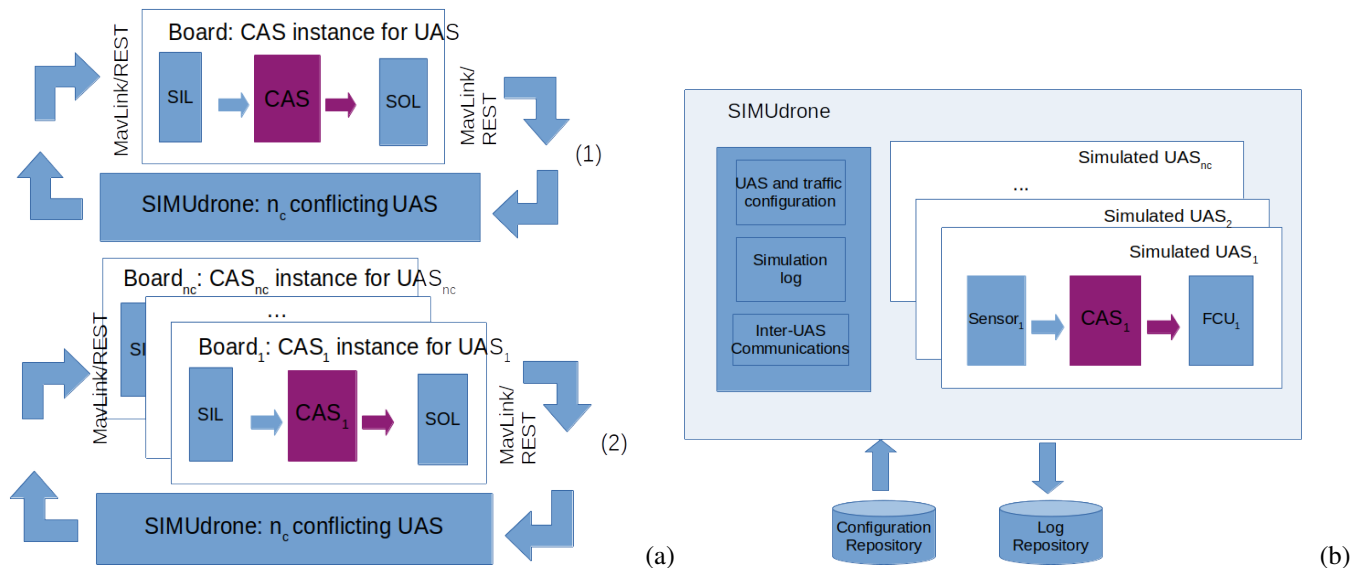
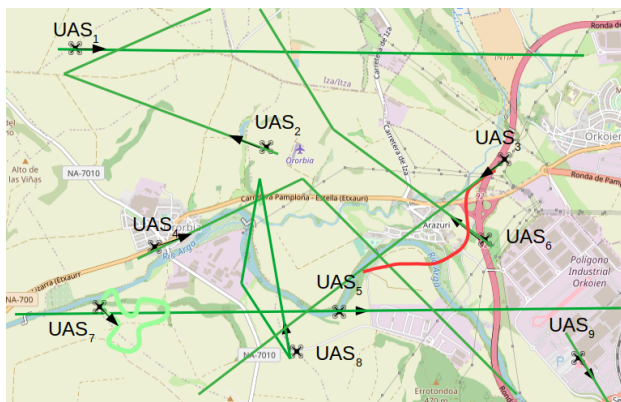
B. Simulated scenarios

Simulated scenarios can be defined programmatically. However, there are two pre-defined scenarios for convenience, a *conflicting-area* and a *conflicting-point* scenario.

A *conflicting-area* scenario consists of a rectangular airspace area where n_C configurable UASs fly. A default configuration defines a two waypoint flight plan for any autopilot-equipped UAS. Initial UAS location, heading, speed and fly distance are randomly distributed using configured intervals. This scenario is useful to test different densities of conflicts and how a conflict influences others, as depicted in Figure 3.

In a *conflicting-point*, *SIMUdrone* simulates n_C UASs that converge in a collision point p_C . To achieve this, *SIMUdrone* defines a conflict-circle, whose center is the collision point p_C . Each UAS is randomly positioned on a conflict-circle circumference. The minimum separation considered among UASs on the conflict-circle circumference is a configuration parameter. The point p_C is the center of the conflict-circle circumference. Therefore, the flight plan is the diameter composed of three points, the initial point on the circumference, w_1 , the center, p_C and w_2 . This scenario is useful when it is required to test a CAS with a fixed n_C conflicting UASs in a limited area, as depicted in Figure 4.

A *conflicting-point* simulation mission starts at waypoint w_1 and it is completed if the UAS arrives at w_2 without being involved in a collision. In addition to the conflict area around


 Figure 2. *SIMUdrone* modes: (a) HIL mode (b) Conflict mode

 Figure 3. *Conflicting-area* simulation of 9 UAS in a 5x5 Km. area. UAS₃ is an autonomous UAS equipped with a CAS that avoids a collision changing its planned path (in green) to the one depicted in red. UAS₇ is a remotely piloted whose path revolves around a point as the pilot performs a visual line of sight flight. The rest of UAS are equipped with an autopilot.

point p_C , there may be other conflict areas located at the beginning of the flight if the separation between UASs is lower than the conflict distance configured. Animation examples are available at [19] and [20].

VI. SIMUDRONE SIMULATION

Next, we present a *SIMUdrone* simulation configured to show an automated integrated continuous simulation that combines an HIL simulation and a software model simulation.

A. Configuration

First, we review *SIMUdrone* general configuration capabilities. After that, we present the configuration used for the simulation.

SIMUdrone is a customizable simulator with a set of configuration parameters to simulate multiple conflict scenarios in

any airspace class, multiple UAS typologies, multiple payloads and different flight plans. Simulated UASs types available in *SIMUdrone* are Vertical Take-Off and Landing (VTOL) or copter. Their class are *micro*, *mini light* or *small* [18] so their Maximum Take-Off Mass (mtom) is less than 150 Kg. This are the most common types in the VLL airspace. Multiple UAS flight altitudes can be simulated. Conflicts can be avoided by changing altitude maneuvers but the limited altitude in VLL airspace and the limited precision for latitude measurement, restrict the crosses at different altitudes. The PF implementation available in *SIMUdrone* does not change altitude. Other known CAS implementations, that can be simulated in any of the two *SIMUdrone* modes, may change altitudes.

The simulated UAS can be configured in three types: autopilot, autonomous with CAS and remotely piloted. The autopilot and the autonomous UAS require a flight plan defined with a set of waypoints, speeds between waypoints and flight altitudes. The remotely piloted UAS requires a pilot location and a maximum distance from it to generate a random path that is contained in a cylinder whose base is defined by the pilot location and the radius is the maximum distance for the visual flight allowed. *SIMUdrone* can be configured to simulate the effect of wind and GNSS's inaccuracies with random variations of the bearing required to fly to the next waypoint.

SIMUdrone simulates conflict detection sensors, like an ADS-B transponder or a vision camera. A simulated vision-based detection system consists of a vision camera mounted in the forward direction of the UAS on the symmetry axis. The CAS implemented in *SIMUdrone* generates an escape maneuver when the distance to a conflict is less than the protection radius (r_p). If the distance is less than the safety radius (r_s) the escape maneuver is sharper. A collision occurs between two or more UASs if the distance between them is

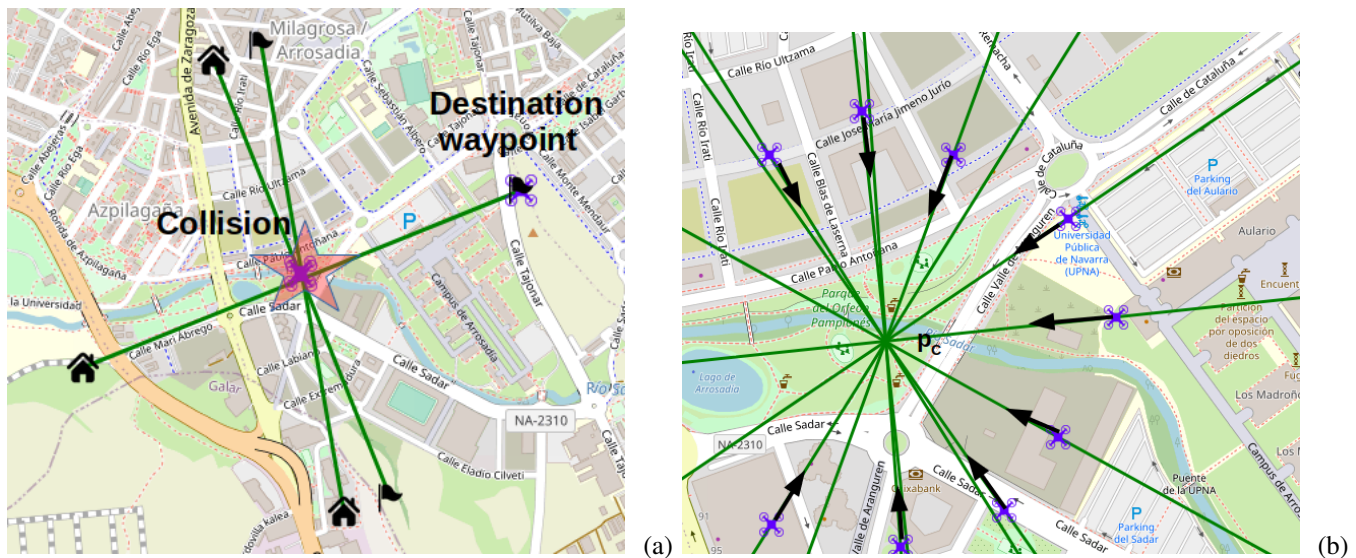


Figure 4. *Conflicting-point* simulations.(a) Two UASs collide and a third one arrives its destination waypoint (b) Nine UASs approaching a collision point pc

less than the collision radius (r_{cc}).

For the simulation performed, a specific set of simulation parameters are selected and summarized in Table II.

TABLE II
MAIN *SIMUdrone* PARAMETERS CONFIGURED.

Parameter	Value
Typology of simulated UAS	Copter
UAS class	Light
Simulated UAS	All autonomous UAS with CAS
CAS integrated	<i>SIMUdrone</i> PF implementation
Onboard sensors	ADS-B and vision camera
n_C	[2,9]
Protection radius (r_p)	100 m
Safety radius (r_s)	50 m
Collision radius (r_{cc})	5 m
Speed range (sp)	[20, 25] m/s
UAS distance to pc	[350, 650] m
Altitude (h)	50 m
GNSS inaccuracies distance	[-2, 20] m random
GNSS inaccuracies bearing	[0, 360] °
Wind effect bearing variation	[-4,4]°

Tables III and IV list the features of the equipment configured in the simulation.

TABLE III
SIMULATED PARAMETERS OF THE VISION-BASED CAMERA.

Parameter	Value
Detection distance (d_{DT})	[300,350] m
Coverage angle (θ)	[140,160] degrees
Frequency (f_c)	20 Hz

B. Simulation results

The *SIMUdrone* conflict simulation performed combines a HIL simulation and two *integrated-conflict-mode* simulations configured as *conflicting-point* scenarios.

TABLE IV
SIMULATED PARAMETERS OF A LOW POWER ADS-B TRANSPONDER.

Parameter	Value
Coverage distance (d_{CB})	[1500,2500] m
Frequency (f_{ec})	1 Hz

The HIL simulation obtains a Response Time, RT, of the *SIMUdrone* CAS implementation running in a PI3 embedded board [21]. The *SIMUdrone* CAS runs in a PI3 board connected to the main *SIMUdrone* instance via MAVLink. *SIMUdrone* generates conflicts that are sent to the board and keeps the conflict scenario state. Next, *SIMUdrone* waits for the CAS response, updates the internal simulation state and measures the response time.

The *integrated-conflict-mode* simulation is performed in two different *conflicting-point* scenarios: One of the scenarios simulates n_C UASs without CAS and the other simulates n_C UASs with the CAS implemented in *SIMUdrone* using the response time obtained from the previous HIL simulation.

Results of the simulations are shown in Table V. A simulated mission with n_C conflicting UASs is successful if every UAS arrives to its destination waypoint. The success rate measures how many iterations are successful. The number of conflicts simulated are between 2 and 9. The upper value is defined using the results of a worst-case high traffic volume (maximum of 100000 flights per day) presented in [22]. Every simulation performed repeats 100 iterations for each scenario.

The performed *SIMUdrone* simulation shows that it allows to combine multiple simulation modes in order to automate the test of an UAS subsystem implementation. When a new version of the UAS subsystem is available, *SIMUdrone* allows to update the simulations results performing a new simulation with the new implementation version. The integrated simulation process configured in *SIMUdrone* runs the updated imple-

TABLE V
SUMMARY OF SIMULATIONS PERFORMED.

n_C	mean		success		Collided UASs (no CAS)		success		Collided UASs with CAS (equip CAS)	
	RT (ms)	%	mean	sdev	mean	sdev	mean	sdev	mean	sdev
2	6.45	59	2.00	0.00	99	2.00	0.00	2.00	0.00	0.00
3	8.44	13	2.01	0.11	78	2.00	0.00	2.00	0.00	0.00
4	9.45	0	2.57	0.89	69	2.06	0.36	2.06	0.36	0.36
5	8.75	0	3.68	0.72	64	2.17	0.56	2.17	0.56	0.56
6	7.68	0	4.80	0.97	38	2.24	0.99	2.24	0.99	0.99
7	8.86	0	5.70	0.76	32	3.00	1.22	3.00	1.22	1.22
8	8.56	0	6.81	1.13	25	2.89	1.21	2.89	1.21	1.21
9	8.45	0	7.75	0.82	20	3.13	1.30	3.13	1.30	1.30

mentation in a embedded onboard, obtains updated response times, updates the software model and simulates multiple scenarios with UASs equipped with the new implementation.

VII. CONCLUSION

UAS evolution requires the rapid implementation and evolution of UAS subsystems for its integration in air traffic. The verification of such UAS subsystems must be performed in simulations because it is expensive if it is physically performed. Therefore, simulators are vital to evaluate proposed algorithms and their performance in specific hardware. In this paper, we present *SIMUdrone*, a simulator that combines scenarios with multiple UASs, HIL simulations and the integration of external algorithms in any subsystem. It allows to configure UASs with different equipment, like an ADS-B transponder or a CAS, and with custom or random generated flight plans. Furthermore, *SIMUdrone* has been tested using a combined HIL simulation with a software simulation of n_C conflicts to verify its capability to integrate and automate a complete stack of continuous simulations. A continuous simulation process reduces the time-to-market and improves air traffic safety.

Additional enhancements will be focused on the development of more simulated components in order to test other collaborative technologies, sensors or new conflict avoidance algorithms. Another line of work is to improve the *SIMUdrone* interoperability with algorithms implemented as external binaries that can not be integrated in the code or executed in *HIL mode*. Further developments of this work will include integration in the *HIL mode* of hardware that uses V2X (Vehicle-to-everything) communication protocols or Controller Area Network (CAN) BUS. Future work will be focused on the creation of a *dataset* with conflicts and resolution maneuvers to be used for UAS air traffic research and for development of CAS.

REFERENCES

- [1] H. M. Qays, B. A. Jumaa, and A. D. Salman, "Design and implementation of autonomous quadcopter using sitl simulator," *Iraqi Journal of Computers, Communications, Control and System Engineering*, vol. 20, no. 1, pp. 1–15, 2020.
- [2] R. Garcia and L. Barnes, "Multi-uav simulator utilizing x-plane," in *Selected papers from the 2nd International Symposium on UAVs, Reno, Nevada, USA June 8–10, 2009*, Springer, 2009, pp. 393–406.
- [3] A. R. Perry, "The flightgear flight simulator," in *Proceedings of the USENIX Annual Technical Conference*, vol. 686, 2004.
- [4] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, IEEE, vol. 3, 2004, pp. 2149–2154.
- [5] Y. Jiang, J. Liao, and Q.-f. Chen, "Hardware-in-loop simulation system for small rotor uav based on jmvmsim," *Computer Simulation*, 2019.
- [6] A. I. Hentati, L. Krichen, M. Fourati, and L. C. Fourati, "Simulation tools, environments and frameworks for uav systems performance analysis," in *2018 14th International Wireless Communications Mobile Computing Conference (IWCMC)*, 2018, pp. 1495–1500. DOI: 10.1109/IWCMC.2018.8450505.
- [7] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and service robotics*, Springer, 2018, pp. 621–635.
- [8] A. Al-Mousa, B. H. Sababha, N. Al-Madi, A. Barghouthi, and R. Younis, "Utsim: A framework and simulator for uav air traffic integration, control, and communication," *International Journal of Advanced Robotic Systems*, vol. 16, no. 5, p. 1729881419870937, 2019.
- [9] J. M. Hoekstra and J. Ellerbroek, "Bluesky atc simulator project: An open data and open source approach," in *Proceedings of the 7th International Conference on Research in Air Transportation, FAA/Eurocontrol USA/Europe*, vol. 131, 2016, p. 132.
- [10] J. Tang, F. Zhu, and L. Fan, "Simulation modelling of traffic collision avoidance system with wind disturbance," *IEEE Aerospace and Electronic Systems Magazine*, vol. 33, no. 4, pp. 36–45, 2018.
- [11] O. T. Baruwaa, M. A. Piera, and A. Guasch, "Timsat-reachability graph search-based optimization tool for colored petri net-based scheduling," *Computers & Industrial Engineering*, vol. 101, pp. 372–390, 2016.
- [12] K. Jensen, L. M. Kristensen, and L. Wells, "Coloured petri nets and cpn tools for modelling and validation of concurrent systems," *International Journal on Software Tools for Technology Transfer*, vol. 9, no. 3, pp. 213–254, 2007.
- [13] A. H. Goktogan, E. Nettleton, M. Ridley, and S. Sukkarieh, "Real time multi-uav simulator," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, IEEE, vol. 2, 2003, pp. 2720–2726.
- [14] V. Rodriguez-Fernandez, H. D. Menéndez, and D. Camacho, "Design and development of a lightweight multi-uav simulator," in *2015 IEEE 2nd International Conference on Cybernetics (CYBCONF)*, IEEE, 2015, pp. 255–260.
- [15] M. A. Day, M. R. Clement, J. D. Russo, D. Davis, and T. H. Chung, "Multi-uav software systems and simulation architecture," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2015, pp. 426–435.
- [16] J.-H. Chuang and N. Ahuja, "An analytically tractable potential field model of free space and its application in obstacle avoidance," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 28, no. 5, pp. 729–736, 1998.
- [17] Y. Zhao, L. Jiao, R. Zhou, and J. Zhang, "Uav formation control with obstacle avoidance using improved artificial potential fields," in *2017 36th Chinese Control Conference (CCC)*, IEEE, 2017, pp. 6219–6224.
- [18] A. C. Watts, V. G. Ambrosia, and E. A. Hinkley, "Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use," *Remote Sensing*, vol. 4, no. 6, pp. 1671–1692, 2012.
- [19] D. Martín-Lammerding, (2021). Two uas collision, <https://dronetology.net/vehicular/collision.html>, [Online]. Available: <https://dronetology.net/vehicular/collision.html> (visited on 01/02/2022).
- [20] —, (2021). Dense air conflicts, <https://dronetology.net/vehicular/dense.html>, [Online]. Available: <https://dronetology.net/vehicular/dense.html> (visited on 01/02/2022).
- [21] Raspberry Pi Foundation. (2021). Raspberry Pi 3, <https://www.raspberrypi.org/>, [Online]. Available: <https://www.raspberrypi.org/> (visited on 02/02/2021).
- [22] V. Bulusu, R. Sengupta, V. Polishchuk, and L. Sedov, "Cooperative and non-cooperative uas traffic volumes," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2017, pp. 1673–1681.

Simulation-based Testing of Service Drones in U-Space Environments

Moritz Höser

Bauhaus Luftfahrt e. V.

82024 Taufkirchen, Germany

moritz.hoeser@bauhaus-luftfahrt.net

Kai-Daniel Büchter

Bauhaus Luftfahrt e. V.

82024 Taufkirchen, Germany

kai-daniel.buechter@bauhaus-luftfahrt.net

Abstract—With the introduction of U-Space, upcoming drone flight controllers are required to combine safety, autonomy and human interaction with evaluable regulatory compliance. In these regards, testing drones in virtual environments is a promising approach to enhance the development of drone controllers. With virtual drone test environments being actively developed, questions concerning general validity arise including a) how drone-testing scenarios should be described, set up and executed in a transferable manner and b) what conclusion developers and regulators can draw from such experiments. This work discusses potential benefits and interim results for creating, executing and later evaluating scenarios for virtual drone testing environments. The work includes early stages on the research on scenario content and simulation execution concept.

Index Terms—drones, virtual testing, simulation, autonomy, U-Space, compliance, scenario.

I. INTRODUCTION

The gradual introduction of the U-Space framework for drone operation is facilitating drone-based services and future airspace applications. Upcoming unmanned service drones are expected to operate comprehensibly and in an automated way while adhering to strict safety standards; their autopilot controllers will be part of a mobile, cyber-physical system that operates in an increasingly complex environment. This complexity poses major challenges to developers and regulators alike and may also raise concerns among the affected public.

Scenario-based testing and simulations can cover some of the requirements to evaluate system behavior in complex situations. The scenarios thereby have to fulfill requirements on reproducibility, interpretability and traceability.

The automotive sector is pushing ahead with scenario-based testing, among other things, in the pursuit of autonomous driving. However, these works are often only of limited use for drones. Drones in the Very Low-Level (VLL) airspace are confronted with the practical challenges in safely providing drone services on the one hand, and on the other hand meeting regulatory requirements of Unmanned Traffic Management (UTM).

This work presents considerations and first results on describing and executing simulation scenarios for service drones. Section II states the underlying research questions and prior research. Section III describes the results as presented in the work. Section IV ends with conclusion and future work.

II. RELATED WORK

A. Research question

Considering the objectives concerning UTM and drones, the authors define the following main research question:

RQ: What content should drone testing scenarios for UTM cover and how can it be described in a tool-independent manner? The RQ targets especially reproducibility, traceability and interpretability of test configurations, supporting the generation of valuable results through formalization of and automation in virtual testing conditions.

B. Related research

Related research is only briefly mentioned here due to limited space but can draw from a wide range of publications and ongoing research. Basic definitions on the subject of scenarios are summarized in [2], among others. Observations on the development cycle of scenarios and validation for automated systems and vehicles can be found in [1] and [3]. OpenScenario [5] is one example for an ongoing, domain-specific approach to establish a scenario description for the automotive industry. Several publications originating from [4] and [6] apply similar concepts to the aviation space.

III. WORK AND RESULTS

A. Challenges and requirements

Service drone platforms face a number of challenges: Firstly, the development of controllers for autonomous drones is undergoing active development, including the need to test real-time sensor data processing and algorithms for autonomous flight. Second, the concerns of future stakeholders should already be taken into account: in addition to developers and operators, these include UTM regulators, but also a concerned public, which will be indirectly confronted with the drones. Third, during and in addition to the actual tasks, drones will also face unexpected challenges that should be simulated to some extent in order to be able to master later situations without interventions and to make tests more convincing for the interested public.

B. Approach

The steps presented here center on the tasks from scenario content modeling up to scenario execution in the simulator.

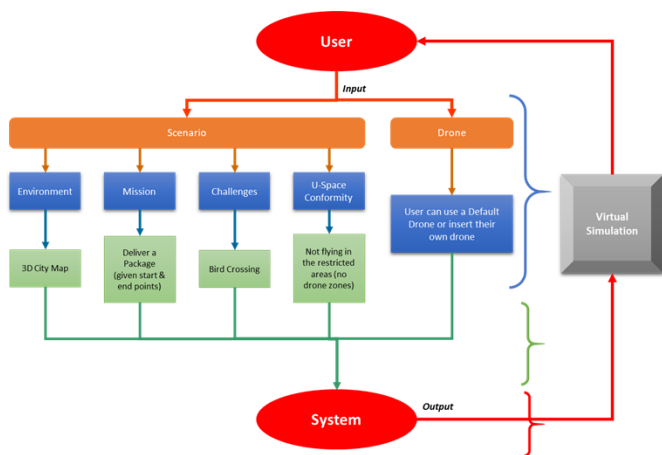


Fig. 1: Scenario elements (in blue) with examples (in green), in the context of a simulation system

Fundamental for the modeling process are the considerations about the actual purpose of the scenarios in its application domain, which is reproduced among others in [1, p. 29]. Making use of scenario-based testing therefore starts with the *requirements* for the scenario, continues with its *implementation* and ultimately ends in the monitoring related tasks *verification and validation*, which provide the actual insight.

The current state of this research focuses on scenario modeling and implementation options, as it sets the stage for a meaningful analysis of monitoring results later on. In the best case, the work should allow other users and researchers to create own scenarios and achieve valuable and comprehensible simulation results for evaluation and development.

C. Results

The figures describe the early results of discussions and research on scenario contents, format and execution. In principle, unmanned drones with a specific task (mission) in an urban or rural area are to be considered. These drones operate in the VLL and must comply with a number of UTM regulations. In Europe, these rules are defined by the U-Space framework, which defines different airspace types and service levels.

The discussed requirements – summarized by the goal of testing drone capability and conformity with the prospect on verification and validation – lead to the *basic elements* of a testing scenario as depicted in Figure 1 and Figure 2.

The basic building blocks, in addition to the environment, are the main elements *U-Space*, *missions* and *challenges*. In the upcoming work, a data model and open specification for the format and content of these building blocks will be developed. In this process, findings from other groups can be drawn upon, including those from the automotive, aviation and simulation fields.

In particular, the so-called “Challenges” are difficult to integrate in the scenario model, since both variability and reproducibility are to be achieved at the same time. Challenges are intended to create an unexpected problem for the drone

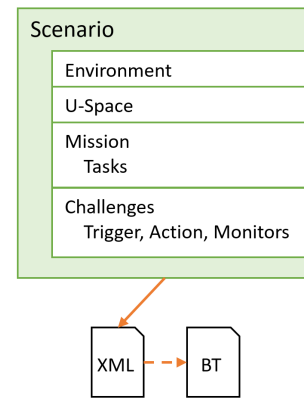
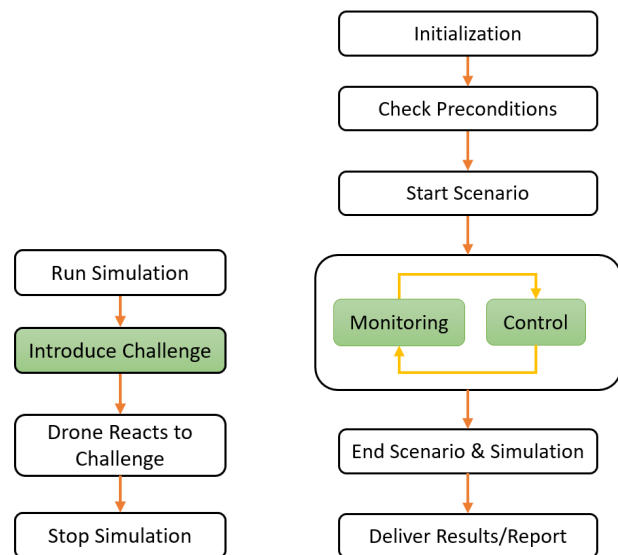


Fig. 2: The scenario elements are described in a scenario file format (XML and DSL). Next, the content is translated into a Behavior Tree (BT) for executing the scenario.



(a) Challenges can be introduced in scenarios and pose an unexpected event to the drone.

(b) The lifeline of a scenario during its simulation. At the core of the simulator, an execution engine monitors and controls the active scenario elements.

Fig. 3: Concept for challenges and scenario control

during or before mission execution. In addition to the laid out general procedure for challenges (Figure 3a), a classification and collection of challenges for drones is in development.

The execution of the scenarios poses a second challenge in simulation-based testing. A number of existing simulation environments from research and open source communities provide a robust foundation for simulation. In the context of this research towards drone algorithm development, the simulator Gazebo plus the Robot Operating System (ROS) is selected as a first development platform.

From an architectural viewpoint, scenario execution is best separated into a *scenario runner* component. Some elements of the scenario, such as mission execution, can and should

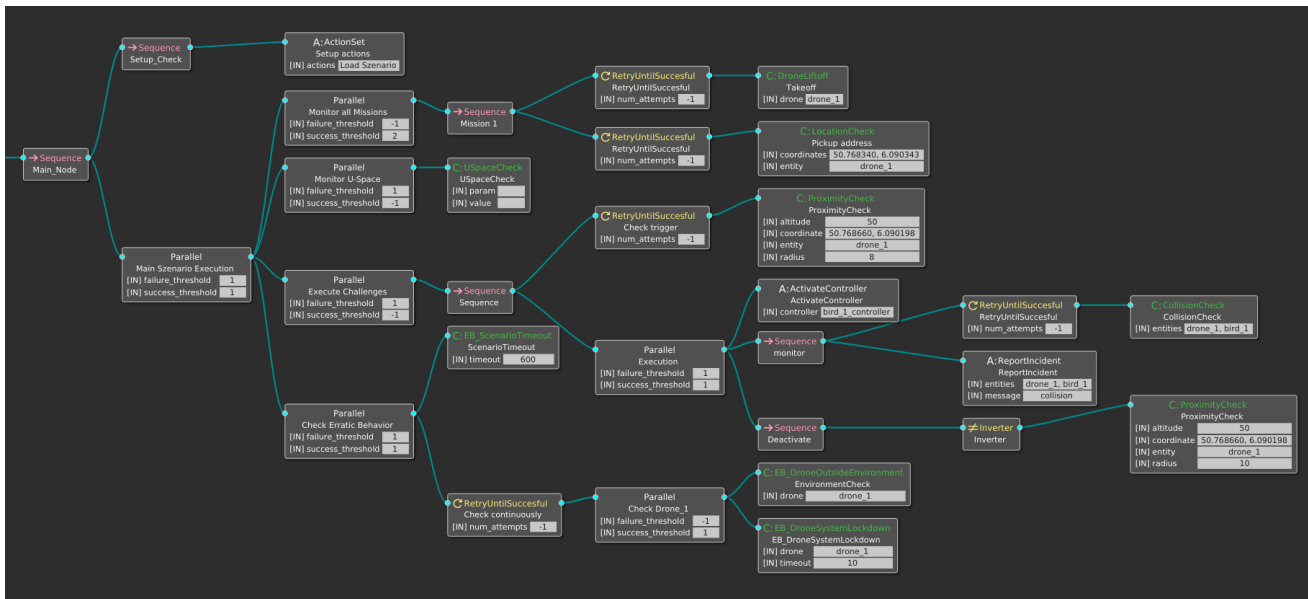


Fig. 4: Outlook on a behavior tree implementation to control the described scenario elements (shortened). Each branch on the 3rd level controls or monitors one category of scenario elements, while the last one checks for erratic behaviour (termination).

be observed passively, while others such as the challenges need an active intervention. This calls for a scenario interpreter component at the runner’s core, similar to the concept depicted in Figure 3b. The particular attention to comprehensibility, but also adaptability and finally portability to other simulation environments, makes it preferable to handle the core interpreter independent from the simulator platform. Figure 4 shows work in progress to describe this interpreter by means of a behavior tree. Behavior trees, with their design for decisions and events plus their availability in simulators, have the potential to guide scenarios in a platform-independent way, but more research is needed in this area. The excerpt shows some part of the main tree to observe the scenario elements. Similar considerations have to be made for the monitoring component, which forms the basis for reporting and evaluation.

The upcoming work will focus on a evaluable scenario format description, a scenario editor, and further aspects of the scenario runner integration and monitoring component.

IV. CONCLUSION AND FUTURE WORK

The presented work in progress investigates how simulation scenarios for virtual testing of service drones can be described and implemented. The research can build on similar work from the automotive sector and aviation scenario studies, but for drone services, there are specific requirements in scenario design and simulation. The main focus here lies on demonstrating drone conformity and capability, which is addressed with a direct integration of U-Space and service missions. Early results on a scenario description model and methods for scenario execution are presented with figures in the poster.

Future work will include a definite scenario format proposal. The next step is to support simulator execution with translation

rules for the behavior tree. Tool support, like a scenario editor, will be necessary to establish a repository of scenarios and challenges to choose from. The plan is then to use the developed scenarios to investigate verification and validation methods for drones in an UTM setting.

ACKNOWLEDGMENT

This work is supported by ECSEL Joint Undertaking (JU) through the Project ADACORSA under grant agreement No 876019. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and Germany, Netherlands, Austria, Romania, France, Sweden, Cyprus, Greece, Lithuania, Portugal, Italy, Finland, Turkey.


We thank our colleague Julia Schaumeier for her early contributions.

REFERENCES


- [1] S. Kalisvaart, Z. Slavik, and O. Op den Camp, “Using Scenarios in Safety Validation of Automated Systems,” in *Validation and verification of automated systems*: Springer, 2020, pp. 27–44.
- [2] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, “Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving,” in *2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*: Spain, 2015, pp. 982–988.
- [3] T. Menzel, G. Bagschik, and M. Maurer, “Scenarios for Development, Test and Validation of Automated Vehicles,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*: Changshu, 2018, pp. 1821–1827.
- [4] S. Jafer, B. Chhaya, U. Durak, and T. Gerlach, “Formal Scenario Definition Language for Aviation: Aircraft Landing Case Study,” in *AIAA Modeling and Simulation Technologies Conference*, Washington, D.C., 2016, pp. 2016–3521.
- [5] ASAM e. V., *OpenSCENARIO*. [Online]. Available: <https://www.asam.net/standards/detail/openscenario/> (accessed: April. 4 2022).
- [6] J. A. Millan-Romera, J. J. Acevedo, A. R. Castano, H. Perez-Leon, C. Capitan, and A. Ollero, “A UTM simulator based on ROS and Gazebo,” in *The 2019 International Workshop on Research, Education and Development on Unmanned Aerial Systems (RED-UAS 2019)*: Cranfield, United Kingdom, 2019, pp. 132–141.

Challenges for Periodic Technical Inspections of Intelligent Cars

Mona Gierl

Institute of Energy Efficient Mobility
University of Applied Sciences
Karlsruhe, Germany
mona.gierl@h-ka.de 

Felix Müller

Institute of Energy Efficient Mobility
University of Applied Sciences
Karlsruhe, Germany
felix.mueller@h-ka.de 

Reiner Kriesten

Institute of Energy Efficient Mobility
University of Applied Sciences
Karlsruhe, Germany
reiner.kriesten@h-ka.de

Philipp Nenninger

Institute of Energy Efficient Mobility
University of Applied Sciences
Karlsruhe, Germany
philipp.nenninger@h-ka.de

Eric Sax

Institute for Information Processing Technologies
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
eric.sax@kit.edu

Abstract—The periodic technical inspection is a regulatory measure to ensure road safety and environmental sustainability during the operation of vehicles. It contains a non-destructive visual and impact assessment of its systems and components. With the advancement of autonomous and connected cars, new technologies, growing number of sensors, and new electrical/electronic-architecture designs find their way into the vehicle, which implies new challenges for the evaluation of road safety and environmental sustainability. In this paper, the need for advanced inspection methods due to upcoming new technologies enabling autonomous driving is investigated. A brief background about ongoing research and regulations addressing the verification and validation of autonomous and connected cars is given. The current procedure of periodic technical inspections in Germany is summarized and prospect challenges - addressing both, advancing technologies for autonomous vehicles and cyber security considerations of connected cars - are identified. Based on the listed challenges, possible improvements are derived, which should serve as a reference work to upcoming discussion about the extent of Periodic Technical Inspections (PTI) for autonomous cars.

Keywords—periodic technical inspection, security, autonomous driving, homologation.

I. INTRODUCTION

As of today, human fault is still the main reason for accidents [1], whereas the advances in technology enable enhanced safety features leading to autonomous, connected vehicles. With the introduction and application of Advanced Driver Assistance Systems (ADAS) and connectivity features (as Car2X) the automotive industry provides intelligent vehicles as a solution for improved road safety.

Prospective vehicles are expected to have 20 times more computational power [2] and to be running on 100 million lines of code [3]. Thus, the technical advances come with an increase in sensor systems to reconstruct the surroundings and a growing number of software solutions which require a higher amount of data and computational effort. One side effect is the growing complexity which might lead to additional

unwanted technical errors. Thus, it is common consensus to apply functional safety and cyber security standards during the development as well as testing throughout the development process and afterwards.

Beside verification and validation activities during development by the Original Equipment Manufacturers (OEMs), the vehicle has to be approved by an accredited authority to get road admission. This allows for an independent analysis on the car's roadworthiness and environmental sustainability across various types and models. Further, road admission depends on the condition of the vehicle which is regularly checked through mandatory periodic technical inspections which, e.g., occur every 2-3 years for passenger cars in Germany [4].

a) Problem statement: Mandatory technical inspections review the roadworthiness of vehicles and probe compliance with national environmental sustainability regulations. Regulatory standards (e.g., Regulation (EU) 2018/858 [7], Directive 2014/45/EU [8], etc.) prescribe a minimum set of required test procedures to show compliance to these regulations. With the advance of autonomous vehicles, a growing number of electronic systems (cameras, RADAR, LIDAR, etc.) are added as common equipment and enable the car to drive autonomously which simultaneously leads to a higher number of safety relevant systems. Consequently, an adaptation from the current mandatory test procedures is required.

b) Contribution: In this paper, current efforts to establish new test procedures for technical inspections are briefly highlighted and upcoming challenges due to the advances of intelligent vehicles are presented. In addition, current test procedures of passenger cars in Germany are summarized and potential improvements for periodic technical inspections based on the listed challenges are elaborated.

c) Classification of driving automation: In the field of autonomous driving, the SAE J3016 Standard defines six levels of automation [5]:

- Level 0 - No automation

- Level 1 - Driver assistance
- Level 2 - Partial automation
- Level 3 - Conditional automation
- Level 4 - High automation
- Level 5 - Full automation

From Levels 0-2 the driver is considered as driving but might be supported by assistance features whereas from Levels 3-5 the driver is not considered to drive even if seated in the “driver’s seat”. These levels are important for the classification within this work as Level 3 or higher levels of driving autonomy features are considered to challenge future technical inspections.

d) Paper structure: Section II provides the background on current verification and validation efforts for autonomous driving functions. Two essential efforts are highlighted, namely the PEGASUS project [6] and the United Nations Regulation UN R155 concerning the approval of vehicles with regards to cyber security [9]. Afterwards, the Periodic Technical Inspection (PTI) in Germany is presented to elaborate on the current mandatory road inspection methods. Based on the legal framework, Section IV derives upcoming challenges for technical inspections. To address the challenges, potential improvements are presented in Section V.

II. BACKGROUND

A first effort towards new verification and validation methods for autonomous driving functions has been made by the PEGASUS Project, which was concluded in 2019 [6]. It is a “Project for the Establishment of Generally Accepted quality criteria, tools and methods as well as Scenarios and Situations for the release of highly-automated driving functions” [6].

A second effort is currently made by the United Nations Economic Commission for Europe (UNECE). It developed “two new UN Regulations on Cybersecurity and Software Updates [...] which are the first ever internationally harmonized and binding norms in this area” [10].

A. PEGASUS

Pioneering the efforts in Germany to create a standard to test autonomous driving functions, in order to clear them for use in production vehicles on public roads, PEGASUS project created a method to assess the Level 3 function “Highway Pilot”. The idea was to define a process that can be used to validate such a system in order to green-light its use on public roads. Instead of driving thousands of kilometers on the roads (distance-based validation), a scenario-based testing approach is presented, which enables a systematic validation of the automated driving function. The result of this project was a possible approach consisting of the following five steps [6]:

- Definition of requirements
- Data processing
- Information storage and processing in a database
- Assessment of the highly automated driving function
- Argumentation

Starting with a collection of all the information available, the PEGASUS Method aims to define logical scenarios and reuses recorded test drives to create a pool of relevant scenarios to test the function. In parallel, the requirements to assess the driving function are defined.

In succession to these two steps, all the gathered information is transferred into databases, where the data can be accessed and augmented with information gathered in the later stages. Based on the scenarios, the parameters for the different test runs are generated as well as the corresponding pass / fail criteria.

After these steps, tests of the driving function are performed and evaluated in order to allow for the creation of a risk assessment. These tests can be performed in simulation, driving on proving grounds and in real traffic, depending on the concrete test. As a final step, the results are compared to the predefined safety argumentation and can be reused for the next test iteration. Based on the results of PEGASUS, new projects are underway, to use the results for the development of procedures for systems of Level 4 and 5.

B. Security Regulations for Type Approval

In 2021, the UNECE WP.29 working party published the regulation text “Uniform provisions concerning the approval of vehicles with regards to cyber security and cyber management system” (UN R155) [9], which is planned to be mandatory for all new vehicle types within the European Union as of July 2022 [10]. The UNECE WP.29 working party for Automated/Autonomous and Connected Vehicles (GRVA) is responsible for the harmonization of vehicle regulations and addresses autonomous and connected vehicles. The proposal is the first regulatory step demanding the integration of security processes and measures during the development of vehicles. As of 2022, there are currently 64 contracting parties including the European Union and others [11].

At the time the regulation takes effect, type approval with regard to cyber security is only granted to vehicle types that satisfy the requirements of the UN R155 regulation. The regulation differentiates between the responsibility of the manufacturer to implement a cyber security management system and the requirements for the approval of a vehicle type. Thus, according to the regulation text [9], the vehicle manufacturer shall provide evidence for:

- Requirements for the Cyber Security Management System (CSMS)
 - CSMS shall be applicable to the development, production and post-production phase
 - demonstrate processes to adequately identify and manage cybersecurity related risks
 - implement incident response capabilities within “a reasonable timeframe”
 - identify and manage supplier-related risks
- Requirements for vehicle types
 - the vehicle manufacturer shall evidence a compliant CSMS

- identify and manage supplier-related risks
- identify critical elements of the vehicle type by performing exhaustive risk assessment
- manage identified risks with proportionate countermeasures
- verify effectiveness of the security measures by performing appropriate and sufficient testing
- provide monitoring and forensic capabilities to enable attack analyses

The exact implementation measures are not defined as these are specified by each manufacturer, ideally through applying relevant standards (e.g., ISO/SAE 21434). Thus, a variety of different security measures is expected to be implemented for next generation vehicles which have to meet the cyber security regulation's requirements. From the perspective of approval authorities, new challenges arise as to define the evidence and test scenarios which prove roadworthiness to grant type approval.

Both projects, PEGASUS and the UN R155 security regulation, indicate significant effort being made to develop advanced test methods for autonomous and connected vehicles and show how current these topics are for the automotive industry. Thus, in our understanding, these efforts are the first steps done to advance test methods for type approvals of intelligent vehicles, however, further research is certainly required. Further, regular technical inspections are also challenged by the introduction of intelligent vehicles, but currently not yet addressed in funded research projects or within regulatory initiatives. Hence, as a first step to also address PTI, in the subsequent sections the present PTI procedure is presented and prospective challenges are outlined.

III. PERIODIC TECHNICAL INSPECTION IN GERMANY

In Germany, as in many other countries, it is mandatory to have your vehicle inspected at regular intervals to ensure its roadworthiness [4, Anlage VIII, 1.2.1]. Depending on the vehicle and its use, these intervals vary. For the purpose of this paper, a standard passenger car is assumed. The work is supposed to include other types of vehicles, but all examples will focus on passenger cars. Here, the usual interval is set to once every two years, after an initial period of three years for new cars, starting with the day of the first registration [4, Anlage VIII, 2.1.2.1.1].

A. Extent

To give a report on the roadworthiness, the technical inspection covers different characteristics of the presented vehicle. As stated in [4, Anlage VIIIa, 6], these include:

- Braking equipment
- Steering
- Visibility
- Photometric equipment and other parts of the electric installation
- Axles, wheels, tires, suspension
- Chassis, frame, platform, attached parts
- Other equipment

- Environmental impact
- Identification and classification of the vehicle

The focus is therefore placed mostly on the mechanical state of the vehicle. These parts are to be inspected visually, through a functional and performance evaluation of the vehicle and its parts, as well as by the reaction of the car and its systems to an action performed by the inspector [12, pp. 75]. The inspection is set up deliberately as described, to allow for the inspection to be performed in a similar manner across different makes and models of vehicles and to not depend on the specific functional implementation of a specific vehicle, but to also assess the performance of certain components or systems.

B. Procedure

The inspection itself does currently consist of multiple parts:

Registration for Inspection:

The Registration for Inspection is the first step, so that the car and its specific testcases are known to the person conducting the inspection.

Test Drive:

A short test drive is performed, in order to ensure that all control units in the car are booted up and operational.

Emissions Test:

For cars with internal combustion engine, the emissions of the car are to be checked before or during the periodic technical inspection, to ensure they are within an accepted range.

Brake Test:

During the brake test, a series of measurements are taken to ensure that the brakes are performing within the expected limits [4, Anlage VIIIa, 4.4]. Multiple ways are available to take these measurements.

Further inspection:

Inspection regarding the composition, condition, function and effect of its components and systems

During the subsequent vehicle inspection, the car is checked visually, manually and electronically while sitting on the shop floor and while lifted up [4, Anlage VIIIa, 4.3].

C. Results

As the ideal result for a PTI, a car passes all tests and is good to continue driving on public roads for the next two years. Having only minor defects (e.g., defective bulbs or scratched exterior mirrors), it is possible to allow the vehicle back on the road with the requirement to have them fixed as soon as possible. If there is one or more major or dangerous defects (e.g., impacting the brake functionality), the car has to be repaired and presented again. If the car is deemed a hazard on the road, the car can be decommissioned. In this most severe case, the car cannot be legally driven on the road. The results of each technical inspection is communicated to a centralized institution, the so called "Zentrale Stelle (FSD)" [13], to be aggregated and evaluated.

TABLE I

ESTIMATION OF THE DEVELOPMENT OF THE AUTOMOTIVE MARKET IN REGARDS TO THE DISTRIBUTION OF CARS SHOWN IN MILLION UNITS, GROUPED BY THE CAPABILITIES ACCORDING TO THE SAE-LEVEL [14].

Year	SAE Level						Robot vehicles
	0	1	2	3	4	5	
2015	59.3	24	8.2	0	0	0	0
2016	59.1	26.9	9.8	0	0	0	0
2017	58.2	28.4	10.9	0	0	0	0
2018	55.7	29.9	11.9	0.1	0	0	0
2019	51.8	33.1	13.4	0.6	0	0	0
2020	49	35.4	15.1	1.1	0	0	0
2021	46.5	37.9	16.5	1.7	0	0	0
2022	42.8	40.2	18.5	2.6	0	0	0.1
2023	41.1	42.5	20.8	2.7	0	0	0.1
2024	37.7	45.3	22.8	5.8	0	0	0.1
2025	35.3	47.4	25.2	7.6	0.1	0	0.2

IV. CHALLENGES FOR TECHNICAL INSPECTIONS

According to estimations of [14], the number of autonomous driving cars is expected to grow and first Level 4 cars are predicted for 2025 to be found on the road. Prospective vehicles are announced to have 20 times more computational power [2], whereas automotive software and sensors are expected to exhibit a 9 % for the software segment and 8 % for the sensors segment compound annual growth rate (CAGR) between 2020 and 2030 [15]. Further, [15] estimate an overall market size of USD 84 billion by 2030 for software development including OS, middleware, functional domains (powertrain, chassis, energy, body, etc.), connectivity and security. Considering these estimations, the following major challenges to impact future PTI were identified:

- Challenge I The condition of vehicle sensors is essential for autonomous driving, thus new test scenarios are required to test the growing number of safety-relevant sensors.
- Challenge II Growing complexity of data to be processed and software which is prone to unintended technical faults.
- Challenge III Vehicle data might not be accessible but is essential for demonstration of roadworthiness.
- Challenge IV Security measures require validation methods that enable inspection engineers to evaluate the roadworthiness.
- Challenge V The composition of software and hardware determines the correct operation of the vehicle system, thus the detection of any unauthorized modifications (e.g., firmware alteration, etc.) is necessary.

V. IMPROVEMENTS FOR THE PERIODIC TECHNICAL INSPECTION (PTI)

With regard to new features in cars, specifically regarding connectivity and autonomous driving, the PTI has to keep up in order to fulfill its intended role to ensure the safety of the

different road users from a technical standpoint. Therefore, the subsequent ideas are proposed as an addition to the PTI to also address security and autonomous driving capabilities.

According to [4, Anlage VIIIa, 4.], the vehicle components and systems shall be examined for their composition, condition, function and effectiveness. As a result, Table II summarizes the proposed improvements based on the defined regulation’s categorization.

A. To Inspect the Security of Vehicles

Security has the special characteristic that if it is running correctly, it should not be recognizable and it should not affect the driving functionalities of the car. Yet, the software also displays “aging effects” due to constantly new evolving attack methods that might allow bypassing implemented security measures. These aging effects are not identifiable by visual examination as it might be the case for mechanical components (e.g., braking pads, etc.). Instead, a regular threat and risk analysis for deployed vehicles as presented in [16] should be considered and could help to analyze the security condition of the vehicle.

Another challenge is the variety of integrated security measures - as the ISO/SAE 21434 aims to provide a security framework to facilitate security by design, it does not provide technology specific solutions. Each manufacturer has to integrate effective measures to protect their critical systems and functions adequately [10]. To prove effectiveness, security testing methods exist that aim to demonstrate both: a) the correct functioning of integrated security measures and b) a low risk for unintended or undefined system states that might provoke misbehavior. However, applying these test methods after the development phase, especially penetration testing, is not desired by manufacturers as the car is not able to differentiate between a hacker and a penetration tester. As a reaction the car might lock down affected electronic control units to protect its assets. To counter this worst-case, but to also be able to inspect the correct functioning and effect of the security measures for deployed vehicles, the approval authorities and OEMs should hold a dialogue on how to enable security testing techniques in a controlled environment including the PTI.

Lastly, lessons learned from the Information Technology (IT) domain show that securing assets is a race between attackers and security engineers, and that vulnerabilities or known attacks are valuable insights to improve the systems security. For this reason, a collection of all known vulnerabilities, similar to the Common Vulnerabilities and Exposures (CVE) database [17], would be beneficial for the automotive industry. First efforts are made by [18]–[20] but the main hindrance is still the strong competition between players in the automotive industry.

B. To Inspect the Operation of the Autonomous Driving Capabilities

In order to ensure the correct function of the autonomous driving system, different parts of the vehicle have to be

TABLE II

OVERVIEW OF THE CURRENT STATE OF THE PERIODIC TECHNICAL INSPECTION IN GERMANY AND THE PROPOSALS TO INCLUDE AN ASSESSMENT REGARDING THE SECURITY OF THE VEHICLE AND ITS AUTONOMOUS DRIVING FUNCTIONS.

Context	Composition	Condition	Function	Effectiveness
Current State	Assessment and identification of built-in parts and components	Assessment of wear and tear (aging, damage, corrosion, etc.)	Actuation of control devices (pedals, levers, switches, etc.) to assess whether the operation is correct in terms of time and function.	Measurement of a component or system for compliance with specified limit values
Security	Check hardware and software to correspond to the specifications of the OEM	Vulnerability analysis and/or threat and risk analysis to identify deprecated/missing security measures	Security testing and reading self-diagnosis results through OBD to monitor correct functioning	CSMS assessment and detection of unintended behavior
Autonomous Driving	Check hardware and software to correspond to the specifications of the OEM	Assessment of the current state of the sensor and actor systems required to perform the driving functions	On-Board-Diagnostic and inspection of the associated functions	Assessment of the performance of the sensors and actors required to perform the driving functions

checked. These include:

- the sensors detecting the surroundings,
- the actuators performing the driving function,
- the control units running the associated software and
- the software to perform the driving function itself.

Different parts of the driving system are subject to different kinds of problems during the daily use of the accordingly equipped vehicles. The hardware, for example, is aging from the moment the system is produced and the vehicle is leaving the assembly line. Therefore, the biggest differences between otherwise identical vehicles is the wear and tear the vehicle has been subjected to during its lifetime. Accidents, not properly performed repairs and just general misalignment can cause the autonomous driving system to malfunction. Therefore, it is proposed that a future PTI has to include a test of the sensor systems to ensure that they are detecting the surroundings, e.g., detecting the objects in the designated areas as well as locating them correctly. All major sensor systems used in the specific vehicle have to be inspected to ensure they are operating according their specifications.

In the same sense, the hardware to perform the driving function has to be inspected. This includes:

- steering,
- braking,
- acceleration and
- communication with other road users.

For the most part, the hardware of these systems is already part of the PTI today. The brakes, for example, are already tested for their performance (see Section III-B). In addition to these tests, triggering of these systems electronically via the driving function has to be tested, especially their capabilities to provide granular access to these functions.

The communication with other road users and infrastructure presents a special case. Nowadays, cars are getting more connected via Car2X communications aiming to enable the exchange of traffic information. In addition, the first visual and acoustic communication systems are getting mounted to luxury cars to enable an interaction with its surroundings [21]. These initiatives can also be seen as part of the autonomous driving systems and therefore have to be inspected during future PTI.

C. To Inspect the Security and the Autonomous Driving Capabilities

As software is essential for both fields of study, a process needs to be defined, which would allow a check for both research topics simultaneously, as that will streamline the inspection and helps to keep time and thus costs low. Therefore, the authors propose to implement a version control system that allows the verification of the installed software in the vehicle under test. Having this system in place allows checking for manipulation on the software side as well as for outdated software that might not include the latest traffic signs or rules and as a result would not be able to follow the latest traffic regulations. A first effort to identify type approval relevant software is done by [22].

As the software can nowadays massively alter the behavior of the car on the road, see for example [23], it is a necessity to have a suite of tests similar to the type approval of such a driver assistance system, to ensure correct functionality after the update has been installed. Such a system should be put in place for all systems which can alter the movement characteristics of the car under test. Having passed such a test scenario, key parameters, such as version identifier and checksums can be recorded in order to be able to match the installed software in the car with the list of approved versions.

VI. CONCLUSION

As of today, present regulations are adapted to enable autonomous cars to get road admission. The advancing technologies challenge these efforts which is the reason for research projects and developed amendments to existing regulations (see Section II). However, these efforts should also address PTI as these are responsible to regularly probe compliance over the life cycle of the vehicle until its decommission. The applied test procedures during a PTI have to adapt to upcoming technologies. Thus, new test methods have to be defined since an increase in autonomy leads to a growing number of safety relevant driving systems that have to be examined for possible technical failures due to wear, tear or tuning. The associated upcoming challenges were identified by the authors and listed in Section IV. Based on these challenges, improvements to address the safety and security of autonomous cars are proposed in Table II. This table was designed to match the current regulatory framework in Germany and elaborates improvements based on the categories: composition, condition, function and effectiveness. The provided information should serve as reference work to upcoming research and discussions about the extent of PTI. Thus, within future work, it is planned to identify evaluation methods for inspection engineers that enable an assessment of the roadworthiness of vehicles throughout their time of road admission. To address the challenges of intelligent vehicles, both advanced sensor systems and continuous security activities need to be considered.

ACKNOWLEDGEMENT

The work presented in this paper was funded by GTÜ Gesellschaft für technische Überwachung mbH in Stuttgart, Germany.

REFERENCES

- [1] Statistisches Bundesamt, "Fehlverhalten der Fahrzeugführer bei Unfällen mit Personenschaden" (Misconduct of drivers in accidents with personal injury), July 2020, online, <https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Verkehrsunfaelle/Tabellen/fehlverhalten-fahrzeugfuehrer.html>, retrieved: April 2022.
- [2] Automotive World, "The new BMW iX xDrive40 and new BMW iX xDrive50" , March 2021, online, <https://www.automotiveworld.com/news-releases/the-new-bmw-ix-xdrive40-and-new-bmw-ix-xdrive50/>, retrieved: April 2022.
- [3] R. Charette, "This Car Runs on Code", 2009, online, <https://spectrum.ieee.org/transportation/systems/this-car-runs-on-code>, retrieved: September 2020.
- [4] 2. Straßenverkehrs-Zulassungs-Ordnung (2nd road traffic licensing regulations), Version including revisions until 18 May 2017.
- [5] SAE International, "SAE J3016 - Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles", 2018
- [6] PEGASUS Project, "Pegasus Method An Overview", online, <https://www.pegasusprojekt.de/files/tmpl/Pegasus-Abschlussveranstaltung/PEGASUS-Gesamtmethode.pdf>, retrieved: March 2022.
- [7] European Parliament and European Council, "Regulation (EU) 2018/858 of the European Parliament and of the Council of 30 May 2018 on the approval and market surveillance of motor vehicles and their trailers, and of systems, components and separate technical units intended for such vehicles, amending Regulations (EC) No 715/2007 and (EC) No 595/2009 and repealing Directive 2007/46/EC", 2018, online, <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32018R0858>, retrieved: January 2022
- [8] European Parliament and European Council, "Directive 2014/45/EU of the European Parliament and of the Council of 3 April 2014 on periodic roadworthiness tests for motor vehicles and their trailers and repealing Directive 2009/40/EC Text with EEA relevance", 2014, online, <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32014L0045>, retrieved: January 2022
- [9] UNECE, "Uniform provisions concerning the approval of vehicles with regards to cyber security and cyber security management system", March 2021, Revision 3, online, <https://unece.org/sites/default/files/2021-03/R155e.pdf>, retrieved: April 2022.
- [10] UNECE, "UN Regulation on Cybersecurity and Software Updates to pave the way for mass roll out of connected vehicles", June 2020, online, <http://www.unece.org/?id=54667>, retrieved: April 2022.
- [11] UNECE, "Status of the 1958 Agreement (and of the annexed regulations)", online, <https://unece.org/status-1958-agreement-and-annexed-regulations>, retrieved: January 2022.
- [12] H. Braun, J. Bönninger, S. Missbach, and R. Süßbier: "Erkennen und Bewerten von Mängeln an elektronischen Systemen und Bauteilen im Kraftfahrzeug" (Detection and evaluation of defects in electronic systems and components in motor vehicles) Kirschbaum, Bonn, 2015. – ISBN 978–3–7812–1920–5
- [13] FSD, "Die FSD - Zentrale Stelle stellt sich vor" (The FSD - Central Office introduces itself), online, <https://fsd-web.de/>, retrieved: April 2022.
- [14] C. Malaquin: "Towards ADAS to Imaging radar for automotive market and technology trends", Microwave & RF Conference 2019.
- [15] McKinsey & Company, "Automotive software and electronics 2030 - Mapping the sector's future landscape", 2019, online, <https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/mapping-the-automotive-software-and-electronics-landscape-through-2030>, retrieved: March 2022.
- [16] M. Gierl, R. Kriesten, P. Neugebauer, and E. Sax, "Reverse Threat Modeling: A Systematic Threat Identification Method for Deployed Vehicles", 18th International Conference on Scientific Computing (CSC 2020), Las Vegas, USA, 2020
- [17] MITRE Corporation, "CVE Security Vulnerability Database", online, <https://cve.mitre.org/>, retrieved: January 2022.
- [18] AUTO-ISAC, "Automotive Information Sharing and Analysis Center", online, <https://automotiveisac.com/>, retrieved: April 2022.
- [19] Upstream Security Ltd., "Autothreat Intelligence Cyber Incident Repository", online, <https://www.upstream.auto/research/automotive-cybersecurity/>, retrieved: March 2022.
- [20] M. Ring, J. Dürrwang, F. Sommer, and R. Kriesten, "Survey on vehicular attacks - building a vulnerability database", 2015 IEEE International Conference on Vehicular Electronics and Safety (ICVES), Yokohama, 2015, pp. 208-212, doi: 10.1109/ICVES.2015.7396919
- [21] K. Groeneveld, "The F 015 Luxury in Motion at the Ars Electronica Festival in Linz: Creative break on the journey to the future", September 2015, online, <https://group-media.mercedes-benz.com/marsMediaSite/ko/en/9919078>, retrieved: April 2022.
- [22] UNECE, "Uniform provisions concerning the approval of vehicles with regards to software update and software updates management system", March 2021, online, <https://unece.org/sites/default/files/2021-03/R156e.pdf>, retrieved: April 2022.
- [23] C. Dhabhar, "Tesla Model 3 Braking Improves With On-Air Software Update", May 2018, online, <https://www.carandbike.com/news/tesla-model-3-braking-improves-with-on-air-update-1859322>, retrieved: April 2022.