Received 2 June 2023, accepted 23 June 2023, date of publication 3 July 2023, date of current version 10 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3291350

RESEARCH ARTICLE

Trajectory Planning of Automated Vehicles Using Real-Time Map Updates

MÁTYÁS SZÁNTÓ[1], CARLOS HIDALGO[2,3], LEONARDO GONZÁLEZ[2], JOSHUÉ PÉREZ RASTELLI[2], (Member, IEEE), ESTIBALIZ ASUA[3], AND LÁSZLÓ VAJTA[1]
[1]Department of Control Engineering and Information Technology, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, 1111 Budapest, Hungary
[2]Industry and Mobility Area, Basque Research and Technology Alliance, Tecnalia, Derio, 48160 Biscay, Spain
[3]Department of Electricity and Electronics, University of the Basque Country, Leioa, 48940 Biscay, Spain

Corresponding author: Mátyás Szántó (mszanto@iit.bme.hu)

This work was supported by the European Commission Horizon 2020 (H2020) Program under the Security By Design IoT Development and Certificate Framework with Front-end Access Control (IoTAC) Research and Innovation Action under Grant 952684.

ABSTRACT The development of connected and automated vehicles (CAVs) presents a great opportunity to extend the current range of vehicle vision, by gathering information outside of its sensors. Two main sources could be aggregated for this extended perception; vehicles making use of vehicle-to-vehicle communication (V2V), and infrastructure using vehicle-to-infrastructure communication (V2I). In this paper, we focus on the infrastructure side and make the case for low-latency obstacle mapping using V2I communication. A map management framework is proposed, which allows vehicles to broadcast and subscribe to traffic information-related messages using the Message Queuing Telemetry Transport (MQTT) protocol. This framework makes use of our novel candidate/employed map (C/EM) model for the real-time updating of obstacles broadcast by individual vehicles. This solution has been implemented and tested using a scenario that contains real and simulated CAVs tasked with doing lane change and braking maneuvers. As a result, the simulated vehicle can optimize its trajectory planning based on information which could not be observed by its sensor suite but is instead received from the presented map-management module, while remaining capable of performing the maneuvers in an automated manner.

INDEX TERMS Connected and automated vehicles (CAVs), dynamic obstacle mapping, external perception, real-time trajectory planning, object avoidance, vehicle-to-network communication (V2N).

I. INTRODUCTION

Connected and Automated Vehicles (CAVs) have been met with an increasing amount of interest in the research of automated vehicles (AVs) lately. Their ability to communicate allows them to become more robust towards hard problems faced by AVs, where a correct assessment of traffic situations is safety critical, such as navigation in congested areas, interactions with emergency vehicles, accidents in the road, among others. This resilience is – amongst other causes – the result of the aggregating information supplied by all vehicles. Such swarm-like networks have shown strong efficiency for problem solving in traffic automation [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Eyuphan Bulut.

The cornerstone of a functional and effective information sharing framework between individual CAVs is a robust communication network. In the automotive world, these technologies have been categorized depending on the agents involved; Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Network (V2N) communication have all been employed to enhance vehicle capabilities and to improve safety in the road. These technologies have been focused on the development of standardized services, and generally make use of specialized hardware. Alongside these developments, vehicles with their complex networks, increasing connectivity, sensors, and actuators, could also be considered, Internet-of-Things (IoT) entities [2].

IoT technologies and services have been on the rise in recent years, and a key solution for these has been the development of low-latency, low-overhead communication schemes and wide internet coverage focusing on connected homes, connected vehicles and smart cities. Among the established solutions available for IoT communication, MQTT is one of the most widely deployed protocols. MQTT provides a network for message delivery between information sources – i.e., publishers –, and users – i.e., subscribers – through a broker entity. In the case of CAVs, this channel of communication can be leveraged to exchange telemetry information about the vehicle or driver, to monitor fleets of vehicles in the case of vehicle renting, to exchange high level control inputs such as changing automated system parameters remotely, or to provide channels for cooperation/perception in leveraging computational load to remote servers; as the one proposed in the below paper [3].

One particularly important bundle of information that CAVs must receive is a map of their surroundings, which can be used for locating themselves and navigating. Use-cases for maps include the planning of trajectories prior to departure and during the maneuvering of these vehicles in complex environments, which often enclose other vehicles, pedestrians, and diverse obstacles. Currently, the way AVs navigate in their environments is mostly performed by internally adding real-time information to static maps – as the name suggests, these are not kept up-to-date and have low refresh frequencies. The mapping process is usually characterized by latencies in the order of years, primarily because of its costliness and resource-demanding nature. The dynamic component of the internal maps of the vehicles is therefore concatenated on-the-fly using their onboard sensors (LIDARs, cameras, etc.) [4].

In recent years, a compelling trend appeared in mapping for automated vehicle development; Namely, the use of crowdsourced data acquisition solutions. This paradigm provides a solution for high-frequency database updating. The process involves numerous data sources, which can provide raw environmental information for map construction [5]. This solution, however, has only seen scrutiny in the realm of non-connected automated agents. We argue that having access to frequently updated maps containing the real-time location of obstacles and other items enables more robust and substantially more effective trajectory planning for automated vehicles. In addition, crowdsourced data contribution facilitates the functionality of a mapping framework, which is able to provide the required information for connected vehicles in real time during maneuvering.

The intersection of the technologies previously outlined represents an opportunity to deploy a mapping solution into an IoT network for vehicles. The benefits of and ease of deployment of an IoT network, make it a real alternative to leverage the safety improvements of V2X communication. In addition to this, crowdsourced mapping is a great fit for these type of networks, leveraging information from all agents involved in the communication and improving reliability of the information provided, which in turn is vital for vehicle assessment of the road situation and decision making.

This paper proposes a novel candidate/employed map (C/EM) management module, that calculates map updates (such as obstacles, vehicles, etc) and shares them through an MQTT network. To do so, this method uses real-time data from physical and simulated CAVs through an MQTT network, enabling a crowdsourcing-like framework. As a result, our solution provides a real-time updated map to CAVs, which is utilized in their internal trajectory planner for navigation. Finally, the individual vehicles are enabled to update their trajectories and switch between maneuvers based on the real-time-updated central maps on the MQTT-shared data. This framework was tested with lane change maneuvers specifically, however, it can be easily scaled and extrapolated to other maneuvers.

Our contributions in this article are:

- We introduce a novel candidate/employed map management solution that enables dynamic mapping of obstacles in a central database based on information supplied by CAV sensors.
- We show that the presented solution makes use of V2N communication over the MQTT protocol providing a novel crowdmapping-based approach for data acquisition and low-latency map updating and publishing.
- We implement the real-time map management framework and integrate it with the trajectory optimization and maneuver planning control units of CAVs.

This article is structured as follows: First, we present the relevant literature, followed by a description of the automated driving framework utilized in this work. We then introduce our novel framework enabling real-time map updates and then provide insights on its integration with the vehicle network. Finally, we present and evaluate the results of testing the functionality of the combined system and formulate our conclusions.

A. LITERATURE REVIEW

In this section, we are introducing the state-of-the-art in relation to the research and results presented in the paper. First, we introduce works referring to the main maneuver presented in the article, the lane change. Secondly, the different automated vehicle trajectory planning methods are reviewed and finally, the methodology of the map creation and updating techniques are summarized.

1) LANE CHANGE

Automated Lane Change maneuvers can be tackled in three steps [6]. The first one is in the decision-making aspect, where the environmental variables (vehicles, obstacles, road shape, etc.) are taken into account in order to decide whether the lane change can be executed or not. The second step is more closely related to the trajectory planning aspect, where the trajectory followed by the vehicle is planned ahead in order to carry out a safe, feasible lane change. The final step is associated with path tracking, which aims to follow a predefined trajectory. From these steps, the first two are the points of interest in this paper.

Regarding the decision-making aspect there are numerous works using methods such as Search Algorithm [7], Finite State Machine (FSM) [8], Deep Reinforcement Learning [9], Game Theory [10], among others. Whereas, for the second step there are works using B-splines-curves [11], Bézier curves [12], further described in the following sub-section. All these works assume vehicles are not connected, relying only on their onboard sensors. Meanwhile, works such as [13], [14], or [15] use communications to coordinate vehicles lane changes in order to optimize the throughput of the road.

2) TRAJECTORY PLANNING FOR AUTOMATED VEHICLES

Trajectory planning has been widely researched in the literature since the early stages of CAV development. It usually considers the dynamic and/or kinematic models of the vehicle to go from a starting position to a final one [16]. The most relevant techniques are listed below.

a: NUMERICAL OPTIMIZATION

Numerical optimization techniques aim to minimize or maximize a function subject to different constrained variables. In the case of trajectory planning, this method is used to smooth previously calculated trajectories while taking into account the vehicle's kinematic. The main approaches are: Function Optimization [17], [18], [19] and Model Predictive Methods [20], [21], [22]. However, these methods have high computational costs, due to the optimization process, that takes place at each motion state, and depend on global waypoints [16], [23].

b: GRAPH-BASED

Graph-based techniques search for the best paths between the car's current state and a goal state in a state space represented as a graph [16], [23]. These techniques discretize the search space imposing a graph on an occupancy grid map with centers of cells acting as neighbors in the search graph [23]. The Graph-Based techniques can be used in both Global Route Planning and Local Trajectory Planning. However, in this study, only the ones used in trajectory planning are considered, being: State Lattice [24], [25], [26], Elastic Band [27], [28], [29], and A-star [23]. However, these methods require a lot of memory and cause heavy computational costs, leading to low planning efficiency [30].

c: GEOMETRY-BASED

The geometry-based techniques interpolate a set of previously defined waypoints to build a smooth trajectory that considers the vehicle's kinematics and dynamics, alongside the passenger‘s comfort [16], [23]. The most common methods are: clothoid curves [31], [32] and Bézier curves [33], [34]. This method presents lower computational costs than the two mentioned before. However, the clothoid curves still have a significant computational cost due to the integration process, as opposed to the Bézier curves that have lower computational cost, because the curvature is defined by control points.

3) MAPS FOR AUTOMATED VEHICLE NAVIGATION

A high-resolution map of the environment is required by most vehicles with higher levels of autonomy in order to ensure their safe and successful operation. In the literature, the map layers that can be used for this purpose are listed as follows [35]:

- Road graph layer, which contains basic structural information about roads – e.g., number of lanes, intersections, and road segments.
- Lane-level maps containing information on lane markings and road segments with sub-centimeter accuracy. These maps are mostly based on detailed information collected from roads by road-infrastructure maintenance organizations.
- High Definition (HD) maps containing high-resolution features for road identification: Geometric spatial maps with labeled elements of road infrastructure, built-up of grouped and labeled points in 3D space. The raw data used in this case are unstructured point clouds, collected predominantly by costly active sensors – e.g., Light detection and ranging units (LIDARs) – that are then processed and labeled.

Methods that enable the creation and management of such maps are introduced in the following section. For clarity, these methods are also listed in Table 1.

a: CROWDSOURCED MAPS

Crowdsourcing is a social computing paradigm that appeared during the late 2000s. It is defined as a method for solving tasks, which require many resources to be used that are not necessary to be in a geographically or temporally joined environment [36].

Gathering data for map-related usage has also seen scrutiny in the literature; using crowdsourced data for achieving this goal is often referred to as Volunteered Geographical Information (VGI) [37], [38]. Arguably, the most widely spread VGI platform is OpenStreetMap (OSM) [39], which is an online platform that became an industry standard for VGI schemes in the past decade.

Crowdmapping, a subcategory of VGI applications, has appeared in the technical literature in the last few years and has shown great usability for traffic developments, such as passenger or freight vehicle automation. For self-driving vehicles, it is crucial to sense and interpret their surroundings correctly, in order to recognize their pose – i.e., location and orientation – and the traffic situation in their environment. As a result, the use of crowdmapping has appeared in several research endeavors [5], [40], [41] as well as industrial applications (Waymo [42], Lyft [43], Uber [44], and Google [45]).

b: MAP UPDATING FOR LOW-LATENCY MAPS

One key problem with maps used for automated vehicle control, and the decision-making algorithms contained therein, is the high latency with regard to the updates these maps receive [46]. The navigation systems of current AVs augment the built-in static maps with real-time sensed obstacles from their environment during movement [23]. As a possible solution, CAVs have been developed to be able to share their individual status with each other. A number of works have been published in the literature that tackle the problem of vehicle-to-infrastructure (V2I) or vehicle-to-vehicle (V2V) communication transfer times [47], [48] to optimize data sharing rates and as a result improve traffic flow.

Numerous methods have been published for optimized object detection and avoidance for singular – i.e., non-connected – automated cars. Occupancy grid mapping algorithms exist for this domain, wherein the environment of a vehicle is divided into small regions, and the probability of it being taken up by an obstacle is constantly updated [49], [50]. Many articles show solutions for mapping dynamic regions in maps via different methods: using hysteretic validation of dynamic obstacle hypotheses [51], using map-decay [52], and particle filtering based on occupancy grids [53].

However, to the best of our knowledge, no work has been published so far that scrutinizes the online mapping of dynamic obstacles making use of the collaborating nature of connected and cooperating cars.

c: EXTERNAL PERCEPTION

External perception allows vehicles to gain knowledge about the existence and characteristics of objects, which are occluded or outside their sensors range. The foundation of this paradigm is that the knowledge base of individual vehicles can be increased and augmented using the information supplied by sensors of other vehicles or infrastructural elements (e.g., traffic signs, bus stops).

The use of external data can result in progress the smoothness of the navigation of automated vehicles as well as increasing road safety [54]. Müller et al. [55] extended the situational awareness of automated vehicles using cameras mounted on top of lampposts. They used the environmental model created via adopting both internally and externally obtained data to optimize and raise the safety levels of the complex maneuver of automated merging at an intersection.

Currently, to the best of our knowledge, there are no solutions that make use of crowdsourcing-based data acquisition for dynamic obstacle mapping using the external perception-like V2N approach.

II. AUTOMATED DRIVING FRAMEWORK

In this section, the automated driving (AD) framework used to test our approach is described, as well as the trajectory planning method used for the lane change maneuver. The AD framework is based on the six blocks architecture defined previously in [56] and [57], consisting of the following components:

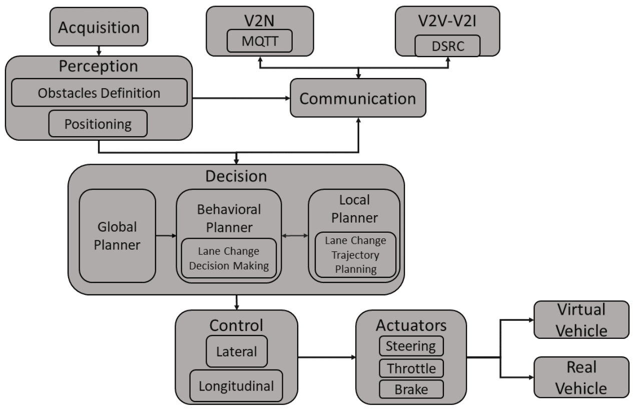


FIGURE 1. Automated Driving six block architecture diagram.

1) Acquisition: This module is in charge of obtaining the information coming through the different sensors (LIDAR, Camera, GNSS, Laser, etc) and sending it to the perception module.
2) Perception: This module interprets these data and generates a representation of the environment and the surrounding obstacles. Furthermore, it provides vehicle positioning.
3) Communication: This module is in charge of obtaining and processing the information coming from other vehicles (V2V), infrastructure (V2I), and/or the cloud (V2N).
4) Decision: Using the information from the perception and communication modules, this module generates a trajectory followed by the vehicle. It is subdivided into a global, behavioral, and local planner.
5) Control: This module is subdivided into a Longitudinal module, which produces throttle and brake signals in order to follow a speed reference and a Lateral module that produces steering signals to follow the trajectory.
6) Actuation: this block interprets the references produced by the control block, so real actuators or simulated vehicle models can read these signals.

The six-block architecture framework is shown in Figure 1. As the decision module is of key importance in relation to the main subject of this study, it is further explained below.

A. DECISION MODULE

As previously mentioned, the decision module is divided into 3 blocks:

- Global Planner: in this block, the first representation of a trajectory is carried out using a predefined OSM map. This definition follows a simplified version of the Lanelet2 standard [58], which is used in an A-star algorithm to find the optimal trajectory from point A to point B. Once this trajectory is found, a drivable space is defined containing the lanes in which the vehicle has to get through to reach its destination [59].

TABLE 1. Methods for automated vehicle navigation-aiding map management.

Classification	Method	Year published	Short description	Disadvantages
Crowdsourced maps	OSM [39]	2008	A crowdsourced platform for VGI data gathering and visualization	Offers no support for low-latency map updates
	Mapillary Vistas [40]	2017	Crowdsourced dataset of human-annotated semantic segmented images collected across 6 continents	Dataset is static that offers no solution for low-latency data updating
	Mapillary Depth [41]	2020	Crowdsourced dataset of street-level images with automatically generated depth information collected across 6 continents	Dataset is static that offers no solution for low-latency data updating
	CrowdMapping [5]	2019	Our previous work describing the basis of a platform for crowdsourcing-based VGI for real-time crowdsourced data gathering and management	Offers no solution for real-time communication of mapped data between connected vehicles
	Google [45] Uber [44] Waymo [42] Lyft [43]	2010 2018 2020 2021	Openly accessible datasets and services released by commercial players	Internally sourced data acquisition & offers no open solution for data updating
Low-latency map updates	Darms et al. [51]	2009	An object mapping and tracking algorithm based on hysteretic validation of dynamic obstacle hypotheses	Requires inputs from several onboard sensors & offers no solution for data sharing between connected vehicles
	Danescu et al. [53]	2011	An object tracking method using a particle filter-based occupancy updating method	Requires birds-eye view images as input & offers no solution for data sharing between connected vehicles
	Veronese et al. [49]	2016	A localization method for autonomous vehicles based on a particle filter-supported 2D occupancy grid	Offers no solution for data sharing between connected vehicles
	Teixeira et al. [52]	2018	A temporally corrected occupancy grid method for obstacle mapping using map-decay for unobservable regions of maps	Offers no solution for data sharing between connected vehicles
	β-SLAM [50]	2019	A simultaneous localization and mapping (SLAM) method for autonomous vehicle positioning and navigation using occupancy grids	Offers no solution for data sharing between connected vehicles
External perception	Müller et al. [55]	2022	A method for extending the situational awareness of connected vehicles using externally sensed obstacles via cameras mounted on top of lampposts	Requires special sensors mounted on infrastructure elements – i.e., lampposts – & takes no data directly from other connected vehicles

- Behavioral Planner: the decision-making is implemented in this block. To do so, a Finite State Machine (FSM) is defined with possible maneuvers that can be executed during the driving process. As has been mentioned, the maneuver analyzed in this work is a lane change, so, two states are possible: keep lane (the base state that is in charge of the basic driving task) and lane change, which will execute depending on environmental variables.
- Local Planner: with the information of the drivable space already supervised by the behavioral planner. This block generates a safe, smooth, and continuous trajectory with a speed profile to be tracked by the vehicle controllers. This trajectory is derived from the Bézier curves approaches presented in previous works [60], [61].

B. LANE CHANGE DEVELOPMENT

The first step in the maneuver is the behavioral planner FSM and the second step is the Bézier local planner. As it can be seen in Figure 2, the FSM states considered for this work, due to the maneuver analyzed, are the Lane Change and the Keep Lane. To go to the Lane change state there are three possible ways:

1) An obstacle is detected in the trajectory of the vehicle.
2) The user decides to change the lane.
3) The lane came to an end or it is closed.

Whereas to go from the lane change state to the keep lane:

1) The lane change is finished.
2) The lane change is aborted.

This work's main case study is when an obstacle is detected in the trajectory.

The second step is the generation of the smooth trajectory used to change lanes. To do so, the work done by Lattarulo et al. [62] was followed, in which different configurations of Bézier parametric curves are defined to design trajectories for different road components and maneuvers such as lane change.

III. REAL-TIME MAP UPDATES

In this section, our novel C/EM scenario is explained. First, we give an overview of the mapping setup, then provide a more in-depth explanation of the mapping algorithm.

A. MAPPING SETUP

Our aim with the mapping framework is to supply the dynamically updated map through the MQTT network: All vehicles

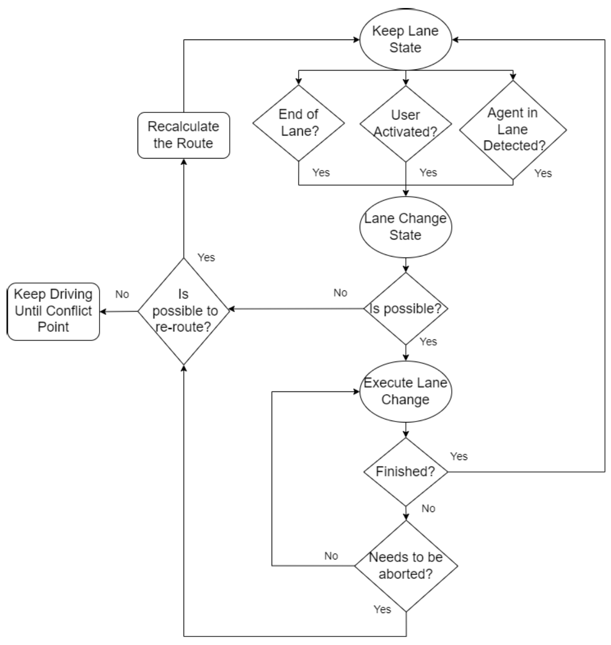


FIGURE 2. State machine diagram for the Keep Lane and the Lane Change states.

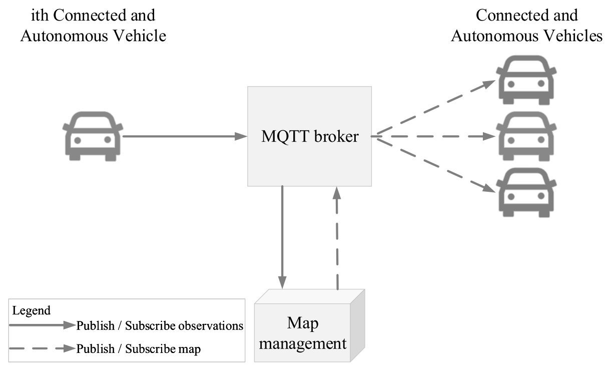

ith Connected and Autonomous Vehicle
Connected and Autonomous Vehicles
MQTT broker
Map management
Legend
Publish / Subscribe observations
Publish / Subscribe map

FIGURE 3. Communication scheme between the individual CAVs, and the map management module. The communication is established via MQTT, therefore all actors shall be connected to the MQTT broker as clients, and publish or subscribe to the same topics.

as well as the map management module are connected to the MQTT broker (Figure 3).

The individual CAVs are publishing pre-processed observation data obtained by their LIDAR sensors over a designated MQTT topic. The map management module is subscribed to this topic and thus receives observations from the vehicles. The information is then processed by the map management module (see Section III-B). If necessary, the updated map is published on another specified topic to the MQTT broker. Since all vehicles are subscribed to this latter topic, they immediately receive map updates when those get published.

B. MAP MANAGEMENT

The map management module's functionality is made possible using our novel C/EM scheme. In this approach, we define two separate maps that are constantly being updated: the **candidate map** ($\mathbf{M}^c$) and the **employed map** ($\mathbf{M}^e$). The candidate map is kept private whereas the employed map is published through the MQTT broker. The ap management module has been implemented using Python as a programming language.

The implemented software uses a state machine architecture. The states are initialization, idle, update map, broadcast map, and exit. During the initialization state, both the employed and the candidate maps are initialized as empty maps.

The functionality of the module following initialization is demonstrated in Figure 4. Once the idle state is reached, the map management module waits for new observations published by the individual CAVs over the "obstacles" topic of MQTT. When a new observation is received, the module decodes it into separate items containing the following parameter fields:

- *ObstacleId*: an ID assigned to the obstacle by the sensor perceiving it,
- *Timestamp*: A timestamp as a string in the format "$YYYY-MM-DDZhh:mm:ss.ms$T".
- *Type*: One of {vehicle, pedestrian, other}.
- *Latitude*: latitude coordinate of the obstacle.
- *Longitude*: longitude coordinate of the obstacle.
- *Speed*: speed of the obstacle.
- *Length*: length of the obstacle.
- *Width*: width of the obstacle.

The payload published over MQTT also contains information on the location of the sensor at the time of the observation – that is, the latitude and longitude coordinates of the LIDAR sensor of the vehicle. This information is necessary since the map management module calculates the observable area prior to the map comparison steps shown in the top row of Figure 4. Two observable maps are calculated as subsets from the employed and candidate map respectively, whose points are closer than a given radius – the default is 50 meters – to the sensor position at time of observation. This results in two sets of items: observable employed items and observable candidate items (denoted by $\mathbf{P}^{e,\,m}$, and $\mathbf{P}^{c,\,m}$, respectively).

Once the observable map subsets are calculated, the newly observed items are matched against the items in the observable subset of the employed map. Then, if any newly observed obstacles remain unmatched, these are matched against the observable subset of the candidate map. For the process steps of matching, see Section III-B1. Finally, if an observed item cannot be assigned to any previously mapped (employed or candidate) item, it is added to the candidate map.

Any obstacle or item listed in one of the maps has the following additional parameters:

- *Obstacle UID*: a globally unique identifier for the item, which is assigned to it as it first appears on either the employed or the candidate map.
- *Number of observations*: a counter showing the amount of times an item has been observed. This number is

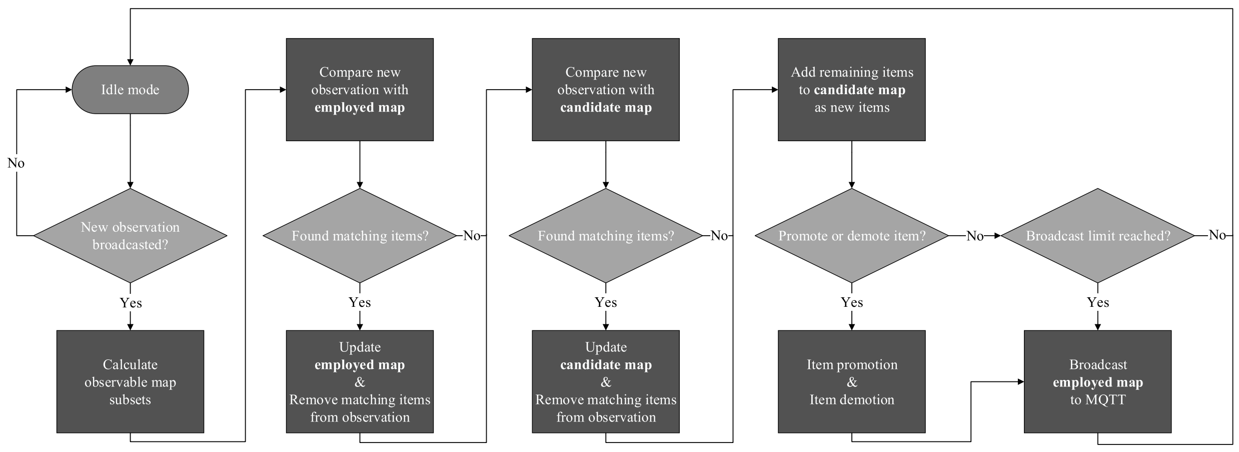

Idle mode
New observation broadcasted?
No
Yes
Calculate observable map subsets
Compare new observation with employed map
Found matching items?
No
Yes
Update employed map & Remove matching items from observation
Compare new observation with candidate map
Found matching items?
No
Yes
Update candidate map & Remove matching items from observation
Add remaining items to candidate map as new items
Promote or demote item?
No
Yes
Item promotion & Item demotion
Broadcast limit reached?
No
Yes
Broadcast employed map to MQTT

FIGURE 4. Process flow diagram of the map management module. The diagram shows the steps the module transitions through after a new observation is received from a CAV over MQTT.

incremented or decremented during the map update steps (see Section III-B1).

- *First timestamp*: timestamp of the first observation of the item.
- *Latest timestamp*: timestamp of the most recent observation of the item.

1) MAP UPDATING

The maps are updated during every iteration when a new observation is received through MQTT (see 2nd and 3rd columns of Figure 4). The updating process finds the potential pairings between the observable subset of a given map and the list of new observations. It does so by calculating the cost matrix ($\mathbb{C}$) using Gaussian Radial Basis Functions (RBF) between the numerical parameters – i.e., latitude, longitude, speed, length, and width – of the mapped and the newly observed items. $\mathbb{C}$ is formulated as

$$\mathbb{C} = \begin{bmatrix} C_{0,0} & C_{0,1} & \dots & C_{0,j} & \dots & C_{0,m} \\ C_{1,0} & C_{1,1} & \dots & C_{1,j} & \dots & C_{1,m} \\ \vdots & \vdots & & & & \\ C_{i,0} & C_{i,1} & & \ddots & & \vdots \\ \vdots & \vdots & & & & \\ C_{n,0} & C_{n,1} & & \dots & & C_{n,m} \end{bmatrix}. \quad (1)$$

It is calculated between mapped items $\mathbf{P}_i^{\mathrm{m}}$, $i = (1, 2, \dots, n)$ and newly observed items $\mathbf{P}_j^{\mathrm{o}}$, $j = (1, 2, \dots, m)$. The elements of $\mathbb{C}$ are calculated as

$$C_{i,j} = \beta \Phi(\mathbf{P}_i^{\mathrm{m}}, \mathbf{P}_j^{\mathrm{o}}, \boldsymbol{\sigma}), \quad (2)$$

where Φ is the Gaussian Radial Basis Function formulated as

$$\Phi(\mathbf{P}_i^{\mathrm{m}}, \mathbf{P}_j^{\mathrm{o}}, \boldsymbol{\sigma}) = \sum_{k=1}^{5} \exp\left(\frac{(P_{i\ k}^{\mathrm{m}} - P_{j\ k}^{\mathrm{o}})^2}{2\sigma_k^2} \right). \quad (3)$$

In the above equation, $k \in$ {latitude, longitude, speed, length, width}, and $\mathbf{P}_i^{\mathrm{m}}$ and $\mathbf{P}_j^{\mathrm{o}}$ are the numerical parameters of the observable obstacles:

$$\mathbf{P}_i^{\mathrm{m}} = \begin{bmatrix} P_{i\ \text{latitude}}^{\mathrm{m}} \\ P_{i\ \text{longitude}}^{\mathrm{m}} \\ P_{i\ \text{speed}}^{\mathrm{m}} \\ P_{i\ \text{length}}^{\mathrm{m}} \\ P_{i\ \text{width}}^{\mathrm{m}} \end{bmatrix}, \ \mathbf{P}_j^{\mathrm{o}} = \begin{bmatrix} P_{j\ \text{latitude}}^{\mathrm{o}} \\ P_{j\ \text{longitude}}^{\mathrm{o}} \\ P_{j\ \text{speed}}^{\mathrm{o}} \\ P_{j\ \text{length}}^{\mathrm{o}} \\ P_{j\ \text{width}}^{\mathrm{o}} \end{bmatrix}. \quad (4)$$

In (3), $\boldsymbol{\sigma}$ is the array of variances corresponding with the numerical parameters describing the individual items. It is formulated as

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{\text{latitude}} \\ \sigma_{\text{longitude}} \\ \sigma_{\text{speed}} \\ \sigma_{\text{length}} \\ \sigma_{\text{width}} \end{bmatrix}. \quad (5)$$

Prior to matching the newly observed items with the mapped ones, a thresholding step is carried out on the cost matrix to forego any weak assignments. In Equation 2, β denotes the masking function:

$$\beta = \begin{cases} 1, & \text{if } \Phi(\mathbf{P}_i^{\mathrm{m}}, \mathbf{P}_j^{\mathrm{o}}, \boldsymbol{\sigma}) \geq T \\ 0, & \text{otherwise} \end{cases}, \quad (6)$$

where T is the preset assignment cost threshold value for the map under inspection – i.e., employed or candidate. Once the cost matrix is finalized, the obstacle matching is calculated using the Hungarian algorithm [63]. This step is denoted by $\mathbf{H}(\mathbb{C})$ in lines 5 and 14 of Algorithm 1.

After the matching is calculated, the numeric parameters of the matched mapped items are updated using an average filter.

$$\mathbf{P}_{\text{matched}}^{\text{m, new}} = \frac{\alpha \mathbf{P}_{\text{matched}}^{\text{m, old}} + \mathbf{P}_{\text{matched}}^{\text{o}}}{\alpha + 1}, \quad (7)$$

where the subscript *matched* denotes that the averaging is performed on the mapped and the newly observed obstacles that have been assigned to each other by the Hungarian algorithm. α is the *number of observations* parameter of the mapped item. If a newly observed item is assigned to a previously mapped item, then this counter is incremented. However, if the item should have been observed – i.e., it is in the observable subset of the map, but it has not been matched with any one of the observed items – then the *number of observations* counter of the item is decremented. For matched items, the *latest timestamp* parameter is also updated to contain the timestamp of this new observation.

As shown in Figure 4, this assignment and updating is first executed on the **employed map**, after which the matched items are removed from the list of new observations. Next, the same matching-updating-removing sequence is performed for the **candidate map**. The remaining items in the new observation are then added to the candidate map with $\alpha = 1$ – meaning that this item has only been observed once.

The final step of the map updating process is item promotion and demotion, which is performed once all the items of the new observation have either been assigned to an item in one of the maps or added as a new item to the candidate map. This is the process of transferring any strong candidate items from the candidate map into the employed map (i.e., promotion) and any weak employed items from the employed map to the candidate map (i.e., demotion). This is performed via hysteretic thresholding to minimize type I. and type II. errors (i.e., falsely demoted employed items and falsely promoted candidate items, respectively) of the transferring process. The threshold values for promotion and demotion therefore must be carefully optimized.

Finally, when the new observation is fully processed, the employed map must be published over MQTT by the map management module, if any of the following conditions are met:

- An item was promoted from the candidate map to the employed map.
- An item was demoted from the employed map to the candidate map.
- The threshold for map re-broadcast has been reached. This threshold parameter contains two separate values – one in the temporal, and one in the spatial dimension. This clause is therefore activated, if the map has not been published for a sufficiently long time period – regardless of whether there have been any new observations published by CAVs – or if the spatial location – i.e., longitude and latitude – of a listed item has changed more then a preset value as a result of the parameter updating shown in Equation 7.

The steps of a map management cycle are shown in Algorithm 1.

Algorithm 1 Map Management Cycle on New Observations

```
Input: $\mathbf{M}^e, \mathbf{M}^c, \boldsymbol{\sigma}$
 1: $\mathbf{P}^{e,m} \leftarrow$ observable_subset($\mathbf{M}^e$)
 2: $\mathbf{P}^{c,m} \leftarrow$ observable_subset($\mathbf{M}^c$)
 3: $\mathbf{P}^o \leftarrow$new_observation[items]
 4: $\mathbb{C}^e \leftarrow \Phi(\mathbf{P}^{e,m}, \mathbf{P}^o, \boldsymbol{\sigma})$
 5: $\mathbf{P}^{e,o}_{\text{matched}} \leftarrow \mathbf{H}(\mathbb{C}^e)$
 6: if $\mathbf{P}^{e,o}_{\text{matched}} \neq \varnothing$ then
 7:     update_map($\mathbf{M}^e, \mathbf{P}^{e,o}_{\text{matched}}$)
 8: end if
 9: for item in $\mathbf{P}^{e,o}_{\text{matched}}$ do
10:     delete item from $\mathbf{P}^o$
11: end for
12: if $\mathbf{P}^o \neq \varnothing$ then
13:     $\mathbb{C}^c \leftarrow \Phi(\mathbf{P}^{c,m}, \mathbf{P}^o, \boldsymbol{\sigma})$
14:     $\mathbf{P}^{c,o}_{\text{matched}} \leftarrow \mathbf{H}(\mathbb{C}^c)$
15:     if $\mathbf{P}^{c,o}_{\text{matched}} \neq \varnothing$ then
16:         update_map($\mathbf{M}^c, \mathbf{P}^{c,o}_{\text{matched}}$)
17:     end if
18:     for item in $\mathbf{P}^{c,o}_{\text{matched}}$ do
19:         delete item from $\mathbf{P}^o$
20:     end for
21:     if $\mathbf{P}^o \neq \varnothing$ then
22:         for item in $\mathbf{P}^o$ do
23:             $\mathbf{M}^c$.add_item(item)
24:         end for
25:     end if
26: end if
27: promote_demote_items($\mathbf{M}^e, \mathbf{M}^c$)
28: if item promoted or item demoted
    or broadcast threshold reached then
29:     $\mathbf{M}^e$.broadcast_map()
30: end if
```

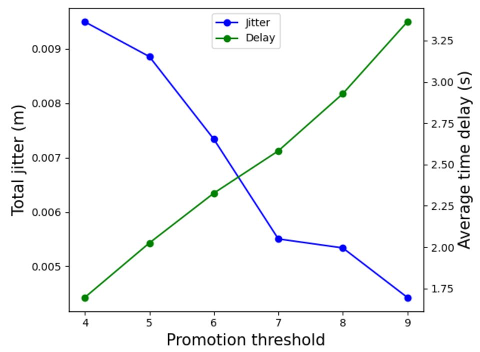


FIGURE 5. Trade-off between time delay of obstacle mapping and location jitter.

IV. RESULTS AND DISCUSSIONS

In this section, the functionality of the map management solution is presented using simulated observation broadcasts. Then, the test scenario is described, taking into account the platforms and procedures used. Furthermore, the results are presented, and so is a discussion of them.

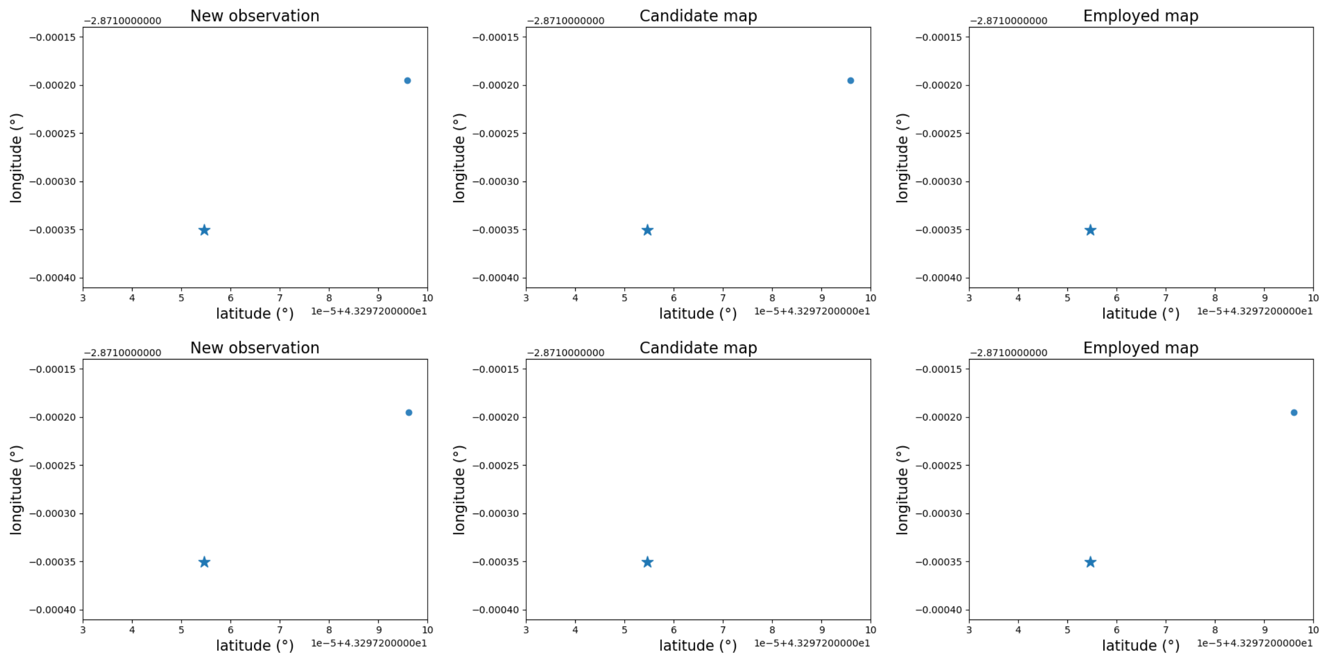


FIGURE 6. Newly observed and mapped obstacles after the 1st (top row) and after the 8th (bottom row) broadcast. Blue star and blue circle denote the ego-vehicle and the obstacle, respectively.

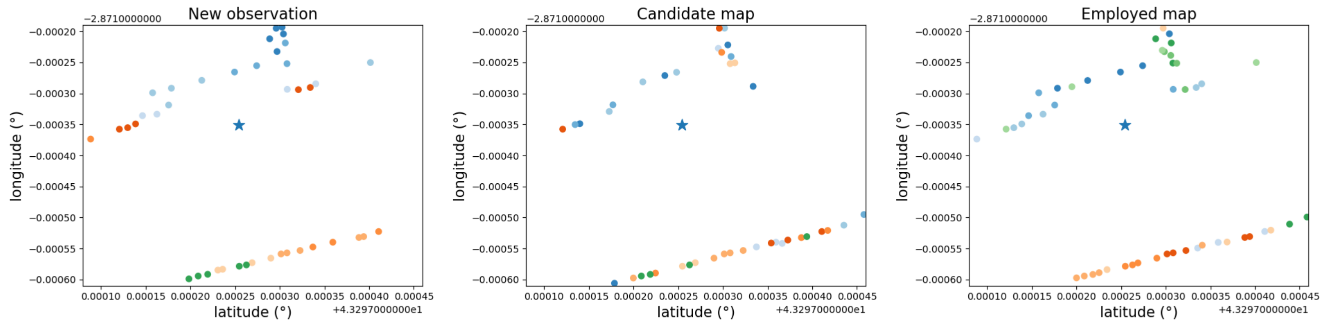


FIGURE 7. Newly observed and mapped obstacles in a simulated environment. Blue stars and circles denote the ego-vehicle and the obstacles, respectively.

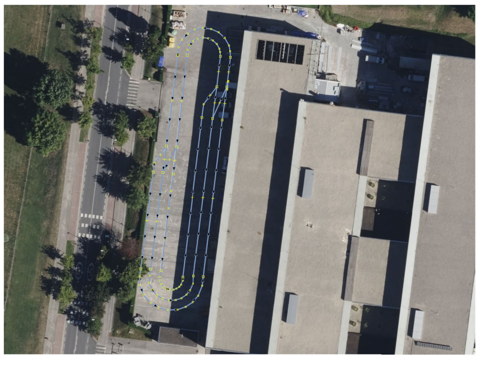


FIGURE 8. Tecnalia Test track.

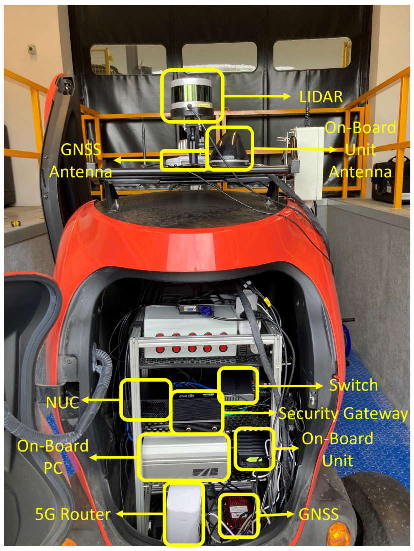
A. MAP MANAGEMENT MODULE

To test the functionality of the C/EM solution, a simulated sequence of observations was presented to the system. The candidate and the employed maps were compared using a series of experiments.

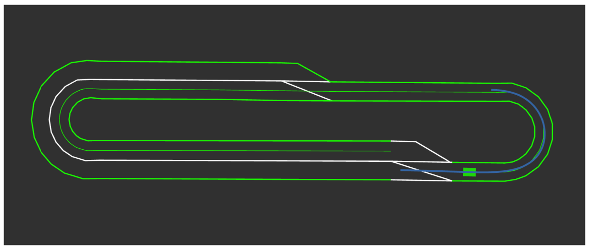
First, we tested a simple simulation where a single obstacle was broadcast by the ego-vehicle over MQTT. The broadcast frequency was 3.33 Hz – i.e., a new broadcast was sent by the simulated ego-vehicle once every 0.3 s. While optimizing the threshold values of the mapping algorithm, we found that there is a trade-off between the long-term precision of the location of an employed obstacle and the time delay it takes the method to promote an obstacle. To understand this phenomenon, we tested the effect of different promotion threshold values on mapping.

We have carried out numerous tests varying the value of the promotion threshold. With the threshold pre-sets, we measured the following values during mapping:

- **Location differences** between the latitude and longitude values of the obstacle in consecutive time frames were

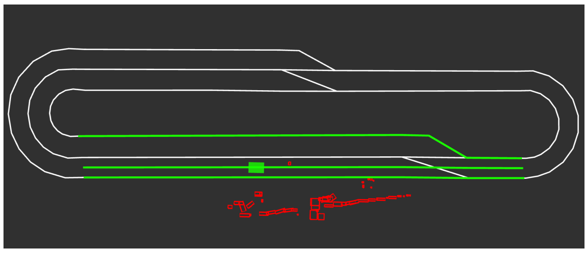


(a) Renault Twizy 80 instrumented

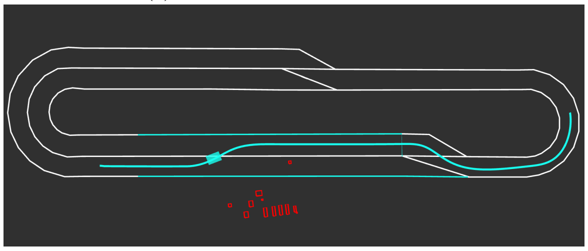


(b) Virtual CAV simulated in ROS

FIGURE 9. **Experimental platforms.**



(a) Real CAV observed in ROS

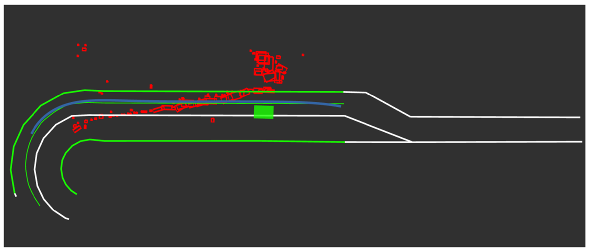


(b) Virtual CAV observed in ROS

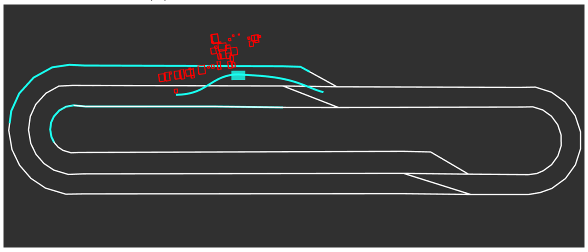
FIGURE 10. **Virtual CAV lane change scenario, from the perspective of both vehicles.**

calculated using the haversine formula. The total jitter was then calculated by summing up the individual location differences through the entire simulation.

- **Time delay** was calculated between the first time frame when the obstacle was broadcast by the ego-vehicle and the timestamp when it was promoted to the employed map.



(a) Real CAV observed in ROS



(b) Virtual CAV observed in ROS

FIGURE 11. **Virtual CAV emergency break scenario, from the perspective of both CAVs.**

The experiments have been carried out 10 times for each promotion threshold value. Time delay values have been averaged over these runs. The results of the experiments are shown in Figure 5.

Based on the results of this test, we opted to use the value 8 for the promotion threshold in our later experiments. We decided to use this value since according to our experimental results, it provided an acceptable time delay while ensuring tolerable noise in the mapped location of the obstacle.

Using the same test simulation as before we tested the functionality of the map management module. We have concluded that the mapping solution successfully included the obstacle to the employed map and published them again over a different MQTT topic. During the test shown in Figure 6, the obstacle was promoted to the employed map once the preset promotion threshold was reached – this value was set to 8, hence it took the mapping system 8 subsequent observations to promote the obstacle.

Moreover, we have tested the capabilities of our newly introduced C/EM solution using a simulation with numerous observed obstacles. Figure 7 shows the functionality of our implementation at a randomly selected time-frame of a series of observations acquired by an ego-vehicle in a simulated environment.

B. TEST SCENARIO

In order to test our solution, the Tecnalia Test Track (Figure 8) was used, where the real CAV detects static obstacles and sends them through MQTT to the map management module. Additionally, a virtual CAV is implemented, which receives information about the obstacles and executes the corresponding maneuver. The goal of the scenario is to show the capabilities of the map and the reaction of vehicles that can neither rely on internal perception nor V2V communication.

The real platform is a Renault Twizy 80 (Figure 9a), instrumented with different sensors such as LIDAR, GNSS, throttle, brake, and steering actuators. Furthermore, it is connecting to the internet safely, thanks to the protection of the IoTAC platform [64] and also enables V2X connectivity through a Commsignia OBU. The virtual CAV (Figure 9b) is simulated with a desktop computer with internet connectivity. Both platforms run the AUDRIC2 architecture described in Section II, with the difference in the model of the actuator. In the Twizy, the control outputs are sent to the real actuators, whereas the virtual CAV runs the architecture using a kinematic bicycle model as an actuator model in Robot Operating System (ROS). By implementing this testing method, situations where one vehicle can not count on its perception system or has one that is compromised can be mitigated in a safe manner without removing the conditions provided by real environments.

C. RESULTS

Mainly, two cases were studied. First, the vehicle's ability to change the lane and avoid obstacles, and second, if the lane change is not possible, execute other maneuvers (emergency braking, re-route, etc.). The first case can be observed in Figure 10 from the perspective of both vehicles. Figure 10a shows the Twizy (green box) and the obstacles detected (red boxes) on the real test track in ROS. These obstacles, due to the clustering algorithm used, were separated into smaller rectangles that were sent to the map management module. Figure 10b shows the virtual CAV (blue box) in ROS with a filtrated amount of obstacles, received by the map management module (red boxes). Furthermore, it can be seen that if one obstacle is in the same lane as the virtual CAV, the lane change trajectory is planned. Thus, the vehicle is able to avoid the obstacle and accomplish the maneuver.

The second case is shown in Figure 11, where obstacles block both lanes as can be seen in Figure 11a. The virtual CAV re-planning the trajectory can be observed in Figure 11b, however, as it detects the obstacle in the other lane, it aborts the maneuver and proceeds with braking in front of the obstacles.

D. DISCUSSIONS

The map management module is capable of introducing obstacles to the candidate and then, to the employed map and broadcasting it over MQTT (Figures 6 and 7). The time delay of this process is characterized by the promotion threshold. We have identified a trade-off between this time-delay and the location jitter of the individual employed obstacles - i.e., how much their location inappropriately changes over time. This trade-off is shown in Figure 5.

As shown in Figure 10 and Figure 11, the proposed approach combining the map management solution with the AUDRIC2 framework through MQTT connection presents promising results as in both cases the maneuvers were completed successfully. In the case of the lane change, the vehicle could react with enough distance (16 m) to perform a safe lane change in 3 s. That is within the time limits of an average lane change [65], [66]. This is achieved without compromising the driving performance since a smooth trajectory is generated without disturbing the lateral behavior of the vehicle. In the case of the braking maneuver, it can be observed that the lane change is activated, however, since it still detects an obstacle blocking the other lane, the virtual CAV proceeds with the braking maneuver, stopping at 6.3 m from the obstacle.

V. CONCLUSION

In this paper, we have introduced a map management module that communicates with CAVs using MQTT. We have shown that this novel solution is capable of introducing low-latency externally perceived obstacles to CAVs connected to it. Using a novel candidate/employed mapping solution, we have proven that objects observed on road segments by a physical CAV can be registered to a map with low-latency and thus known to other CAVs. We have demonstrated the abilities of the novel candidate/employed mapping in two maneuvers; first, a successful and safe lane change maneuver was carried out with sufficient clearance – that is not coming closer than 16 m to the obstacles and performing the lane change in 3 s. Secondly, if the lane change is unavailable (i.e., the traffic lane is blocked), the maneuver is aborted successfully and the vehicle stops at a safe distance (6.3 m) from all obstacles. Both maneuvers were successful using our solution based on information from external sources (i.e., sensors of other CAVs) gathered in the employed map and distributed over MQTT.

A. FUTURE WORK

Seeing the results of our novel proposed solution, we plan on further investigating its usage in more complex and realistic scenarios. One way to scale up the complexity presented in this study, is to increase the number of agents participating in the sharing of data; physical and virtual CAVs as well as independent road users. Another generalization that could be further researched, is to extend the study into more complex and general maneuvers, which would be representative of CAVs continuous operation, and its effects in and its effects with regard to crowdsourced mapping solutions.

The robust functionality of our proposed solution – i.e., its resilience to faulty data points and outliers – can be substantially increased by involving more data acquisition vehicles. Information resulting in faulty data points can be caused by malfunctions in the acquisition, perception, or even communication systems in the vehicle, including possible attacks in a vehicle asset. It is important to correctly model the effects in the mapping system, to have better understanding on the requirements needed for a high level of availability and correctness in the mapping solution proposed.

A way of testing this, would be to increase the number of physical as well as virtual vehicles in complex scenarios.

In the case of larger scenarios, a grid-based version of our solution could be implemented, that would enable the use of locally crowdsourced maps. In this setup, the map could be divided into smaller regions, over which different map management modules would be run simultaneously. This development however raises questions regarding the different modalities involved in our solution, which can be researched in a future study.

A key component for crowdsourcing is to define the task in a way that makes it as inclusive as possible. In the case of our proposed solution, this was achieved through the use of MQTT. Our MQTT solution used the currently supplied mechanisms of cybersecurity present in the protocol; each vehicle was authorized and whitelisted for the topics, and certificates were necessary to establish connectivity to the broker. Nonetheless, other channels of communication can be evaluated for the functionality of the map management solution in scenarios. An in depth analysis of risks and threats for this communication channels need to be performed before a full deployment. An alternative is to make use of V2X communication protocols – e.g., Dedicated Short-Range Communication (DSRC), cellular V2X – which are being studied and employed in the CAV sector, and extend their cooperative services by using our proposed solution.

REFERENCES

[1] M. Schranz, G. A. D. Caro, T. Schmickl, W. Elmenreich, F. Arvin, A. Şekercioğlu, and M. Sende, "Swarm intelligence and cyber-physical systems: Concepts, challenges and future trends," *Swarm Evol. Comput.*, vol. 60, Feb. 2021, Art. no. 100762. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210650220304156

[2] K. Kiela, V. Barzdenas, M. Jurgo, V. Macaitis, J. Rafanavicius, A. Vasjanov, L. Kladovscikov, and R. Navickas, "Review of V2X–IoT standards and frameworks for ITS applications," *Appl. Sci.*, vol. 10, no. 12, p. 4314, Jun. 2020.

[3] B. Mishra and A. Kertesz, "The use of MQTT in M2M and IoT systems: A survey," *IEEE Access*, vol. 8, pp. 201071–201086, 2020.

[4] J. Van Brummelen, M. O'Brien, D. Gruyer, and H. Najjaran, "Autonomous vehicle perception: The technology of today and tomorrow," *Transp. Res. C, Emerg. Technol.*, vol. 89, pp. 384–406, Apr. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X18302134

[5] M. Szántó and L. Vajta, "Introducing CrowdMapping: A novel system for generating autonomous driving aiding traffic network databases," in *Proc. Int. Conf. Control, Artif. Intell., Robot. Optim. (ICCAIRO)*, Dec. 2019, pp. 7–12.

[6] Y. Liu, X. Wang, L. Li, S. Cheng, and Z. Chen, "A novel lane change decision-making model of autonomous vehicle based on support vector machine," *IEEE Access*, vol. 7, pp. 26543–26550, 2019.

[7] S. Liu, H. Su, Y. Zhao, K. Zeng, and K. Zheng, "Lane change scheduling for autonomous vehicle: A prediction-and-search framework," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 3343–3353.

[8] S. Hwang, K. Lee, H. Jeon, and D. Kum, "Autonomous vehicle cut-in algorithm for lane-merging scenarios via policy-based reinforcement learning nested within finite-state machine," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17594–17606, Oct. 2022.

[9] J. Wang, Q. Zhang, D. Zhao, and Y. Chen, "Lane change decision-making through deep reinforcement learning with rule-based constraints," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–6.

[10] A. Ji and D. Levinson, "A review of game theory models of lane changing," *Transportmetrica A, Transp. Sci.*, vol. 16, no. 3, pp. 1628–1647, Jan. 2020.

[11] L. Peng, Y. Yan, J. Wang, D. Han, Y. Yao, and G. Yin, "Hierarchical motion planning system with consideration of the dynamic lane-changing behaviour," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2022, pp. 3455–3460.

[12] S. Liu, J. Xie, and J. Y. Wang, "Design of collision avoidance method for vehicle emergency lane change based on Bezier curve," *Adv. Transp. Stud.*, vol. 1, pp. 35–46, Mar. 2022.

[13] T. Li, J. Wu, C.-Y. Chan, M. Liu, C. Zhu, W. Lu, and K. Hu, "A cooperative lane change model for connected and automated vehicles," *IEEE Access*, vol. 8, pp. 54940–54951, 2020.

[14] Y. Zheng, W. Ding, B. Ran, X. Qu, and Y. Zhang, "Coordinated decisions of discretionary lane change between connected and automated vehicles on freeways: A game theory-based lane change strategy," *IET Intell. Transp. Syst.*, vol. 14, no. 13, pp. 1864–1870, Dec. 2020.

[15] M. Atagoziev, E. G. Schmidt, and K. W. Schmidt, "Lane change scheduling for connected and autonomous vehicles," *Transp. Res. C, Emerg. Technol.*, vol. 147, Feb. 2023, Art. no. 103985.

[16] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.

[17] W. Lim, S. Lee, M. Sunwoo, and K. Jo, "Hierarchical trajectory planning of an autonomous car based on the integration of a sampling and an optimization method," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 613–626, Feb. 2018.

[18] G. Che, L. Liu, and Z. Yu, "An improved ant colony optimization algorithm based on particle swarm optimization algorithm for path planning of autonomous underwater vehicle," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 8, pp. 3349–3354, Aug. 2020.

[19] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, "Unified multiobjective optimization scheme for aeroassisted vehicle trajectory planning," *J. Guid., Control, Dyn.*, vol. 41, no. 7, pp. 1521–1530, Jul. 2018.

[20] D. J. Kim, Y. W. Jeong, and C. C. Chung, "Lateral vehicle trajectory planning using a model predictive control scheme for an automated perpendicular parking system," *IEEE Trans. Ind. Electron.*, vol. 70, no. 2, pp. 1820–1829, Feb. 2023.

[21] H. Guo, C. Shen, H. Zhang, H. Chen, and R. Jia, "Simultaneous trajectory planning and tracking using an MPC method for cyber-physical systems: A case study of obstacle avoidance for an intelligent vehicle," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4273–4283, Sep. 2018.

[22] L. Li, Y. Miao, A. H. Qureshi, and M. C. Yip, "MPC-MPNet: Model-predictive motion planning networks for fast, near-optimal planning under kinodynamic constraints," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4496–4503, Jul. 2021.

[23] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixão, F. Mutz, L. de Paula Veronese, T. Oliveira-Santos, and A. F. D. Souza, "Self-driving cars: A survey," *Expert Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113816.

[24] C. Zhang, D. Chu, S. Liu, Z. Deng, C. Wu, and X. Su, "Trajectory planning and tracking for autonomous vehicle based on state lattice and model predictive control," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 2, pp. 29–40, Summer 2019.

[25] T. Stahl, A. Wischnewski, J. Betz, and M. Lienkamp, "Multilayer graph-based trajectory planning for race vehicles in dynamic scenarios," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3149–3154.

[26] K. Bergman, O. Ljungqvist, T. Glad, and D. Axehill, "An optimization-based receding horizon trajectory planning algorithm," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 15550–15557, 2020.

[27] Z. Yongzhe, B. Ma, and C. K. Wai, "A practical study of time-elastic-band planning method for driverless vehicle for auto-parking," in *Proc. Int. Conf. Intell. Auto. Syst. (ICoIAS)*, Mar. 2018, pp. 196–200.

[28] S. Y. Gelbal, B. Aksun-Guvenc, and L. Guvenc, "Collision avoidance of low speed autonomous shuttles with pedestrians," *Int. J. Automot. Technol.*, vol. 21, no. 4, pp. 903–917, Aug. 2020.

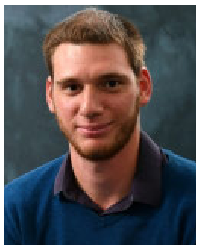
[29] W. Xu, R. Sainct, D. Gruyer, and O. Orfila, "Safe vehicle trajectory planning in an autonomous decision support framework for emergency situations," *Appl. Sci.*, vol. 11, no. 14, p. 6373, Jul. 2021.

[30] J. Ziegler and C. Stiller, "Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 1879–1884.

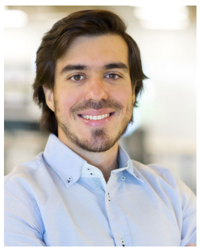
[31] G. Ignéczi and E. Horváth, "A clothoid-based local trajectory planner with extended Kalman filter," in *Proc. IEEE 20th Jubilee World Symp. Appl. Mach. Intell. Informat. (SAMI)*, Mar. 2022, pp. 467–472.

[32] E. Lambert, R. Romano, and D. Watling, "Optimal path planning with clothoid curves for passenger comfort," in *Proc. 5th Int. Conf. Vehicle Technol. Intell. Transp. Syst.*, 2019, pp. 609–615.

[33] C. Hidalgo, R. Lattarulo, J. Pérez, and E. Asua, "Hybrid trajectory planning approach for roundabout merging scenarios," in *Proc. IEEE Int. Conf. Connected Vehicles Expo. (ICCVE)*, Nov. 2019, pp. 1–6.
[34] R. Lattarulo and J. P. Rastelli, "A hybrid planning approach based on MPC and parametric curves for overtaking maneuvers," *Sensors*, vol. 21, no. 2, p. 595, Jan. 2021.
[35] R. Liu, J. Wang, and B. Zhang, "High definition map for automated driving: Overview and analysis," *J. Navigat.*, vol. 73, no. 2, pp. 324–341, Mar. 2020.
[36] J. Howe, "The rise of crowdsourcing," *Wired Mag.*, vol. 14, no. 6, pp. 1–4, Jun. 2006.
[37] M. F. Goodchild, "Citizens as sensors: The world of volunteered geography," *GeoJournal*, vol. 69, no. 4, pp. 211–221, Nov. 2007.
[38] D. Sui, S. Elwood, and M. Goodchild, *Crowdsourcing Geographic Knowledge: Volunteered Geographic Information (VGI) in Theory and Practice*. Cham, Switzerland: Springer, 2012.
[39] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," *IEEE Pervasive Comput.*, vol. 7, no. 4, pp. 12–18, Oct. 2008.
[40] G. Neuhold, T. Ollmann, S. R. Bulò, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5000–5009.
[41] M. L. Antequera, P. Gargallo, M. Hofinger, S. R. Bulò, Y. Kuang, and P. Kontschieder, "Mapillary planet-scale depth dataset," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 589–604.
[42] P. Sun, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2443–2451.
[43] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, L. Chen, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska, "One thousand and one hours: Self-driving motion prediction dataset," in *Proc. Conf. Robot Learn.*, 2021, pp. 409–418.
[44] I. A. Barsan, S. Wang, A. Pokrovsky, and R. Urtasun, "Learning to localize using a LiDAR intensity map," in *Proc. Conf. Robot Learn.*, 2018, pp. 605–616.
[45] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver, "Google street view: Capturing the world at street level," *Computer*, vol. 43, no. 6, pp. 32–38, Jun. 2010.
[46] H. G. Seif and X. Hu, "Autonomous driving in the iCity—HD maps as a key challenge of the automotive industry," *Engineering*, vol. 2, no. 2, pp. 159–162, Jun. 2016.
[47] Md. M. K. Tareq, O. Semiari, M. A. Salehi, and W. Saad, "Ultra reliable, low latency vehicle-to-infrastructure wireless communications with edge computing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.
[48] Y. Gu, J. Liu, and L. Zhao, "Low-latency transmission and caching of high definition map at a crossroad," in *Proc. Int. Conf. Commun. Netw. China*. Cham, Switzerland: Springer, 2019, pp. 264–277.
[49] L. D. P. Veronese, J. Guivant, F. A. A. Cheein, T. Oliveira-Santos, F. Mutz, E. de Aguiar, C. Badue, and A. F. D. Souza, "A light-weight yet accurate localization system for autonomous cars in large-scale and complex environments," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 520–525.
[50] J. Clemens, T. Kluth, and T. Reineking, "β-SLAM: Simultaneous localization and grid mapping with beta distributions," *Inf. Fusion*, vol. 52, pp. 62–75, Dec. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253516301579
[51] M. S. Darms, P. E. Rybski, C. Baker, and C. Urmson, "Obstacle detection and tracking for the urban challenge," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 475–485, Sep. 2009.
[52] T. Teixeira, F. Mutz, K. S. Komati, L. Veronese, V. B. Cardoso, C. Badue, T. Oliveira-Santos, and A. F. D. Souza, "Memory-like map decay for autonomous vehicles based on grid maps," 2018, *arXiv:1810.02355*.
[53] R. Danescu, F. Oniga, and S. Nedevschi, "Modeling and tracking the driving environment with a particle-based occupancy grid," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1331–1342, Dec. 2011.
[54] P. Mallozzi, P. Pelliccione, A. Knauss, C. Berger, and N. Mohammadiha, "Autonomous vehicles: state of the art, future trends, and challenges," *Automot. Syst. Softw. Eng.*, pp. 347–367, 2019.
[55] J. Müller, J. Strohbeck, M. Herrmann, and M. Buchholz, "Motion planning for connected automated vehicles at occluded intersections with infrastructure sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17479–17490, Oct. 2022.
[56] R. Lattarulo, J. Pérez, and M. Dendaluce, "A complete framework for developing and testing automated driving controllers," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 258–263, Jul. 2017.
[57] C. Hidalgo, R. Lattarulo, C. Flores, and J. P. Rastelli, "Platoon merging approach based on hybrid trajectory planning and CACC strategies," *Sensors*, vol. 21, no. 8, p. 2626, Apr. 2021.
[58] F. Poggenhans, J. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr, "Lanelet2: A high-definition map framework for the future of automated driving," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 1672–1679.
[59] R. Lattarulo, C. Hidalgo, A. Arizala, and J. Perez, "AUDRIC2: A modular and highly interconnected automated driving framework focus on decision making and vehicle control," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 763–769.
[60] R. Lattarulo, L. González, and J. Perez, "Real-time trajectory planning method based on N-order curve optimization," in *Proc. 24th Int. Conf. Syst. Theory, Control Comput. (ICSTCC)*, Oct. 2020, pp. 751–756.
[61] R. Lattarulo and J. Perez, "Fast real-time trajectory planning method with 3rd-order curve optimization for automated vehicles," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6.
[62] R. Lattarulo, L. González, E. Martí, J. Matute, M. Marcano, and J. Pérez, "Urban motion planning framework based on N-Bézier curves considering comfort and safety," *J. Adv. Transp.*, vol. 2018, pp. 1–13, Jul. 2018.
[63] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1955.
[64] M. Siavvas, E. Gelenbe, D. Tsoukalas, I. Kalouptsoglou, M. Mathioudaki, M. Nakip, D. Kehagias, and D. Tzovaras, "The IoTAC software security-by-design platform: Concept, challenges, and preliminary overview," in *Proc. 18th Int. Conf. Design Reliable Commun. Netw. (DRCN)*, Mar. 2022, pp. 1–6.
[65] H. Ataelmanan, O. C. Puan, and S. A. Hassan, "Examination of lane changing duration time on expressway," *IOP Conf. Series, Mater. Sci. Eng.*, vol. 1144, no. 1, May 2021, Art. no. 012078.
[66] T. Toledo and D. Zohar, "Modeling duration of lane changes," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1999, no. 1, pp. 71–78, Jan. 2007.



MÁTYÁS SZÁNTÓ received the B.Sc. and M.Sc. degrees in mechatronics engineering from the Budapest University of Technology and Economics and the Ph.D. degree from the Department of Control Engineering and Information Technology, Budapest University of Technology and Economics, while being involved in the mentoring of students and the teaching of image processing subjects. His research interests include crowdsourcing for mapping purposes, 3D vision, and sim-to-real technologies.



CARLOS HIDALGO received the B.Sc. degree in electronic engineering from Simón Bolívar University, Venezuela, in 2018. He is currently pursuing the Ph.D. degree in physical engineering with the University of the Basque Country, Spain. He is a Research Engineer with the Cooperative, Connected and Automated Mobility Team, Tecnalia, a member of the Basque Research and Technology Alliance, Spain. His research interests include decision-making and control strategies for connected and automated vehicles.

LEONARDO GONZÁLEZ received the B.E. degree in electronic engineering from Simón Bolívar University, Venezuela, in 2015. He joined Tecnalia, a member of the Basque Research and Technology Alliance, in 2017, where he is currently a Researcher in the field of automated vehicles, with the Cooperative, Connected and Automated Mobility Team. His research interests include decision systems, risk estimation approaches, and remote monitoring for automated vehicles.

JOSHUÉ PÉREZ RASTELLI (Member, IEEE) received the B.E. degree in electronic engineering from Simon Bolívar University, Venezuela, in 2007, and the M.E. and Ph.D. degrees from the University Complutense of Madrid, in 2009 and 2012, respectively. He has been a Research Leader of the Connected and Automated Driving Team, Tecnalia, a member of the Basque Research and Technology Alliance, since 2015. He has more than 15 years of experience in the intelligent transportation system field and more than 180 publications related to automated driving and ADAS.

ESTIBALIZ ASUA was born in Bilbao, Spain, in 1982. She received the electronic engineering and Ph.D. degrees from the University of the Basque Country (UPV/EHU), Spain, in 2005 and 2009, respectively. The thesis topic was micro- and nano-positioning using smart materials. After several teaching positions with UPV/EHU and the Public University of Navarre, Spain, she is currently an Associate Professor with UPV/EHU. Her current research interests include the development and application of smart sensors and actuators and scientific instrumentation, and the development of intelligent systems to improve the advanced comfort and driver assistant systems in vehicles.

LÁSZLÓ VAJTA received the Master of Electrical Engineering degree in BME, in 1975, the University Ph.D. degree, in 1981, the M.B.A. degree, in 2000, and the Ph.D. degree in picture processing, in 2005. He was a Research and Development Expert Engineer, in 1980. He was the Deputy Dean for Economic Affairs (2005–2008) and the Dean of the Faculty of Electrical Engineering and Informatics (2008–2016). He was the Vice-Rector for innovation and company relations with the Budapest University of Technology and Economics (2008–2012). Since 1989, he has been the CEO of Telekodex Ltd. He is currently an Associate Professor with the Department of Control Engineering and Information Technology. His special fields are control engineering, computerized image processing, robotics, and industrial and energy informatics. He worked at many companies in the frame of co-operation participating in numerous industrial projects. He lead numerous research projects financed by national and EU grants. He has over 200 publications mostly in international journals and conference proceedings. He owns eight patents.