Journal of Advanced Transportation

Cooperative Systems for Autonomous Vehicles

Lead Guest Editor: José M. Armingol
Guest Editors: Cristina Olaverri-Monreal, Fernando García, Vicente Milanés, and David Martín

Cooperative Systems for Autonomous Vehicles

Journal of Advanced Transportation

Cooperative Systems for Autonomous Vehicles

Lead Guest Editor: José M. Armingol
Guest Editors: Cristina Olaverri-Monreal, Fernando García, Vicente Milanés, and David Martín

Editorial Board

Contents

Hindawi
Journal of Advanced Transportation
Volume 2018, Article ID 8714549, 1 page
https://doi.org/10.1155/2018/8714549

Editorial

Cooperative Systems for Autonomous Vehicles

José M. Armingol[1], **Cristina Olaverri-Monreal**[2], **Fernando García**[1], **Vicente Milanés**[3] **and David Martín**[1]

[1] *Universidad Carlos III de Madrid (Intelligent Systems Lab), Madrid, Spain*
[2] *Johannes Kepler University, Linz, Austria*
[3] *Renault SAS, Guyancourt, France*

Correspondence should be addressed to José M. Armingol; armingol@ing.uc3m.es

Received 28 August 2018; Accepted 28 August 2018; Published 18 September 2018

Intelligent Transport Systems are responsible for providing technology to improve the operation, management, and safety of land transport in urban, interurban, and rural roads. With the current interest in the development of autonomous vehicles, cooperative systems C-ITS can be considered as catalysts of these, and their implementation, not limited to independent actions, but adding perception, decision, and additional actions supported by communications, aiming to the cooperative autonomous driving.

In this scenario, the European Platform C-ITS approved the deployment of the first set of cooperative systems, called Day 1, which will use V2V and V2I communications. At this time, it is expected that this initial set will be expanded to the C-ITS Day 1.5, with the intention that, in 2020, 20% of vehicles have all these capabilities. Therefore, a shared traffic situation arises among vehicles with cooperative capacities with autonomous or semiautonomous functions, with others with purely manual driving. This situation will be the most common for a long period of time.

For all these reasons, the objective of this special issue focuses on research and development of technologies of C-ITS systems, focused on the cooperative autonomous driving. It is in complex scenarios where C-ITS systems can provide further solutions to the problems identified. In addition, studies of the impact of these technologies in drivers are another important aspect. These studies will focus on both drivers of vehicles equipped with C-ITS solutions and those not equipped with them, with the purpose of establishing the basis to ensure coexistence in shared traffic between the two types of vehicles, during the transition to the fully autonomous driving.

C-ITS increase the quality and reliability of information available to drivers about their immediate environment, other vehicles, and road users by sharing information such as vehicle position, direction, and speed, with other connected vehicles in real-time. This information helps us observe and assess what is happening on our roads, building a picture of real-time traffic situations.

This technology can respond to the traffic situation in a number of ways such as providing warning messages to driver via variable message signs, reducing variable speed limits, and updating available traveler information.

Cooperative Intelligent Transport Systems enable vehicles, infrastructure, personal mobile devices, and transport management systems to share information about the road safely through a wireless network.

Conflicts of Interest

The authors declare no conflicts of interest.

José M. Armingol
Cristina Olaverri-Monreal
Fernando García
Vicente Milanés
David Martín

Hindawi
Journal of Advanced Transportation
Volume 2018, Article ID 3785957, 11 pages
https://doi.org/10.1155/2018/3785957

Research Article

Implementation and Evaluation of a Traffic Light Assistance System Based on V2I Communication in a Simulation Framework

Cristina Olaverri-Monreal,[1] **Javier Errea-Moreno**,[2] **and Alberto Díaz-Álvarez**[3]

[1]*Department of Information Engineering and Security, University of Applied Sciences Technikum Wien, Vienna, Austria*
[2]*Department of Electrical and Electronic Engineering, Universidad Pública de Navarra, Pamplona, Navarra, Spain*
[3]*Universidad Politécnica de Madrid, Instituto Universitario de Investigación del Automóvil (INSIA), Madrid, Spain*

Correspondence should be addressed to Cristina Olaverri-Monreal; olaverri@technikum-wien.at

Received 30 October 2017; Accepted 8 February 2018; Published 18 March 2018

Academic Editor: Yuchuan Du

Cooperative Intelligent Transportation Systems (C-ITS) make the exchange of information possible through cooperative systems that broadcast traffic data to enhance road safety. Traffic light assistance (TLA) systems in particular utilize real-time traffic light timing data by accessing the information directly from the traffic management center. To test the reliability of a TLA system based on networked intervehicular interaction with infrastructure, we present in this paper an approach to perform theoretical studies in a lab-controlled scenario. The proposed system retrieves the traffic light timing program within a range in order to calculate the optimal speed while approaching an intersection and shows a recommended velocity based on the vehicle's current acceleration and speed, phase state of the traffic light, and remaining phase duration. Results show an increase in driving efficiency in the form of improvement of traffic flow, reduced gas emissions, and waiting time at traffic lights after the drivers adjusted their velocity to the speed calculated by the system.

1. Introduction

Cooperative Intelligent Transportation Systems (C-ITS) make the exchange of information possible through cooperative systems that broadcast traffic data to enhance road safety. Lab-controlled platforms provide the test bed conditions required to perform realistic experiments with massive amounts of valuable data. This allows for the evaluation of a variety of protocols, as well as interaction with in-vehicle systems and services [1].

According to research developed in the United States by the National Highway Traffic System Administration (NHTSA), drivers who did not respect red lights in an intersection controlled by a traffic light caused from 1997 until 2004 an average of 51% of the fatal traffic crashes [2].

The technology behind traffic light timing programs is rapidly evolving. The aim is to further develop adaptive signal control technology (ASCT) in an effort to adapt traffic light timing programs to the demands of real-time traffic and thus reduce traffic congestion in urban areas. However, these technologies have only been implemented in a small number of road networks, and the majority of urban areas still have pretimed control systems installed in their traffic systems. Their preset time intervals are the same every time the signal cycles, regardless of changes in traffic volume [3]. This pretimed mode is ideally suited to closely spaced intersections where traffic volumes and patterns are consistent. It can be used to provide efficient coordination with other traffic signals, since both the start and end of green light phases are predefined. On the other hand, pretimed control tends to be inefficient at intersections where traffic arrivals are random. This causes redundant and unnecessary waiting times, significant traffic jams, and excessive delays [4].

Some systems use different preset time intervals for morning rush hour, evening rush hour, and other busy times based on historical data. Even if the information of the traffic lights is stored at each traffic management center, it is not yet

used to help drivers to improve their driving behavior while approaching an intersection controlled by a traffic light [6]. Some works have demonstrated that visual warning signals can improve driving behavior [7], and due to the complex traffic situations at urban intersections there is a need for more intelligent control solutions.

Along this line of intelligent solutions, vehicle-to-everything communication (V2X) is applied to Intelligent Transportation Systems (ITS) to establish not only vehicle-to-vehicle (V2V) connection, but also to the surrounding infrastructure. This connectivity enhances traffic situational awareness across the entire road network and aims at optimizing traffic flow, reducing the number of road accidents, lowering traffic congestion, and minimizing carbon emissions. V2X technologies require in-vehicle technology (i.e., on-board units) and roadside units (RSU) attached to the infrastructures of the road. The world's first digital highway was established in 2015 on the A9 highway in Bavaria, Germany [8]. Digital highways are stretches of road for introducing and testing applications based on V2X. More than 50 car companies, such as Audi, Mercedes, and BMW, have expressed their interest in testing new technologies on this digital highway.

ITS provide real-time information for decision-making algorithms that can be then used by travelers and traffic management systems. Different advanced driver assistance systems (ADAS) have been developed to make drivers aware of their surroundings while driving. Some of the most common include forward collision warning (FCW), which detects a leading vehicle and calculates relative speed and distance in order to generate warning signals based on time to collision (TTC); traffic sign recognition (TSR) or systems that recognize different traffic signs to augment the field of view of the driver; traffic light recognition (TLR), to detect and recognize traffic lights; and traffic light assistance (TLA) systems that acquire data related to the signal timing, tailback, and geometry of intersections and combine it with in-vehicle data [9]. For each of them, collection of different data is required, such as speed or distance to the following car or next traffic light.

As a natural advance in the research in cooperative systems and automated vehicle development, autonomous driving technology includes V2X capabilities. We implemented a simulator platform to test those ADAS that rely on vehicle-to-infrastructure- (V2I-) based communication and present in this paper a TLA system in a simulated context. The proposed system retrieves the traffic light timing program within a range, in order to calculate the optimal speed while approaching an intersection. The system warns the driver by displaying a message in the dashboard, showing a recommended velocity based on the vehicle's current acceleration and speed, phase state of the traffic light, and remaining phase duration. Driving efficiency is increased once the driver adjusts their velocity to the speed calculated by the system. By retrieving the information from the traffic light predefined program, it is possible to improve traffic flow and reduce traffic delays.

In order to develop the approach, Simulation of Urban Mobility (SUMO) [10], Unity 3D [11], and City Engine [12] were used for the development of the two different scenarios for the experiments. Routing directions were displayed in the user-controlled vehicle (UCV) in both scenarios.

Scenario 1. UCV had the TLA system activated.

Scenario 2. UCV did not have the TLA system activated.

We defined the following research questions that were tested within the described framework.

(1) Is there a significant amount of time saved at the light, waiting, and as a consequence travel time from start to destination, by implementing and using TLA technology?

(2) Do TLA systems improve the efficiency of traffic flow?

(3) Do TLA systems reduce the amount of CO_2 emissions?

We additionally define the following null hypothesis:

H0: the TLA system that conveys the optimal velocity for approaching an intersection does not reduce the time waiting at a traffic light or the total time spent to reach the destination, nor does it result in an improvement of traffic flow.

The next section considers related work in simulation approaches. Section 3 describes the development of the simulation platform. Section 4 presents the implementation of the TLA system. Section 5 describes the experimental procedure to test the TLA. Section 6 reports on the performance and validation results of the developed system. Finally, Sections 7 and 8 discuss the results and conclude the paper.

2. Related Work

Several works have reported on the application of TLA services based on C-ITS standards. For example, the authors in [14] describe some significant steps for cooperative signaling that involve connecting to the city infrastructure to generate the service data. They argue that, in most European cities, traffic light controllers are equipped with inadequate hardware and do not allow proper communication with the traffic management center.

The authors in a further work [9] present and test a traffic light assistant prototype that combines V2X and mobile communication in real traffic. This TLA prototype extracts signal timing, tailback, geometry, and hazard information of intersections and combines it with internal vehicle data. Concepts for processing, management, and visualization of V2X and Traffic Signal Information (TSI) messages are also described.

In order to test V2V and V2I application developments in a controlled system, [15] combines a traffic and communication simulator with a multiuser driving simulator for a low-cost testing environment. In line with this research, we present in this paper an approach that simulates a visual TLA system in Unity 3D, using SUMO as the traffic simulator in order to test the research questions and hypotheses defined in the previous section.

Car manufacturers have already announced the first traffic light information systems based on V2I communication through 4G LTE connection [16]. The first real life experiments with V2I services were carried out at the end of 2017.

Regarding the effectiveness of the output modality to convey information related to traffic lights, both visual and auditory warnings have been studied in simulation frameworks. Most of these studies focused on visual warnings for displaying the messages created by the system. In [7], five different human machine interfaces (HMI) were evaluated. Each of the HMI provided the same information that was displayed in various manners. The study concluded that all the HMI contributed to augmenting situational awareness regarding the phase of the traffic light.

A further study that focused on conveying information based on TTC regarding a vehicle running an upcoming red light stated that visual warnings distracted drivers [17]. However, the relationship between auditory warning systems and driving behavior is still ambiguous and requires further investigation, although many visual ADAS systems are already integrated into vehicles and fully accepted by drivers.

TLA systems utilize real-time traffic light timing data by accessing the information directly from the traffic management center and combining it with GPS data. This makes it possible for vehicles to calculate exactly which traffic light they are approaching, whether that light is red or green, and how long it will be until it changes. The vehicle also calculates if it will be able to drive through an intersection with a green traffic light based on how fast it is moving. All of these data alert the drivers of any upcoming situations posed by the next intersection and help them to adjust their speed in order to avoid waiting at red lights.

Evaluation of new in-vehicle technology using simulators has been performed in different works. For example, in [18] a user interface for a novel traffic regulation system was assessed. The interface conveyed information to the driver which was based on the ubiquitous optimized management of individual intersections where physical devices were replaced by virtual traffic lights (VTL). However, only a few works address in-vehicle technology based on simulated environments with V2I capabilities, and as far as it is known to the authors, no approach of a visual TLA system has been implemented in Unity 3D using SUMO as the traffic simulator.

3. Simulation Framework Implementation

3.1. Generation of the Scenario Network. Using the traffic network defined in the XML SUMO network file, a complex network was created to be used as the scenario of the simulation. SUMO uses an X-Y coordinate system to locate all the elements of the network. The selected area was the Neubaugurtel Strasse in Vienna, as the amount of traffic lights on this road is significant. The source used for extracting data was Open Street Map (OSM) [19] since it allows the selection of areas of interest and automatically generates an .osm file. The desired area was defined manually as shown in Figure 1. In order to create the corresponding .net.xml file,

```
(1) <tlLogic id="joinedS_6" type="static"
(2) programID="0" offset="0">
(3)   <phase duration="86" state="GGG"/>
(4)   <phase duration="4" state="yyy"/>
(5) </tlLogic>
(6) <tlLogic id="joinedS_7" type="static"
(7) programID="0" offset="0">
(8)   <phase duration="88" state="GGGGGG"/>
(9)   <phase duration="2" state="yyyyyy"/>
(10) </tlLogic>
```

Algorithm 1: Errors in the TLS program.

Figure 1: Definition of the area from Open Street Map used in the experimental scenario.

the application NETCONVERT was used. Additional data included in OSM regarding buildings, parks, rivers, and so forth was visualized by using the POLYCONVERT tool.

To edit the network the tool NETEDIT was used, and the junctions in which there were a significant number of intersections were merged. Connections between roads were also edited.

3.2. Modification of TLS Programs for the Created Network. The .net.xml file generated from the NETCONVERT tool includes information about the traffic light systems (TLS). After verifying the .net.xml file, it was found that not all of the TLS programs with only one intersection had a red phase. An example of these errors can be found in the code in Algorithm 1.

In addition, some phases were longer than in reality. Therefore, this behavior was adapted for real conditions by shortening the length of the phases and/or adding missing red light phases. Once these two modifications were done, the edition of the network was completed.

3.3. Connection of SUMO with Unity 3D Using the TraCI Protocol. In order to create a connection between the SUMO and Unity 3D simulators, an interface was required to implement the communication. The communication protocol Traffic Control Interface (TraCI) was used to this end. TraCI's architecture is Client-to-Server, so that information that is required dynamically can be accessible to both SUMO and Unity 3D. SUMO acts as a TraCI server and it will listen

for incoming connections from external applications. Once SUMO has accepted the connection request, the TraCI client takes control of the simulation, which enables the client to affect the behavior of the simulation by sending commands to SUMO. As described in [20], in order to implement the TraCI protocol the existing library TraCI as a Service (TraaS) [21] was selected, as it allows remote control of SUMO from any programming language.

3.4. Scenario Set-Up. The Unity project created to evaluate the implemented TLA consisted of a set of folders, such as prefabs (vehicles and roads models), scripts, materials (characterization of models), Plug Ins (data base connection), and TraCI library (implementation of TraCI protocol). These folders were located in the Assets folder to be read.

As previously mentioned, we created an urban scenario to test the use cases with and without the TLA system implemented. Once the scenario was initialized, information about the simulated objects of interest such as buildings and their shape was retrieved. Additionally, real tree data were downloaded from the Vienna Municipal Departments 41 [22] and 42 and then imported into CityEngine as an object file, which was used to create the experiment's urban environment. An additional script was implemented to apply textures to the objects imported to CityEngine. Key buildings were thus recognized by the participants.

All the described parts constituted the final scenario that was imported as an .fbx file into Unity 3D as a GameObject. Once the urban environment was completed, a Mini Cooper vehicle was imported into the Unity 3D game engine and defined through scripts that also described the steering wheel and wheel movement.

4. Traffic Light Assistance Implementation

Vehicles in SUMO automatically follow a defined TLS schedule. In order to represent the traffic lights in Unity 3D, we used models of traffic lights and created a script to manipulate their behavior. Traffic lights are not defined as such in SUMO. Instead, SUMO uses TLS to control specific intersections. Consequently, one intersection might have more than one traffic light within a single TLS. Therefore, traffic lights were placed manually in their corresponding positions, and timing was adjusted in accordance with the TLS's schedule in SUMO.

We implemented a script that contained a matrix of Boolean variables describing the different modes of the traffic light. Each mode was defined by three variables, one per state, that were ordered as follows: red-yellow-green. Depending on the actual phase of the traffic light, the variable duration was modified according to the phase duration in SUMO. For changing the color of the traffic light, the three variables of the actual mode were verified. The color was set depending on which position of the matrix was true. The code in Algorithm 2 shows an example in which the traffic light state is green.

A TLA ADAS system was created and integrated into a UCV. This system computes the Green Light Optimal Speed Advisory (GLOSA) algorithm [5] shown in Algorithm 3 in order to calculate the optimal speed for approaching the next traffic light.

In order to compute this algorithm, several integral components of information about the TLS and the UCV were required. The information collected for each TLS was as follows:

(i) Time until next green phase.

(ii) Current state of the TLS.

(iii) Complete GreenYellowRed program of the TLS.

(iv) Links controlled by the TLS.

SUMO defines a fixed-time program for each TLS, and it can be either static or actuated. TLS programs describe the behavior of all the traffic lights placed at an intersection. For example, in the following state definition of a TLS phase, $<phaseduration = 25 state = RRggggRr/>$ each character in the string within a phase's state describes the state of one link of the TLS (g for green, y for yellow, and r for red). One lane may contain more than one link, for example, one for vehicles turning right and one for vehicles continuing straight.

The state of a TLS is defined as a string of characters. Each character describes one link controlled by the TLS. When the position of the character and the lane at which the vehicle is located match, the state is verified. Then, two variables called $t$min and $t$max are defined. Tmin contains the minimum time until the traffic light is green, defining the exact moment at which it would change from red to green. $t$max contains the time at which the next green phase will end.

Based on the performed calculations several messages regarding a recommended velocity for the UCV were displayed in the dash board screen. The messages displayed depended on the value returned by the TLA algorithm and the corresponding calculated velocity based on the distance of the UCV to the traffic light. If this distance was smaller than 80 meters, the GLOSA algorithm was computed and the correspondent message to reach the traffic light in the green phase from the list below was conveyed to the driver.

(i) Actual speed is optimal.

(ii) Speed needs to be decreased.

(iii) Speed needs to be increased according to the recommended range.

(iv) The system informs of a necessary stop at traffic light due to the long duration of the red phase in combination with the speed calculated by the algorithm.

Figure 2 shows one example for a message that recommends reducing velocity in order to reach the traffic light in the green phase.

5. Experimental Procedure

In order to test the performance of the TLA system, experiments were performed on drivers under lab-controlled conditions. After being welcomed, each participant was given a brief explanation of the tasks to perform in the simulator.

```
(1)  //-- Mode 0:
(2)  false, // direction 1 red
(3)  false, // direction 1 yellow
(4)  true, // direction 1 green
(5)  // Mode 0: Direction 1: Green
(6)    //
(7)    if (traffic_lights_mode == 0)
(8)    {
(9)    direction_time = 50; //
(10)     traffic_lights_counter += Time.deltaTime
(11)   * 2.0f;
(12)           if (traffic_lights_counter >= 2.0f
(13)           * direction_time)
(14)            {
(15)              traffic_lights_mode = 1;
(16)              traffic_lights_counter = 0.0f;
(17)            }
(18)     }
(19) //-- set direction 1 light green
(20)           if (traffic_lights_status_table
(21)           [(traffic_lights_mode * 3) + 2 +
(22) ((int)(crossway_direction) * 3)] == true)
(23)           {
(24)           traffic_light_renderer.transform
(25)           .GetComponent
(26)           <Renderer>().materials [4]. color =
(27)           new Color (1.0f, 1.0f, 1.0f, 1.0f);
(28)           if (use_real_lights == true)
(29)            {
(30)              lightsource_green1.
(31)              GetComponent<Light>()
(32)              .enabled = true;
(33)              if (traffic_light_type_single
(34)              == false)
(35)                {
(36)                lightsource_green2.
(37)                GetComponent<Light>()
(38)                .enabled = true;
(39)                }
(40)            }
```

Algorithm 2: Green light algorithm.

They then drove for five minutes in order to get familiarized with the simulation platform. The flow chart of Figure 3 depicts the experimental procedure. A total of 25 participants aged between 21 and 55 completed the experiment (mean age = 27.5, SD = 9.2, 64% male, 36% female). They were asked to follow the traffic rules and to not exceed the speed limit for an urban area of 40 km/h. All participants followed the route directions displayed on the dash board as they would with a navigation system.

In Scenario 1, the TLA ADAS system was not activated; therefore the data were logged for a representation of the baseline condition. Once the participants reached the end of the route, Scenario 2 with the activated TLA system was loaded and the corresponding data logged (see example in Figure 2). The duration of each phase was 15 minutes. When the experiment was completed, each participant was asked to fill out a questionnaire to rate their experience with the simulator and the efficiency of the system implemented. Figure 4 shows a participant during the experiment with the TLA system activated.

5.1. Data Acquisition and Processing. The pertinent data for testing the hypotheses were logged. First, the state and remaining phase time of the link at which the UCV was placed was retrieved and stored. Additional data related to the UCV regarding speed, distance, and time were logged and stored in a SQLite database. The data were then processed to calculate the main metrics from the simulation experiments: driver's average speed, total traveled time, delay or waiting time at traffic lights, CO_2 emissions, and number of stops.

In order to calculate the CO_2 emissions, a method was used relying on the speed-based model proposed by the National Institute for Land, Infrastructure and Transport (NILIM) [23]. This model is based on the size and weight

(1) Find the closest traffic light
(2) Calculate distance and time to traffic light T_{TL}
(3) Check phase at T_{TL}
(4) **if** GREEN **then**
(5) Continue trip
(6) Target speed (U_t) = U_{max}
(7) **else if** RED **then**
(8) Calculate remaining red time (T_{red})
(9) Calculate target speed for T_{red} + T_{TL}: U_t
(10) **else if** YELLOW **then**
(11) Calculate remaining yellow time (T_{yellow})
(12) Check for possible acceleration
(13) Calculate target speed for T_{yellow} + T_{red} + T_{TL}: U_t
(14) **end if**
(15) Advisory speed = max(U_t, U_{min}) & min(U_t, U_{max})

Algorithm 3: GLOSA algorithm [5].

Figure 2: TLA system conveying information regarding a decrease of the current UCV velocity.

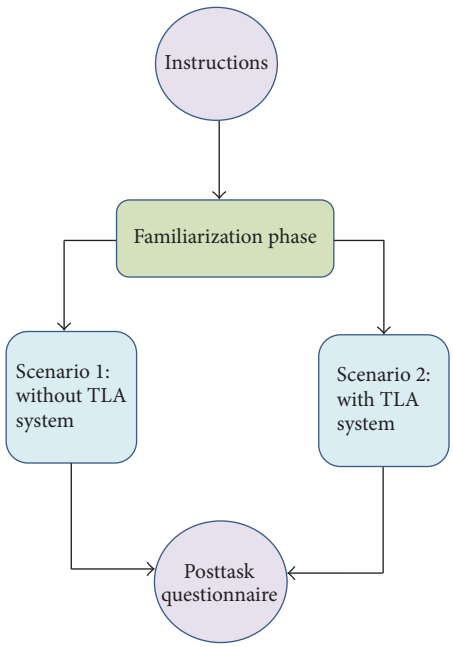


Figure 3: Flow chart of the experimental procedure.

Figure 4: Participant performing experiment with TLA system activated.

of the vehicle in question (small and large), the average speed of the vehicle, and the distance traveled. A Mini Cooper was used as the UCV in the simulation; therefore the approximation for a small vehicle was used to determine the CO_2 emissions as described in (1) [13].

(i) $EF_{CO2}(V_i)$ = emission factor of CO_2 per unit distance.
(ii) V_i = average travel speed of vehicle i.
(iii) a_1, a_2, a_3, and a_4 = parameter:

$$EF_{CO2}(V_i) = \frac{a_1}{v_i} + a_2 v_i + a_3 v_i^2 + a_4. \quad (1)$$

The resulting equation for the total amount of CO_2 emissions of a vehicle is as follows: $ECO2_i = EF_{CO2}(v_i)x_i$, where x is the total distance traveled. Figure 5 [13] shows the relationship between average speed and CO_2 emissions.

An approximation of the curve for the small vehicles was written into the C# program language in order to calculate the CO_2 throughout the evaluation of the simulator. The variables were logged into the database.

Regarding the collection of data related to the qualitative evaluation of the system, a Likert scale was used with values that ranged as follows: 1 = strongly disagree to 5 = strongly agree. The questionnaire asked the participants to rate the following four elements:

(1) Effect of the TLA on the driver's behavior
(2) Satisfaction with the messages displayed by the TLA system
(3) Performance of the TLA system
(4) Distraction level of TLA

Table 1: Driving performance depending on the activation of the TLA.

	W/o TLA		W/TLA		T-test ($\alpha = 0.05$)	
	Mean	*SD*	*Mean*	*SD*	*p*	*t(24)*
Travel time (s)	398	52.57	370.2	43.8	3.19	0.0038^{**}
Delay (s)	108.59	35.82	71.38	26.64	6.82	$4.69e-7^{***}$
Speed (kmh^{-1})	19.16	2.57	20.4	2.2	−2.88	0.008^{**}
CO_2 (mg)	411.91	28.12	406.33	22.11	1.19	0.24
Stops no.	13.6	3.88	8.88	4.4	5.66	$7.69e-6^{***}$

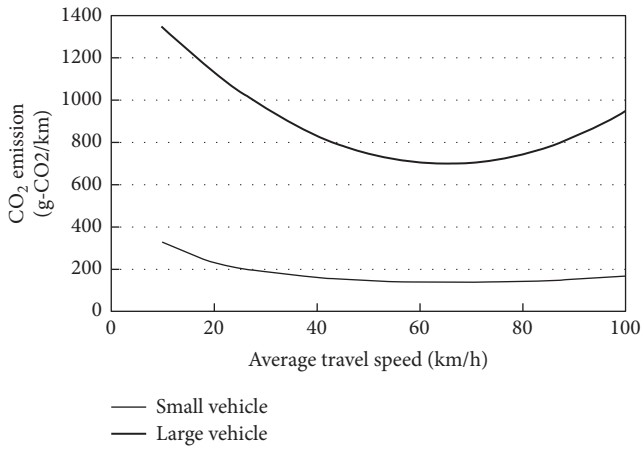


Figure 5: Relationship between average speed and CO_2 emissions according to [13].

5.2. Data Analysis. Metrics related to driving performance were compared using the paired t-test to determine whether the TLA system improved driver performance and to draw conclusions about the impact of the proposed ADAS system. The test calculates the difference within each before-and-after pair of measurements, determines the means of these changes, and reports whether this mean of difference is statistically significant.

The α level of significance used for the performance of the t-test was 0.05. The following steps were followed in order to test the validity of the null hypothesis [24]:

(i) Calculation of the sample difference mean ($\tilde{d}$) is as follows. In order to get this value, the difference between each value of the observed paired driving metrics from the different scenarios (X and Y) was calculated (see equation (2) [24]).

$$D_i = Y_i - X_i.$$
$$\tilde{d} = \frac{1}{N} * \sum_{i=0}^{n} D_i \quad (2)$$

(ii) Calculation of the standard deviation of the difference (S_d) in order to calculate the standard error of the mean difference (SE(d)) is as follows. SE(d) = S_d/N, where N is the strength of the sample.

(iii) Calculation of the t-statistics (T) is as follows. This calculation follows a t-distribution with ($N - 1$) degrees of freedom under the null hypothesis.

$$T = \frac{d}{\mathrm{SE}(d)}. \quad (3)$$

The T value is compared with the $T(N - 1)$ distribution using the table of the t-distribution, which gives the p value for the paired t-test.

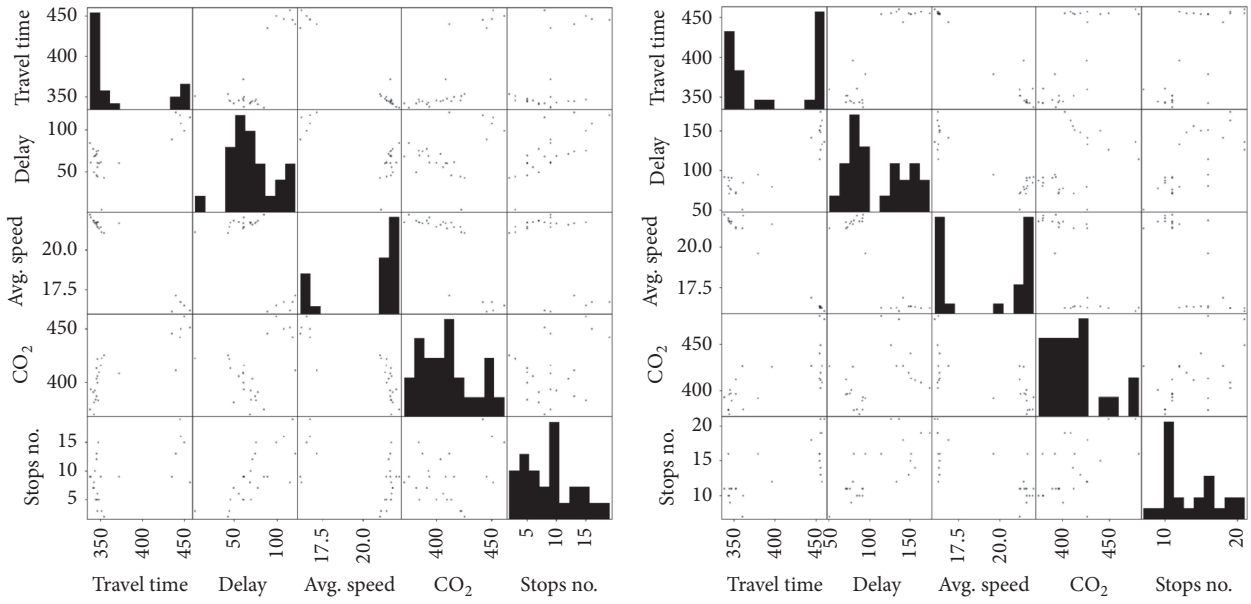
The p value determines whether the results are statistically significant or not. The null hypothesis can only be rejected when the p value obtained is smaller than the significance value (0.05).

6. Results

6.1. Quantitative Evaluation. The total time traveled, average speed, total time waiting at red lights, number of stops at traffic lights, and CO_2 emissions were the parameters measured during the experiments. Table 1 shows the comparison of the mean values and the standard deviation between the scenario without the TLA and the scenario with the TLA system. The average values of CO_2 emissions were very similar. However, there was a slight reduction of 3% with the TLA system activated. By implementing the TLA system the total time spent completing the route was remarkably reduced. This difference in time was statistically significant with respect to the one obtained without the TLA system. Regarding the number of stops at traffic lights, 20 traffic lights were placed throughout the scenario. Without the implementation of the TLA system, the average number of stops per participant was 13.6 (68% of the traffic lights), whereas with the help of the TLA system, the average number of stops was reduced to 8.88 (44.4%). Due to this fact, the total time spent waiting for the traffic light to change to its green phase was reduced in 34.22%.

Figure 6 shows how the activation of the TLA produces an increase in the average speed and a decrease in the rest of the indicators. Figure 7 shows the efficiency improvement by comparing the variables in both scenarios.

6.2. Correlations. Correlation matrices between the monitored indicators in Scenarios 1 and 2 are shown in Tables 2 and 3. The values correspond to the Spearman correlation coefficient between each pair of variables.



(a) Scenario 1 (with Traffic Light Assistance (TLA))

(b) Scenario 2 (without TLA)

Figure 6: Scatter plots for each pair of the results for both scenarios, showing the relationships between each pair of variables. The diagonal (the relationship between a variable and itself) represents the histogram of the variable on its own domain.

Table 2: Correlation matrix between the monitored indicators in Scenario 1.

	Total time	Delay	Avg. speed	CO_2	Stops no.
TotalTime	1.000000	0.757185	−0.986162	0.809822	0.558153
Delay	0.757185	1.000000	−0.730260	0.341748	0.748499
AveSpeed	−0.986162	−0.730260	1.000000	−0.820531	−0.516156
CO_2	0.809822	0.341748	−0.820531	1.000000	0.228826
Stops no.	0.558153	0.748499	−0.516156	0.228826	1.000000

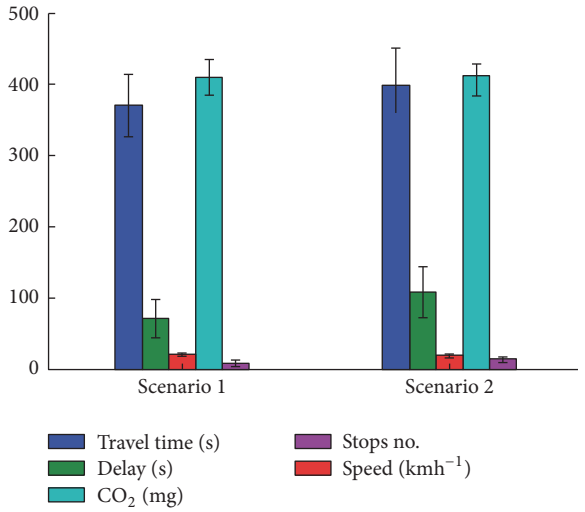
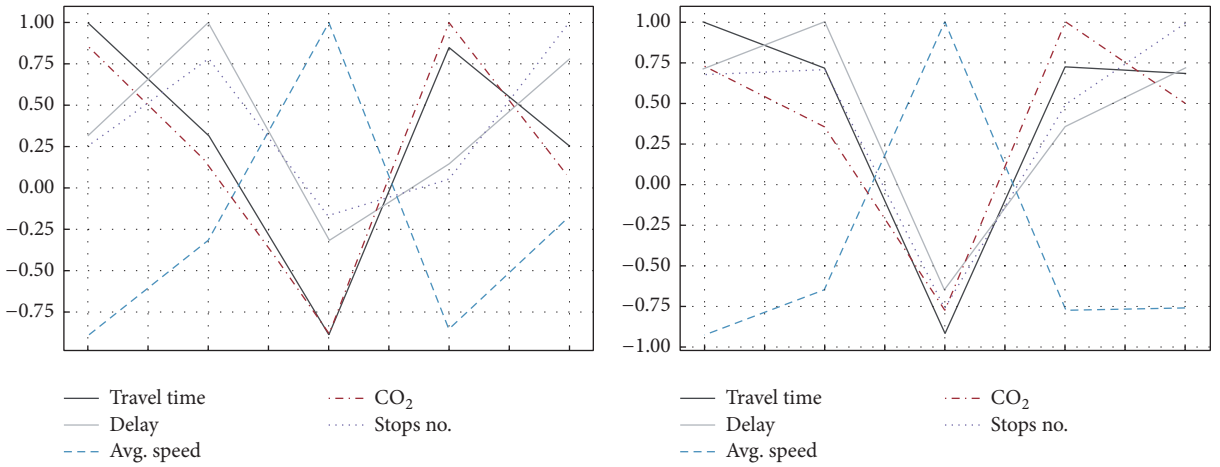


Figure 7: Comparison of the average variable values in Scenario 1 (with TLA) and Scenario 2 (without TLA).

Figure 8 depicts the correlations between the variables in Tables 2 (Scenario 1) and 3 (Scenario 2). Although both graphs seem to maintain similar correlations, Figure 8(a) shows a decrease in the impact of the average velocity on CO_2 emissions in Scenario 1. This behavior is a consequence of the TLA activation, and it has an effect on the decrease of delays, producing on the one hand an increase in average speed and, on the other hand, more constant regimes that reduce acceleration and jerks, therefore lower emissions and fuel consumption.

6.3. Qualitative Evaluation. Results regarding the qualitative evaluation showed that almost all of the participants were able to understand the messages displayed by the system. 3 participants (12%) did not follow the recommendations of the system because they were engaged in observing the surrounding scenario. Seven participants (28%) felt distracted by the TLA system, the mean value obtained for the subjective distraction level being 3.08 (Likert scale range 1 = strongly disagree to 5 = strongly agree).

The participants rated the system as helpful and useful, the mean value obtained being 4.12. A graphical representation of the ratings obtained from the postquestionnaire is shown in Figure 9. Some extra comments and possible improvements to the TLA system were recorded. For example, one participant mentioned that speed advice messages could also be triggered as audio messages instead of visual



(a) Scenario 1 (with TLA)

(b) Scenario 2 (without TLA)

Figure 8: Graphical representation of the correlations depicted in Tables 2 (Scenario 1) and 3 (Scenario 2).

Table 3: Correlation matrix between the monitored indicators in Scenario 2.

	Total time	Delay	Avg. speed	CO_2	Stops no.
TotalTime	1.000000	0.894721	−0.980759	0.699022	0.734660
Delay	0.894721	1.000000	−0.907758	0.383627	0.725905
AveSpeed	−0.980759	−0.907758	1.000000	−0.675797	−0.755984
CO_2	0.699022	0.383627	−0.675797	1.000000	0.513207
Stops no.	0.734660	0.725905	−0.755984	0.513207	1.000000

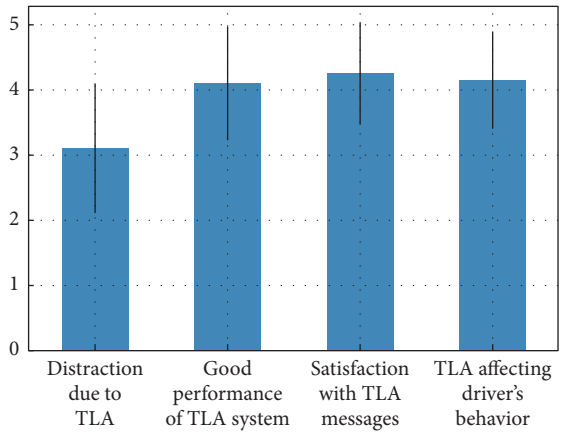


Figure 9: Ratings obtained from the qualitative postquestionnaire.

messages, even if the information conveyed would have to be expressed in a more complex manner. Additionally, 20% of the participants made the suggestion of placing the TLA system in a higher position, to have the messages at a more optimal point within the driver's field of view.

7. Discussion

The driving performance metrics measured and logged throughout the simulator experiments were total traveled time, delay or waiting time at traffic lights, average speed, CO_2 emissions, and number of stops.

The travel distance for each participant was around 2.1 km whereas, in [5], the travel distance was less than 1 km. The maximum speed permitted for driving was 40 km/h as the scenario was developed in an urban area. Drivers were focused on not exceeding this limit. At the beginning of the experiments, participants were familiarized with the pedals and steering wheel and became accustomed to the manner in which the messages that signaled the route were presented.

The mean number of stops without the TLA activated was significantly higher than with the system activated, the TLA system thereby having a significant impact on the drivers behavior.

With the activation of the TLA system, the waiting time or delay was drastically reduced by 34%. Previous literature showed a reduction of 30% [15]. Additionally, the total time of the journey was reduced because participants were able to avoid red lights. The total travel time was reduced by 9% when using the TLA system, meaning a shorter time than in [15]. The difference in the number of traffic lights, length of the path, and maximum permissive speed in both experiments were the factors which influenced the difference in total traveled time.

There was an interrelation between the travel time and the average velocity. With the implementation of the TLA system, driving performance metrics like waiting time at traffic lights or delay and total travel time were reduced in

comparison with the scenario without the proposed system, which resulted in an increase in the average speed.

Results regarding CO_2 emissions showed that the variation was proportional to the difference between the mean speeds obtained. Similar results were obtained in [15] regarding CO_2 and average speed.

The purpose of the postquestionnaire was to confirm that the proposed system was suitable for further experiments. Based on the opinion of the participants, the experience with the system was very satisfactory. The lowest value within the questionnaire was obtained from the level of distraction caused by the system. However, participants considered the messages clear and intuitive.

8. Conclusion and Future Work

We presented in this work the design and implementation of an in-vehicle TLA system which provides speed advice regarding optimal velocity when approaching an intersection controlled by a traffic light. In order to implement the TLA system, a driving simulator was developed. This simulation consisted of software components based on SUMO and Unity 3D.

The defined hypothesis was tested and according to the results, the alternative hypothesis *H1: the TLA system that conveys the optimal velocity for approaching an intersection reduces the time waiting at a traffic light and the total time spent to reach the destination and also improves traffic flow* was accepted.

The validation of the simulator was achieved through the high participant ratings. They did not find the system irritating and they described it as useful and helpful. In conclusion, the proposed system had positive effects on the drivers' behavior and its performance was satisfactory.

The potential of TLA systems has been demonstrated in this work. However, opportunities for extending the scope of this work beyond its current state remain. The UCV retrieves the data directly from the control management center that maintains control of the operation of the traffic lights. The development of a new scenario with traffic lights and attached RSU in order to directly communicate with vehicles will follow in future work. To this end systems that enable the preparation and execution of V2X simulations (i.e., VSimRTI [25]) can be used to take advantage of existing libraries to communicate with SUMO.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the "KiTSmart Project City of Vienna Competence Team for Intelligent Technologies in Smart Cities," Project no. 18-07 funded by national funds through the MA 23, Urban Administration for Economy, Work and Statistics, Vienna, Austria, and the EU Erasmus + framework program.

References

[1] C. Olaverri-Monreal, "Autonomous vehicles and smart mobility related technologies," *Infocommunications Journal*, vol. 8, no. 2, pp. 17–24, 2016.

[2] R. Subramanian and L. Lombardo, "Analysis of fatal motor vehicle traffic crashes and fatalities at intersections, 1997 to 2004," Tech. Rep., 2007, https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/810682.

[3] V. Astarita, V. P. Giofrè, G. Guido, and A. Vitale, "The use of adaptive traffic signal systems based on floating car data," *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 4617451, 13 pages, 2017.

[4] USDOT, "Traffic signal timing manual," 2017, https://ops.fhwa.dot.gov/publications/fhwahop08024/chapter5.htm.

[5] K. Katsaros, R. Kernchen, M. Dianati, and D. Rieck, "Performance study of a Green Light Optimized Speed Advisory (GLOSA) application using an integrated cooperative ITS simulation platform," in *Proceedings of the 7th International Wireless Communications and Mobile Computing Conference (IWCMC '11)*, pp. 918–923, Istanbul, Turkey, July 2011.

[6] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, 2011.

[7] M. Krause and K. Bengler, "Traffic light assistantevaluation of information presentation," *Advances in Human Aspects of Road and Rail Transportation*, p. 166, 2012.

[8] H. Breuer, "Urban mobility: radars key role on smart roads," 2017, https://www.siemens.com/innovation/en/home/pictures-of-the-future/mobility-and-motors/urban-mobility-radar-technology-for-highways.html.

[9] B. Bernais, A. Lotz, and H. Pu, "Design and implementation of a traffic light assistance system based on c2x and tsi messages," in *Proceedings of the AmE 2016-Automotive meets Electronics; 7th GMM-Symposium*, pp. 1–6, 2016.

[10] SUMO, "DLR - Institute of Transportation Systems - SUMO – Simulation of Urban MObility," 2016, http://sumo.dlr.de.

[11] Unity Technologies, "Unity - fast facts," 2016, https://unity3d.com/public-relations.

[12] Esri, "Esri cityengine - 3d modeling software for urban environments," 2017, http://www.esri.com/software/cityengine.

[13] K. Tsukui, H. Hanabusa, H. Oneyama, Y. Oshino, and M. Kuwahara, "CO2 and noise evaluation model linked with traffic simulation for a citywide area," *International Journal of Intelligent Transportation Systems Research*, vol. 7, no. 1, pp. 59–65, 2009.

[14] M. Zweck and M. Schuch, "Traffic light assistant: Applying cooperative ITS in European cities and vehicles," in *Proceedings of the 2013 2nd IEEE International Conference on Connected Vehicles and Expo, ICCVE 2013*, pp. 509–513, Las Vegas, NV, USA, December 2013.

[15] K. Gajananan, S. Sontisirikit, J. Zhang et al., "A cooperative ITS study on green light optimisation using an integrated traffic, driving, and communication simulator," in *Proceedings of the 36th Australasian Transport Research Forum, ATRF 2013*, October 2013.

[16] A. USA, "Audi announces the first vehicle to infrastructure (v2i) service - the new traffic light information system," 2017, https://www.audiusa.com/newsroom/news/press-releases/2016/08/audi-announces-first-vehicle-to-infrastructure- service.

[17] X. Yan, Q. Xue, L. Ma, and Y. Xu, "Driving-simulator-based test on the effectiveness of auditory red-light running vehicle warning system based on time-to-collision sensor," *Sensors*, vol. 14, no. 2, pp. 3631–3651, 2014.

[18] C. Olaverri-Monreal, P. Gomes, M. K. Silvéria, and M. Ferreira, "In-vehicle virtual traffic lights: a graphical user interface," in *Proceedings of the 2012 7th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–6, 2012.

[19] OpenStreetMap, "Openstreetmap Statistics Openstreetmap stats report," 2016, http://www.openstreetmap.org/stats/data_stats.html.

[20] C. Biurrun-Quel, L. Serrano-Arriezu, and C. Olaverri-Monreal, "Microscopic driver-centric simulator: Linking Unity3d and SUMO," in *Proceedings of the 5th World Conference on Information Systems and Technologies (WorldCist'17)*, vol. 569, pp. 851–860, Springer, Madeira, Portugal, 2017.

[21] "Traas - TraCI as a Service," http://traas.sourceforge.net/cms/.

[22] "Geodatenviewer der Stadtvermessung Wien," 2017, https://www.wien.gv.at/ma41datenviewer/public/.

[23] N. Oshiro, M. Matsushita, Y. Namikawa, and H. Ohnishi, "Fuel consumption and emission factors of carbon dioxide for motor vehicles," *Civil Engineering Journal*, vol. 43, no. 11, pp. 50–55, 2001.

[24] StatisticSolutions, "Statistic Solutions (M)Anova Analysis," 2017, http://www.statisticssolutions.com/manova-analysis-paired-sample-t-test/.

[25] "VSimRTI - Smart Mobility Simulation," 2017, https://www.dcaiti.tu-berlin.de/research/simulation/.

Hindawi
Journal of Advanced Transportation
Volume 2018, Article ID 2493401, 11 pages
https://doi.org/10.1155/2018/2493401

Research Article

Hybrid Optimization-Based Approach for Multiple Intelligent Vehicles Requests Allocation

Ahmed Hussein, Pablo Marín-Plaza, Fernando García, and José María Armingol

Intelligent Systems Lab (LSI) Research Group, Universidad Carlos III de Madrid (UC3M), Calle Butarque 15, Leganés, 28911 Madrid, Spain

Correspondence should be addressed to Ahmed Hussein; ahussein@ing.uc3m.es

Received 17 November 2017; Accepted 29 January 2018; Published 11 March 2018

Academic Editor: Aboelmaged Noureldin

Self-driving cars are attracting significant attention during the last few years, which makes the technology advances jump fast and reach a point of having a number of automated vehicles on the roads. Therefore, the necessity of cooperative driving for these automated vehicles is exponentially increasing. One of the main issues in the cooperative driving world is the Multirobot Task Allocation (MRTA) problem. This paper addresses the MRTA problem, specifically for the problem of vehicles and requests allocation. The objective is to introduce a hybrid optimization-based approach to solve the problem of multiple intelligent vehicles requests allocation as an instance of MRTA problem, to find not only a feasible solution, but also an optimized one as per the objective function. Several test scenarios were implemented in order to evaluate the efficiency of the proposed approach. These scenarios are based on well-known benchmarks; thus a comparative study is conducted between the obtained results and the suboptimal results. The analysis of the experimental results shows that the proposed approach was successful in handling various scenarios, especially with the increasing number of vehicles and requests, which displays the proposed approach efficiency and performance.

1. Introduction

Self-driving cars could revolutionize how people get around; with the introduction of automation into roads, it will contribute to solving the issues related to the traffic accidents, congestion, and energy consumption. The driverless vehicle technologies are advancing on a great scale, specifically the multiple sensors fusion techniques, deep learning, and computational intelligence. Together, they enable these vehicles to understand the nearby surroundings and take appropriate actions to navigate on their own from one point to another. The technology does not stop at vehicle being fully automated; however multiple of them cooperate together to achieve smoother driving operation [1–3].

Researchers in the Intelligent Transportation Systems (ITS) field are investing more time on the self-driving cars research and development [4–6]. However, during the last decade, Multirobot Systems (MRS) fell under the research attention of the ITS community. This increased interest comes from the significant advantages and higher potential provided by MRS over single robot systems. MRS can be simply understood to be a group of robots cooperating together for accomplishing a certain task or mission [7, 8]. With reference to the literature review survey, the coordination and cooperation among multiagent systems can be modeled as Multirobot Task Allocation (MRTA) problem. MRTA is a NP-hard problem, which concerns the use of the available resources in an efficient manner. Accordingly, the decision of which robot will do which task strongly affects the performance of the system [9, 10].

Accordingly, this paper introduces a hybrid optimization-based approach to solve the MRTA problem. This approach can coordinate among several intelligent vehicles and aid in the cooperation for better overall performance. The algorithm is implemented on automated golf carts and tested in off-road environments in real-life scenarios. Moreover, in order to prove the proposed solution efficiency, several experiments are carried out over well-known benchmarks. The benchmarks optimal solution is obtained after hundreds of hours of computations. The proposed approach was able to obtain a near-optimal solution in a matter of seconds. This proves the proposed approach high performance.

The remainder of the paper is organized as follows: Section 2 presents the previous work in the field of multiple automated vehicles and the allocation problem, followed by Section 3, which introduces the problem formulation, solution constraints, objective function, and the proposed solving hybrid approach. In Section 4, the used platform is described, along with the experimental setup in the designed architecture, selected scenarios, and the evaluation metrics. Results and discussion are illustrated in Section 5. Finally, the conclusion and future work are summarized in Section 6.

2. Related Work

Autonomous cars are becoming more frequent in nowadays scientific and industrial context, since Google launched their driverless car project in 2011 [4]. From then onwards, many other approaches proposed different architectures and solutions, all of them moving towards the development of the autonomous vehicle. For instance, Mercedes with the Bertha project proved the viability of autonomous vehicles in German roads based on advance sensing capabilities [11]. The V-Charge project researches in the direction of allowing automated valet parking for self-driving cars [12]. Moreover, several vehicle manufacturers have proposed different solutions in the field of autonomous vehicles which are close to markets, such as BMW and Audi [13], Mercedes-Benz [14], and Volvo [15]. Furthermore, several other proposals offer the possibility of including autonomous vehicles in public transportation systems [5, 6].

The self-driving technology is advancing rapidly, but in order to safely deploy vehicles on public roads, cooperation with other road users is mandatory. This cooperation would allow the safe interaction with other vehicles, whether with a human driver or driverless. Although some of the proposed solutions already handle the presence of other vehicles in the road [16–18], by handling static and dynamic obstacles and adapting the trajectory accordingly, the cooperative driving is not yet achieved. Cooperative driving demands the information exchange and control strategies deployed in all the vehicles involved following a two-way communication. In this sense, during recent years many works have addressed this topic, trying to provide solutions based on different configuration and solutions. The Grand Cooperative Driving Challenge 2016 was created with the purpose of boosting cooperative automated vehicles in the form of a cooperative based competition [19]. On the one hand, authors in [20] presented an auction-based cooperative control for autonomous vehicles; on the other hand, authors in [21] proposed an approach to enhance common motion planning algorithms; this proposal allows cooperation with human-driven vehicles. Additionally, a novel concept is presented, based on a centralized strategy, using maneuver templates, which are formalized collaborative maneuvers, to select cooperative driving strategies [22]. All this work proves the importance of collaboration and communication among vehicles to allow a safe and efficient deployment of autonomous vehicles really in driving scenarios.

Since cooperation is essential among multiple vehicles on the road, coordination becomes the first issue to solve. In the MRS, the question of which robot is going to execute which task is answered through task allocation problem, which is commonly known as MRTA problem. By reviewing the literature, it was found that different optimization approaches have been used in order to solve the general task allocation problem and were also used in order to solve the MRTA problem. In [23], a mixed integer linear programming optimization approach was used in order to allocate heterogeneous robots for maximizing the coverage area of the regions of interest. In [24], a simulated annealing approach was used to solve the allocation of MRS through formulating the MRTA problem as multi-Traveling Salesmen Problem (mTSP). Then in [25], a market-based approach was proposed to solve the MRTA problem for heterogeneous robots and task formulated as mTSP. Moreover, the task allocation problem was also solved using hybrid optimization approaches such as the tabu search with random search method in [26] and tabu search with noising method in [27].

3. Methodology

In this section, the MRTA problem modeling and formulation are introduced, followed by the selected objective cost function. Afterward, the proposed algorithm is described, highlighting the main contribution. MRTA problem [7] is formulated to allocate multiple robots, in this case, vehicles, to numerous tasks, in this case, requests. The procedures are summarized as follows:

(1) Given a set of n vehicles, $V = \{V_1, V_2, \ldots, V_n\}$.

(2) Given a set of m requests, $R = \{R_1, R_2, \ldots, R_m\}$.

(3) Allocation of the requests to the vehicles occurs, $A : R \to V$.

(4) Output set S is the best allocation of the requests to the vehicles:

$$S = \{(V_1 \;\; R_1) \;\; (V_2 \;\; R_2) \;\; \cdots \;\; (V_k \;\; R_k)\} \quad \text{for } 1 \leq k \leq m. \tag{1}$$

(5) Allocation S minimizes or maximizes a certain objective function in order to get the best performance of the system.

3.1. Problem Formulation. A variant of mTSP is used to model MRTA problem. In the standard mTSP formulation, n nodes are defined with the edges distances and m salesmen are known. The salesmen are required to cover all the available nodes and return back to their starting node, such that each salesman makes a round trip. The mTSP can be formally defined on a graph $G = (V, E)$ where V is the set of n nodes and E is the set of edges. Let $c = (c_{ij})$ be the distance matrix associated with E. Assuming the more general case which is

an asymmetric mTSP, thus $c_{ij} \neq c_{ji}\ \forall (i,j) \in E$. The mTSP can be formulated as follows [9]:

$$x_{ij} = \begin{cases} 1 & \text{if edge } (i,j) \text{ is used in the tour} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$\text{minimize} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} \times x_{ij} \tag{3}$$

$$\sum_{j=2}^{n} x_{1j} = m \tag{4}$$

$$\sum_{j=2}^{n} x_{j1} = m \tag{5}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j = 2, \ldots, n \tag{6}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 2, \ldots, n \tag{7}$$

$$x_{ij} \in \{0,1\}, \quad \forall (i,j) \in E \tag{8}$$

$$\sum_{i \in S}\sum_{j \in S} x_{ij} \leq |\text{SubTour}| - 1, \quad \forall \text{SubTour} \subseteq V \setminus \{1\},\ \text{SubTour} \neq \phi, \tag{9}$$

where (3) represents the objective function which is the summation of the total distance traveled, (4) and (5) ensure that exactly m salesmen departed their starting node and returned back. Equations (6), (7), and (8) are the usual assignment constraints. Finally, (9) is the subtour elimination constraint.

The proposed formulation is extended and adapted to the problem, where vehicles represent the salesmen and requests represent the cities. Therefore, vehicles capabilities and requests requirements are considered and included in the mTSP implementation. The added features of the vehicles are capacity, velocity, energy, efficiency, and sensors, and the ones for the requests are timestamp, priority, and passengers.

Where the vehicle capacity is the maximum number of passengers that it can hold, the velocity is a representation of the maximum speed it can reach, the energy is a representation of the battery level, the efficiency is a representation of the aging factor, and finally the sensors are a set of on-board devices to consider the vehicles as heterogeneous robots. On the other hand, the request timestamp is the date and time of the request creation, the priority is a representation or the request urgency based on the request type, and passengers are a representation of the number of users for the request.

3.2. Solution Construction. The solution is constructed as a set that includes a list of all vehicles in the system, followed by their assigned requests. The order of the list defines the quality of the solution according to the objective function. For example, for a problem with three vehicles and five requests, one of the possible candidate solutions is represented as follows:

$$\text{Candidate Solution} = [V1 \ \ R1 \ \ R2 \ \ V2 \ \ R3 \ \ R4 \ \ V3 \ \ R5]. \tag{10}$$

Any combination of this list presents another candidate solution, and since the mTSP is a permutation problem, therefore the order of this list affects the quality of the solution as per the objective function. The order implies that each vehicle is going to execute all requests succeeding it. In candidate solution (10), requests 1 and 2 are executed by vehicle 1, requests 3 and 4 are executed by vehicle 2, and finally request 5 is executed by vehicle 3.

3.3. Objective Function. Although the MRTA problem is formulated as an instance of the mTSP, the same objective function of the mTSP previously explained in (3) cannot be straightforwardly used as the objective function for the MRTA problem. Therefore, some variations had to be introduced to the objective function of the mTSP in order to be effectively used for the MRTA problem [28].

There are three main variations of the MRTA problem objective function compared with the mTSP objective function. First, it is a multiobjective function instead of a single objective function, second, the variable to be minimized is the time rather than the distance, and, third, the time of the maximum subtour is minimized rather than the total time, thus dealing with it as a MinMax problem. Then, for k subtours and r requests in each subtour, the total traveling time is calculated as follows:

$$f(\mathbf{x}) = \arg\max_{j \in \{1,2,\ldots,k\}} \frac{\sum_{i=1}^{r-1} \text{distance}\left(\text{subtour}_{j_i}, \text{subtour}_{j_{i+1}}\right)}{\text{vehicle velocity}_j} + \arg\max_{j \in \{1,2,\ldots,k\}} \frac{\sum_{i=1}^{r} \text{execution time}\left(\text{subtour}_{j_i}\right)}{\text{vehicle efficiency}_j}. \tag{11}$$

3.4. Solution Constraints. Although any arrangements of the vehicles and requests solution set are considered as a candidate solution for the mTSP problem, this does not guarantee that this solution is feasible for the MRTA problem. Therefore, few constraints are applied to the obtained solution, to ensure its validity and feasibility, which means checking whether the vehicle capabilities and request requirements are matching.

The first constraint is related to the capacity; for example, a vehicle with a maximum capacity of 4 passengers cannot handle a request of 6 passengers in one go; thus the request is decomposed into several requests and reallocated accordingly. The next constraint is the energy; the vehicle battery level is always taken into consideration before the final allocation of the request, since if the vehicle does not have sufficient energy for a specific request, it is reallocated to another vehicle. Another constraint is the priority level of the request, which implies that some requests, maintenance request, for instance, must be executed first. Last but not least,

there is a constraint related to the mounted on-board sensors, which check if the request requires the presence of a specific sensor in the vehicle.

These applied constraints strongly affect the search space of the problem through decreasing the number of candidate solutions that can be accepted as feasible solutions. One may think that this decrease of the number of feasible solutions among the candidate solutions may make it easier for the applied algorithm to find the best solution than the case without the constraints. However, these applied constraints, in fact, make it more difficult and more time consuming to find the best solution. This is mainly because the algorithm will be visiting a large number of solutions that are candidate solutions of the mTSP but are not feasible to solve the MRTA problem.

3.5. Proposed Algorithm. The proposed solution is designed as metaheuristic optimization algorithm. It is a hybrid approach, which is based on both trajectory-based and population-based techniques. The trajectory-based one is the family of optimization techniques that use a single solution throughout the algorithm, in order to find the near-optimal solution. While the population-based one is the family of optimization techniques that iteratively transforms a set of solutions, in order to generate a new population of solutions with the aim of finding the near-optimal solution [29].

On the one hand, SA was selected as a trajectory-based approach, where the neighboring operator is randomly chosen at each iteration for diversity. The mutation operators are swapping, deletion and insertion, inversion, and scrambling. On the other hand, genetic algorithm (GA) was selected as the population-based approach, where the selected mutation operators are the same; however, additional crossover operators are selected, which are partially mapped crossover and order crossover.

The swapping operator chooses two random elements of the solution list and swaps them with each other. Deletion and insertion operator chooses a random element and deletes it from its current position and randomly inserts it in a new position. The inversion operator chooses two random positions and inverts the order of elements between these two positions. The scrambling operator picks two random positions and scrambles the elements between these two positions. Figure 1 illustrates an example of all proposed operators, where positions 1 and 6 are selected as the two random elements. At each iteration, one of the four mutators is randomly chosen in order to generate a neighboring solution of the current solution. The four methods vary in their diversification and intensification effect on the generated neighboring solution.

The crossover operator is the mimicking of the biological recombination between chromosomes, when some portion of the genetic material is swapped between chromosomes producing a new offspring chromosome. On the one hand, in the partially mapped crossover, two points are selected at random in both parents solutions and the offspring is created by exchanging the in-between chromosomes. On the other hand, in the order crossover a portion of one parent is mapped to a portion of the other parent; then from

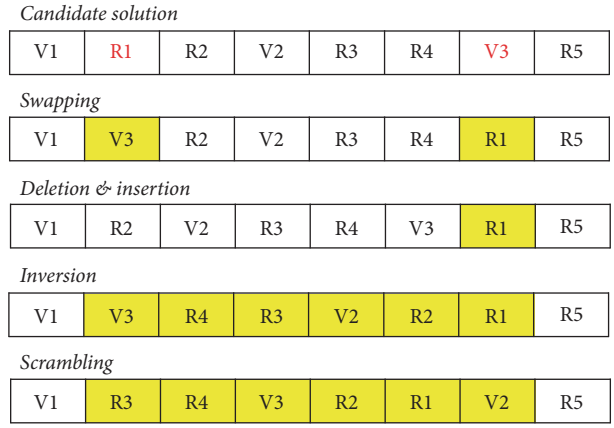


Figure 1: Example for the mutation operators.

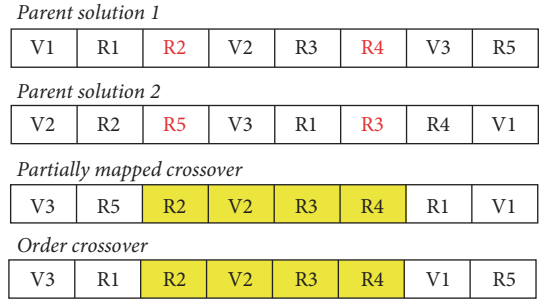


Figure 2: Example for the crossover operators.

the replaced portion onwards, the rest is filled up by the remaining genes, where already present genes are omitted and the order one is preserved. Figure 2 illustrates an example of proposed crossover operators, where positions 2 and 5 are selected as the two random elements. At each iteration, one of the two crossover mutators is randomly chosen in order to generate a neighboring solution of the current solution. Additionally, the random choice of the used operator gives the algorithm both the explorative and exploitative features that are useful in escaping local minimum and finding a better solution through searching in the neighbors of elite solutions, respectively.

The algorithm is used to solve the formulated model of the MRTA problem. Inputs are the list of the requests with their requirements, the list of vehicles with their capabilities, and the matrix of the distances between the points of interests of the requests locations. At each iteration of the algorithm, a solution is constructed to be evaluated; initially, it is random. The initial solution set is always filled with random elements of the requests list and the vehicles list until both lists are empty; the only constraint is that the start of the solution must be an element of the vehicles list. After the construction of the initial candidate solution or any other neighboring solution through the algorithm iterations, the feasibility of this solution must be checked, according to the solutions constraints that are previously explained in Section 3.4. As the algorithm progresses, neighboring solutions of the current solution must be generated in order to explore

the search space of the problem. Algorithm 1 presents the proposed algorithm used to solve the MRTA problem in this paper.

Here the parameters can be defined as follows:

(i) *parentsList* is the list with parents solutions.

(ii) *childrenList* is the list with children solutions.

(iii) *nextGenerationList* is the list with next generation solutions.

(iv) *iterationsNumber* is the number of iterations.

(v) *elitismPercent* is the elitism percent.

(vi) *populationSize* is the population size.

(vii) *initialTemperature* is the initial temperature.

(viii) *finalTemperature* is the final temperature.

(ix) *currentTemperature* is the current temperature.

(x) *iterationsPerTemperature* is the number of iterations per temperature decrement.

(xi) α is the geometric coefficient.

(xii) *transitionProbability* is the transition probability.

(xiii) *currentAllocation* is the current allocation solution.

(xiv) *currentCost* is the current solution cost.

(xv) *neighborAllocation* is the neighbor allocation solution.

(xvi) *neighborCost* is the neighbor solution cost.

(xvii) *bestAllocation* is the best allocation solution.

(xviii) *bestCost* is the best solution cost.

4. Experimental Work

In this section, the real-world platform is described, followed by the proposed ROS-based architecture for the multiple vehicle cooperation. Moreover, it presents the selected scenarios along with the chosen evaluation metrics.

4.1. Platform Description. In this work, two automated vehicles were selected as the platforms for the real-world experiments. Figure 3 shows the vehicles; they are part of the Intelligent Campus Automobile (iCab) project. They are electric golf carts, which are modified mechanically and electronically for the purpose of fully automated vehicles.

Each vehicle is equipped with multiple on-board sensors, such as GPS and compass module, stereo-camera, laser-rangefinder, 3D LiDAR, optical encoders, and ultrasonic sensors. All sensors are connected to the on-board embedded computers, which are connected to a 4G router with a constant Internet connection [30].

The on-board embedded computers use Robot Operating System (ROS) architecture to carry out numerous technologies. The architecture is divided into three layers. (1) Low-level commands are executed in the reactive layer, such as communicating with sensors and actuators. (2) Communication between the layers and decomposing the complex task to simple ones are executed in the sequencer layer. (3) Finally, high-level commands are executed in the deliberative layer, which is mainly responsible for environment perception, navigation, planning, localization, and mapping among others [31].

Figure 3: iCab 1 and iCab 2 automated vehicles platforms.

4.2. Communication and Cooperation Architecture. Communication and cooperation among a team of unmanned vehicles is a crucial task. Since in case of long duration missions, or vehicle failure, it is unlikely that a single vehicle is enough to complete the operation [10], therefore, for the communication among vehicles, the *multimaster_fkie* package is used [32]. It combines the required nodes to establish and manage a multimaster network over the implemented ROS-based architectures. In order to use the aforementioned package, vehicles must operate under a common network. Thus, for secure and stable paradigm, a Virtual Private Network (VPN) is used to overlay the commonly available one, which is implemented following the ITS Cooperative ITS (C-ITS) ETSI TR 101 607 V1.1.1 (2013-05) protocol. Detailed description for using the adopted communication schemes is explained in [33].

For the cooperation, *mrta* node is implemented as the one responsible for handling the task allocation problem. The inputs for the node are vehicle status topic, vehicle pose topic, and list of requests topic. The allocation output is published to the *task_executor* node of each vehicle. The idea of this node is to decompose the requests allocation into executable tasks, according to the vehicles capabilities. The objective is to execute the user transport requests by multiple automated vehicles, using coordination and cooperation mechanisms. Passengers use their own smartphones to create a request by selecting the number of the passengers and the pick-up and drop-off locations. The transport requests are then stored on a webserver, which is accessible by every vehicle in the system [34].

To avoid the centralized paradigm, leader token approach is proposed. Thus, at each iteration, the vehicles run a token selection algorithm, which determines the leader token holder for this loop. The algorithm decision is based on the vehicle status and current computational load. Only the leader vehicle communicates with the webserver for the updates of the requests lists and then publishes it to all other vehicles in the system. Accordingly, this mechanism ensures

Input: Requests list *requests*, Vehicles list *vehicles*, Distances matrix *distances*
Output: Best allocation *bestAllocation*

(1) **for** $i \leftarrow 1$ **to** *populationSize* **do**
(2) *parentsList* ← generateValidSolution(*requests*, *vehicles*, *distances*)
(3) **end**
(4) *currentAllocation* ← minimumOf(*parentsList*)
(5) **for** $i \leftarrow 1$ **to** *iterationsNumber* **do**
(6) **if** $i \leq 25\%$ of *iterationsNumber* **then**
(7) *elitismPercent* = 20%
(8) **else if** $i > 25\%$ of *iterationsNumberAND* $i \leq 50\%$ of *iterationsNumber* **then**
(9) *elitismPercent* = 30%
(10) **else if** $i > 50\%$ of *iterationsNumberAND* $i \leq 75\%$ of *iterationsNumber* **then**
(11) *elitismPercent* = 40%
(12) **else**
(13) *elitismPercent* = 50%
(14) **end**
(15) *childrenList* ← crossover(least 20% of *parentsList*)
(16) *childrenList* ← mutation(top 80% of *parentsList*)
(17) *nextGenerationList* ← minimum *elitismPercent* of *parentsList*
(18) *nextGenerationList* ← minimum *elitismPercent* of *childrenList*
(19) **for** $j \leftarrow 1$ **to** $(100\% - elitismPercent \times 2)$ of *iterationsNumber* **do**
(20) *currentAllocation* ← generateValidSolution(*requests*, *vehicles*, *distances*)
(21) *currentCost* ← getAllocationCost(*currentAllocation*)
(22) *bestCost* ← *currentCost*
(23) **while** *currentTemperature* < *finalTemperature* **do**
(24) **for** $i \leftarrow 1$ **to** *iterationsPerTemperature* **do**
(25) *neighborAllocation* ← generateNeighborSolution(*currentAllocation*)
(26) *neighborCost* ← getAllocationCost(*neighborAllocation*)
(27) **if** *neighborCost* < *currentCost* **then**
(28) *currentAllocation* ← *neighborAllocation*
(29) *currentCost* ← *neighborCost*
(30) **if** *neighborCost* < *bestCost* **then**
(31) *bestAllocation* ← *neighborAllocation*
(32) *bestCost* ← *neighborCost*
(33) **end**
(34) **else**
(35) **Generate**: random number *randomNumber* $\in]0, 1[$
(36) $transitionProbability = \exp\left(-\dfrac{neighborCost - currentCost}{currentTemperature}\right)$
(37) **if** *transitionProbability* > *randomNumber* **then**
(38) *currentAllocation* ← *neighborAllocation*
(39) *currentCost* ← *neighborCost*
(40) **end**
(41) **end**
(42) **end**
(43) *currentTemperature* = *currentTemperature* $* \alpha$
(44) **end**
(45) *nextGenerationList* ← *currentAllocation*
(46) **end**
(47) **if** minimumOf(*parentsList*) < *currentAllocation* **then**
(48) *currentAllocation* ← minimumOf(*parentsList*)
(49) **end**
(50) *parentsList* ← *nextGenerationList*
(51) **end**
(52) *bestAllocation* ← *currentAllocation*

Algorithm 1: Proposed hybrid optimization-based algorithm.

the consistency and synchronization of the requests through a distributed paradigm.

4.3. Selected Scenarios. In order to test the proposed approach, several scenarios are selected in both simulation and real-world. The simulation scenarios are selected from well-known benchmarks of mTSP; this is in order to have the optimal cost available for comparison. Each scenario consists of a different number of cities, which are distributed over the environment in various locations.

Figure 4 shows the four selected scenarios, where the top left one is Christofides/Eilon with 51 cities, top right one is Berlin (Groetschel) with 52 locations, bottom left one is Christofides/Eilon with 76 cities, and bottom right one is Rattled Grid (Pulleyblank) with 99 cities [35, 36]. Each scenario has only one depot, represented by the red marker in the graphs.

On the other hand, the real-world scenario was designed in a way to evaluate the functionality of the proposed solution and the architecture in the platforms. The scenario involved three users using the application to create transportation requests. The three users had different starting points and different destinations; moreover the request time was close to others to evaluate how the vehicles are going to respond. Figure 5 shows the environment map with the vehicles and passengers locations, the two vehicles are represented with the golf-cart clip-art, and the passengers representation is marked in three different colors, with a circular shape of the same color for the desired destination. The experiment was video recorded and its results discussion is in the next section.

4.4. Evaluation Metrics. The proposed hybrid optimization-based solution is used to solve each scenario of the MRTA problem and the results are recorded for evaluation. In order to evaluate the quality, two evaluation metrics are introduced, which are the allocation cost and computational time.

The first evaluation metric is the allocation cost of the best allocation found. The allocation cost is calculated based on the objective function. Thus two allocations costs are computed, one is the MinMax cost, which represents the length of the longest subtour in the allocation, and the other is the total overall cost of all subtours.

The second metric is the computational time required by the algorithm to reach the best solution. The timer starts after the databases of the vehicles and requests are read and stops when the algorithm stops; then the elapsed time is reported. Since the computational time calculation is based on the machine, the computer used for all experiments has the specifications in Table 1.

Table 1: System specifications.

Processor	4 cores, 4 threads @3.8 GHz
Memory	16 GB DDR4-2133
Storage	HDD 1 TB 7200 RPM @3 Gb/s

5. Results

In this section, a comparative study is conducted between the proposed approach results and the reference optimal results presented in [35, 36]. The optimal results are obtained for the four well-known benchmarks, which are described in the scenarios subsection. The authors in [35] adjusted the benchmark settings, to have all vehicles located at the depot, where the vehicles are required to visit all locations in the scenario. With the condition that each location is visited exactly once, vehicles must return to the depot afterward.

Table 2 summarizes all the results from both the optimal solutions costs, compared to the obtained solutions costs, in addition to the computational time for the obtained solutions costs. The optimal costs are obtained using IBM CPLEX; it took 96 hrs for the eil-51 dataset, 120 hrs for the berlin-52 dataset, 168 hrs for the eil-76 dataset, and 216 hrs for the rat-99 dataset. On the other hand, the reported values of the obtained costs are the mean of the 25 experiments of each scenario.

These results show that the proposed approach was able to converge to the near-optimal values in much less computational time, which can actually be considered running in real-time. For the MinMax costs, Table 3 shows the deviation errors to the optimal costs, presenting the fact that all errors are less than 10%. However, the more visible contribution is that the error is actually decreasing with the increasing number of vehicles. This proves that the proposed approach is more capable of handling multiple vehicles and obtaining more accurate solutions in all tested benchmarks.

Since there are several possibilities for the solutions permutations, when the MinMax solution is optimized, this does not guarantee the optimization for the total cost as well. However, the proposed approach objective function was designed to take this into consideration. Table 4 shows the deviation errors from the optimal total costs. The same behavior of the MinMax costs is quite obvious, where the more the vehicles introduced to the system, the more capable the proposed approach to find near-optimal allocations. Moreover, in the case of berlin-52 dataset, the proposed approach obtained allocations better than the suboptimal ones in the case of 5 and 7 vehicles.

The conducted comparative study highlighted the high performance of the proposed approach in different scenarios and its scalability in handling multiple vehicles. Therefore, the proposed approach was able to not only obtain near-optimal allocations in much less computational time, but also outperform the CPLEX approach in one of the scenarios, under the same set of constraints and conditions.

Additionally, the real-world experiment was successful in terms of allocation of the requests and the vehicles performance to pick up and drop off the passengers, along with the automated navigation from the starting points to the destinations. The two iCab platforms were connected via the V2V communication and once the first request was created on the webserver, the token-holder vehicle collected the requests and the available vehicles to start the proposed task allocation algorithm. The first request was allocated to iCab-1 and while the vehicle is navigating to the first destination point, the second request came in, at which the allocation algorithm output was to allocate it to iCab-1, since this

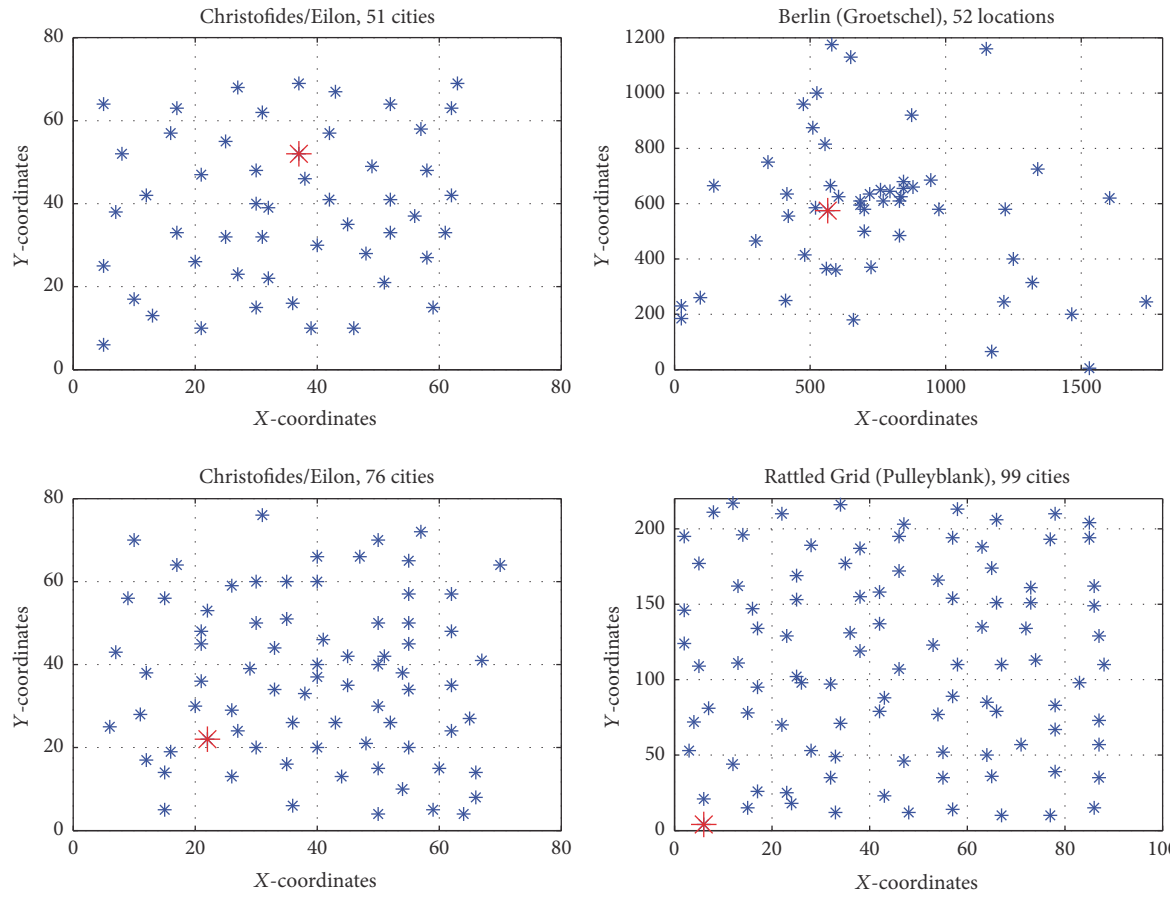


Figure 4: mTSP selected benchmark scenarios. Red marker is the depot.

Table 2: Benchmarks comparative results.

Benchmark	Vehicles	Optimal MinMax	Optimal total	Obtained MinMax	Obtained total	Comp. time (sec)
eil-51	2	222.73	444.33	236.49	470.97	124.86
	3	159.57	477.15	167.11	498.64	137.06
	5	123.96	615.19	129.36	629.64	158.86
	7	112.07	762.83	116.87	764.24	182.67
berlin-52	2	4110.21	8217.94	4315.83	8630.19	131.94
	3	3244.37	9591.15	3387.35	10115.18	140.97
	5	2441.39	12084.90	2509.84	**11969.61**	163.10
	7	2440.92	16768.79	2491.83	**15628.95**	194.21
eil-76	2	280.85	561.48	307.68	613.75	281.89
	3	197.34	587.65	210.97	632.07	292.83
	5	150.30	748.43	158.24	772.34	333.45
	7	139.62	964.69	143.74	970.42	386.62
rat-99	2	728.71	1456.95	801.68	1603.31	442.96
	3	587.17	1751.95	636.41	1903.04	495.36
	5	469.25	2336.22	487.71	2399.87	555.85
	7	443.91	3074.30	462.59	3135.78	645.43

Figure 5: Environment map with the real-world scenario vehicles and passengers locations.

Table 3: Deviation errors to optimal MinMax costs.

Benchmark/vehicles	eil-51	berlin-52	eil-76	rat-99
2 vehicles	6.18%	5.00%	9.55%	10.01%
3 vehicles	4.73%	4.41%	6.90%	8.39
5 vehicles	4.36%	2.80%	5.29%	3.93%
7 vehicles	4.28%	2.09%	2.95%	4.21%

Table 4: Deviation errors to optimal total costs.

Benchmark/vehicles	eil-51	berlin-52	eil-76	rat-99
2 vehicles	6.00%	5.02%	9.31%	10.05%
3 vehicles	4.50%	5.46%	7.56%	8.62
5 vehicles	2.35%	**−0.95%**	3.19%	2.72%
7 vehicles	0.18%	**−6.80%**	0.59%	2.00%

optimizes the overall solution. Therefore, iCab-1 continues the navigation to the first destination point and then picks up the second request to continue afterward to the second destination point. The last request was communicated while the vehicle is navigating and this time the allocation algorithm assigned it to iCab-2, which was free and able to pick up the passenger. All computations are executed on the on-board vehicles computers in real-time, and since the problem was addressing 2 vehicles and 3 requests in an environment with 12 points of interests, the computational time was approx. 1860 milliseconds for each allocation solution.

6. Conclusion and Future Work

In recent years, the MRS has received a significant consideration from various researchers in the ITS field. This is due to the fact that MRS can improve the concept of cooperative driving of automated vehicles. Moreover, since the number of automated vehicles on the public roads is increasing, the necessity of MRS is also increasing. One of the main challenges for the MRS is the task allocation, to identify which task should be executed by which vehicle, and it is commonly known as MRTA problem. In this work, the focus on the MRTA problem was from the aspect of transportation requests, where users make requests through a mobile application to be driven from one point to another; then an automated vehicle in the surrounding takes the job and so on. The idea of task allocation can be extended to much more than that to improve the quality of cooperative driving.

The main intentions of this work are to propose a generic approach for solving MRTA problem, which not only should be responsible for providing a solution that is feasible, but also should be optimized. Since the optimized allocation enables the appropriate use of all available resources and thus increasing the overall system performance and decreasing the costs, accordingly, a hybrid metaheuristic optimization-based approach is proposed, which combines both the simulated annealing and the genetic algorithm approaches, to ensure the exploitation and exploration aspects of the search space and obtain near-optimal solutions in less computational time.

Several test scenarios were implemented in order to evaluate the efficiency of the proposed approach. These scenarios are based on well-known benchmarks; thus a comparative study is conducted between the obtained results and the suboptimal results provided in the previous work over the same benchmarks. The analysis of the experimental results shows that the proposed approach was successful in handling various scenarios, especially with the increasing number of vehicles and requests, which displays the proposed approach efficiency and scalability. Additionally, the proposed approach was tested on automated vehicles platforms in a real-world scenario, at which the vehicles performed the assigned tasks flawlessly, which proves the functionality of the approach outside the simulation environment.

While the proposed approach was successful in solving the MRTA problem and optimizing the solutions of the benchmarks in significantly less time, there are some aspects that should be considered for future improvement. The cooperation architecture is designed to handle heterogeneous tasks, which means that tasks have specific requirements that constrain specific vehicles to complete it. Moreover, heterogeneous vehicles, not only in terms of capabilities, but also in terms of type, for instance, carry out experiments with both ground and aerial automated vehicles. Last but not least, it was assumed that the distance matrix includes the shortest free paths between the points of interests for the tasks; however, since the vehicles have already on-board trajectory planning algorithms, the approach should consider incorporating this work in the distance calculation.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper. The iCab platforms, which were utilized in the real-world experiments, are property of Intelligent Systems Lab (LSI) Research Group at Universidad Carlos III de Madrid (UC3M), which is the affiliation of all authors.

Acknowledgments

This research is supported by Madrid Community Project SEGVAUTO-TRIES (S2013-MIT-2713) and by the Spanish Government CICYT Projects (TRA2015-63708-R and TRA2016-78886-C3-1-R).

Supplementary Materials

The recorded video for the real-world experiment, which was described in Section 4.3, is included. The video is played at 3x of the normal speed. *(Supplementary Materials)*

References

[1] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, and H. Fujii, "Vehicle Control Algorithms for Cooperative Driving with Automated Vehicles and Intervehicle Communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 3, pp. 155–160, 2002.

[2] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 112–121, 2014.

[3] D. L. Bock, D. Kettles, and J. Harrison, "Automated, autonomous and connected vehicle technology assessment," *Electric Vehicle Transportation Center (EVTC '16)*, 2016.

[4] S. L. Poczter and L. M. Jankovic, "The google car: driving toward a better future," *Journal of Business Case Studies*, vol. 10, no. 1, pp. 1–7, 2014.

[5] A. Y. S. Lam, Y.-W. Leung, and X. Chu, "Autonomous vehicle public transportation system," in *Proceedings of the 3rd International Conference on Connected Vehicles and Expo, ICCVE 2014*, pp. 571–576, Austria, November 2014.

[6] P. Khaligh and U. Weidmann, "A conceptual framework for the interactions of autonomous public transport systems and urban planning guideline," in *Proceedings of the Swiss Transport Research Conference*, pp. 1–18, 2016.

[7] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[8] L. E. Parker, *Multiple Mobile Robot Systems*, Springer, 2008.

[9] M. Badreldin, A. Hussein, and A. Khamis, "A Comparative Study between Optimization and Market-Based Approaches to Multi-Robot Task Allocation," *Journal of Advances in Artificial Intelligence*, vol. 2013, pp. 1–11, 2013.

[10] A. Khamis, A. Hussein, and A. Elmogy, "Multi-robot task allocation: A review of the state-of-the-art," *Cooperative Robots and Sensor Networks*, vol. 604, Springer International Publishing, pp. 31–51, 2015.

[11] J. Ziegler, P. Bender, M. Schreiber et al., "Making bertha drive-an autonomous journey on a historic route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014.

[12] P. Furgale, U. Schwesinger, M. Rufli et al., "Toward automated driving in cities using close-to-market sensors: An overview of the V-Charge Project," in *Proceedings of the 2013 IEEE Intelligent Vehicles Symposium, IEEE IV 2013*, pp. 809–816, Australia, June 2013.

[13] DailyNews, "Bmw, audi push self-driving cars closer to reality," in *Daily News - Autos*, 2013.

[14] A. Bragman, *Mercedes-benz tech brings cars closer to self driving*, USA Today, 2016.

[15] L. Laursen, "Volvo to test self-driving cars in traffic," *IEEE Spectrum - Tech Talk*, 2013.

[16] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A Review of Motion Planning Techniques for Automated Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016.

[17] P. Bender, O. S. Tas, J. Ziegler, and C. Stiller, "The combinatorial aspect of motion planning: Maneuver variants in structured environments," in *Proceedings of the IEEE Intelligent Vehicles Symposium, IV 2015*, pp. 1386–1392, Republic of Korea, July 2015.

[18] X. Qian, F. Altché, P. Bender, C. Stiller, and A. De La Fortelle, "Optimal trajectory planning for autonomous driving integrating logical constraints: An MIQP perspective," in *Proceedings of the 19th IEEE International Conference on Intelligent Transportation Systems, ITSC 2016*, pp. 205–210, Brazil, November 2016.

[19] C. Englund, L. Chen, J. Ploeg et al., "The Grand Cooperative Driving Challenge 2016: Boosting the introduction of cooperative automated vehicles," *IEEE Wireless Communications Magazine*, vol. 23, no. 4, pp. 146–152, 2016.

[20] H. Rewald and O. Stursberg, "Cooperation of autonomous vehicles using a hierarchy of auction-based and model-predictive control," in *Proceedings of the 2016 IEEE Intelligent Vehicles Symposium, IV 2016*, pp. 1078–1084, Sweden, June 2016.

[21] M. Naumann and C. Stiller, "Towards cooperative motion planning for automated vehicles in mixed traffic," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–6, 2017.

[22] S. Manzinger, M. Leibold, and M. Althoff, "Driving strategy selection for cooperative vehicles using maneuver templates," in *Proceedings of the 28th IEEE Intelligent Vehicles Symposium, IV 2017*, pp. 647–654, USA, June 2017.

[23] N. Atay and B. Bayazit, "Mixed-integer linear programming solution to multi-robot task allocation problem," Tech. Rep., Washington University, vol. 54, St. Louis, USA, 2006.

[24] A. R. Mosteo, *Multi-robot Task Allocation for Service Robotics: From Unlimited to Limited Communication Range*, Universidad de Zaragoza, 2010.

[25] A. Hussein and A. Khamis, "Market-based approach to Multi-robot Task Allocation," in *Proceedings of the 2013 International Conference on Individual and Collective Behaviors in Robotics, ICBR 2013*, pp. 69–74, Tunisia, December 2013.

[26] W. Kmiecik, M. Wojcikowski, L. Koszalka, and A. Kasprzak, "Task allocation in mesh connected processors with local search meta-heuristic algorithms," *Lecture Notes in Computer Science*, vol. 5991, Springer Berlin Heidelberg, no. 2, pp. 215–224, 2010.

[27] W. Chen and C. Lin, "A hybrid heuristic to solve a task allocation problem," *Computers & Operations Research*, vol. 27, no. 3, pp. 287–303, 2000.

[28] A. Maji and M. K. Jha, "Multi-objective highway alignment optimization using a genetic algorithm," *Journal of Advanced Transportation*, vol. 43, no. 4, pp. 481–504, 2009.

[29] A. Hussein, P. Marin-Plaza, F. Garcia, and J. M. Armingol, "Optimization-based approach for cooperation and coordination of multi-autonomous vehicles," in *Proceedings of the International Conference on Computer Aided Systems Theory (Eurocast)*, pp. 286-287, 2017.

[30] A. Hussein, P. Marin-Plaza, D. Martin, A. De La Escalera, and J. M. Armingol, "Autonomous off-road navigation using stereo-vision and laser-rangefinder fusion for outdoor obstacles detection," in *Proceedings of the 2016 IEEE Intelligent Vehicles Symposium, IV 2016*, pp. 104–109, Sweden, June 2016.

[31] P. Marín-Plaza, J. Beltrán, A. Hussein et al., "Stereo vision-based local occupancy grid map for autonomous navigation in ros," in *Proceedings of the Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, vol. 3, pp. 703–708, Rome, Italy, Feburary 2016.

[32] S. H. Juan and F. H. Cotarelo, *Multi-master ros systems , Institut de Robotics and Industrial Informatics [master, thesis]*, 2015.

[33] A. Hussein, P. Marin-Plaza, F. Garcia, and J. M. Armingol, "Autonomous cooperative driving using v2x communications in off-road environment," in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC '17)*, pp. 1–6, 2017.

[34] A. Kokuti, A. Hussein, P. Marin-Plaza, A. de la Escalera, and F. Garcia, "V2X communications architecture for off-road autonomous vehicles," in *Proceedings of the 2017 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pp. 69–74, Vienna, Austria, June 2017.

[35] R. Necula, M. Breaban, and M. Raschip, "Tackling the Bi-criteria facet of multiple traveling salesman problem with ant colony systems," in *Proceedings of the 27th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2015*, pp. 873–880, Italy, November 2015.

[36] R. Necula, M. Breaban, and M. Raschip, "Performance evaluation of ant colony systems for the single-depot multiple traveling salesman problem," in *Proceedings of the 10th International Conference on Hybrid Artificial Intelligent Systems, HAIS 2015*, pp. 257–268, Spain, June 2015.

Hindawi
Journal of Advanced Transportation
Volume 2018, Article ID 5278565, 14 pages
https://doi.org/10.1155/2018/5278565

Research Article

Leader-Follower Based Locally Rigid Formation Control

Krishna Raghuwaiya, Bibhya Sharma, and Jito Vanualailai

School of Computing, Information & Mathematical Sciences, The University of the South Pacific, Suva, Fiji

Correspondence should be addressed to Bibhya Sharma; bibhya.sharma@usp.ac.fj

Received 4 August 2017; Accepted 5 December 2017; Published 25 February 2018

Academic Editor: Fernando García

This paper addresses motion control of a cooperative intelligent transport system (C-ITS) of nonholonomic mobile robots navigating a dynamic environment while maintaining a locally rigid formation. We consider the design of acceleration-based control inputs that govern the motion of cooperative intelligent transport system (C-ITS) using the artificial potential fields method for the avoidance of obstacles and attraction to designated targets. The control scheme utilizes a new leader-follower strategy using Cartesian coordinates to accomplish a collision-free locally rigid formation of an autonomous and intelligent transportation system. The concepts of virtual parking bays and *minimum distance technique* (MDT) are utilized to attain prescribed orientations of the formation at the final destination. The robustness of the control scheme is established by considering the effect noise on the formation, while the effectiveness of the proposed nonlinear control laws is demonstrated through computer simulations.

1. Introduction

Connected and autonomous vehicles play a huge role in the transportation industry worldwide and will govern the industry in the coming years. The connectedness is governed through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, which is facilitated using wireless and cellular technologies at current times. Such key communications of the *cooperative intelligent transport system* (C-ITS) allow respective shareholders to utilize information and coordinate task transfers in a cooperative manner. The networked V2V and V2I interactions deliver substantial social benefits in terms of safer, traffic-efficient, cooperative autonomous driving and energy-efficient transportation systems on our roads and highways [1, 2].

Cooperative tasks are more efficiently performed with desired robustness using multiple robots, which may not be possible with single robots. The benefits become apparent when considering distributed tasks, dangerous or hazardous tasks, and tasks that contain redundancies and when providing flexibility to task execution and robustness of systems [3–8]. In addition, there is the added advantage that working with multiple robots is less expensive than with just one specialized robot. To date there have been various applications of the control of C-ITS, for example, collaborative mapping and planning, search and rescue operations, flocking and schooling, transportation of large objects, undersea and space exploration, target seeking, competitive games, service robots, and military reconnaissance and surveillance [4, 8–13]. The ability of C-ITS to autonomously navigate in stable configurations while avoiding obstacles and collisions is also central to real-world applications [8, 14]. In many applications, the execution of the task requires formation control [3], and the accomplishment of the overall operation depends on each mobile robot operating in a prescribed manner. Typical examples include carrying a heavy load on roads and highways, hunting, and enclosing on an enemy to name a few.

The concept of formation control of mechanical systems, an integrated branch of motion planning and control of C-ITS, is receiving unprecedented attention from researchers all over the world for real-world applications [14]. The basic idea of formation control algorithms is to ensure that a group of mobile robots move effectively as a whole to jointly perform certain task(s). Examples of formation control tasks include delegation of feasible formations, establishing formations, maintaining and mobilizing different formation shapes, and switching between formations [15]. There are numerous approaches in literature in relation to the observance of a prescribed formation of a flock during motion [14]; however,

split/rejoin maneuvers and rigid formations are the prevalent approaches. The split/rejoin maneuvers are frequently cited in flocks of birds, swarms of insects and ants, and herds of animals. The applications of split/rejoin maneuvers in the field of robotics include reconnaissance, sampling, and surveillance. In contrast, rigid formations (*globally rigid* or *locally rigid*) are required in many engineering applications, for example, parallel and simultaneous transportation of vehicles or delivery of payloads [12, 14, 16]. Alterations or distortions in formation must be detected early by a system and allow for reconfiguration for continued operation [7] in a locally rigid formation.

In recent literature, a variety of control strategies have been proposed for formation or cooperative control of mobile robots and have been roughly categorized into virtual structure, behavior-based, and leader-follower schemes. The behavior-based approach normally [3] assigns different possible behaviors (e.g., formation keeping, obstacle avoidance, collision avoidance, and target attraction) to each individual robot. The ultimate action of each robot is determined by evaluating the comparative importance of each behavior. The limitation of the behavior-based approach is that it is difficult to analyze its behavioral performance mathematically and therefore it is difficult to guarantee system stability [17]. The virtual structures [6, 18–20] consider the entire formation as a single virtual rigid body. This networked structure can then be considered similar to a physical body and thus it is easy to maintain the formation of the whole group during maneuvers; that is, the virtual rigid body progresses as a whole in a given course with some given bearing. In the leader-follower approach [14, 17, 21–24], one robot in the multirobot formation, a virtual robot, is assigned the role of the leader to pursue some team objective, while the follower robots follow their designated leader. The follower robots then place themselves relative to the leader and maintain a predefined offset with a desired relative position [17]. In [25], Sharma et al. proposed a leader-follower scheme in a Lyapunov-based decentralized formation control planner for a swarm of 2-link mobile manipulators. The acceleration-based controllers ensured a locally rigid formation for the swarm. Locally rigid formation was achieved by integrating maximum and minimum interrobot distance bounds with desired headings in the control scheme.

In this paper, we adopt the architecture of the Lyapunov-based Control Scheme (LbCS) of [14], an artificial potential field (APF) method to design attractive and repulsive potential field functions to control the formation of C-ITS. An extension to [25], this paper presents a new set of nonlinear time-invariant control laws through an amalgamation of the LbCS and a new leader-follower scheme to maintain and mobilize a locally rigid formation. The limitation of the approach is twofold; firstly it comes with the disadvantages of APFs, in particular the local minima, and secondly the dependence on the lead robot and the poor disturbance rejection properties [26] of the leader-follower scheme. However, the novelty of the new approach lies in its ability to design continuous nonlinear control laws to translate locally rigid formations of nonholonomic systems tagged with dynamic constraints. The LbCS controllers are elegant and very simple to construct compared to the mathematically and computationally intensive ones in literature, and LbCS has an in-built process of converting these constraints into artificial obstacles and incorporating them into the controllers. Finally, the virtual connectivity from the new leader-follower scheme facilitates diverse tasks compared to merely cooperative agents [26].

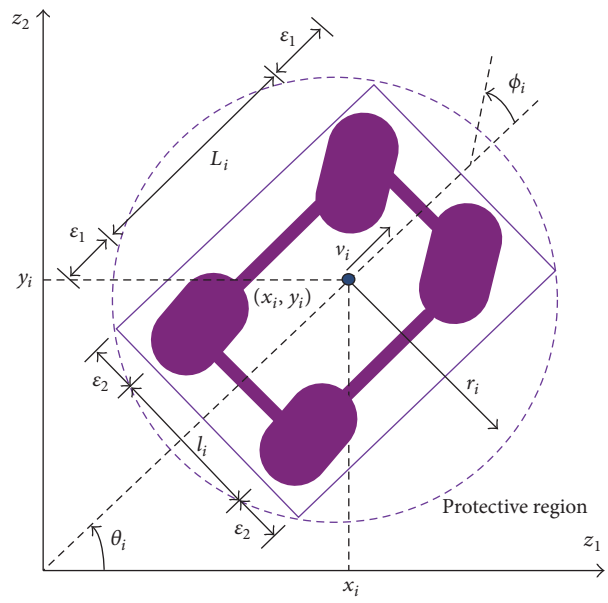


FIGURE 1: Kinematic model of the car-like mobile robot.

The advantage of the proposed scheme is that it uses Cartesian coordinate representation to avoid any singular points as encountered when using polar coordinates, even though polar coordinate representation may still be simpler to use. Another advantage of the proposed scheme in this paper is that it has a dual frame methodology, a new leader-follower approach, and the assignment of a single target for the group of vehicles. The overarching framework is a leader-follower scheme for C-ITS to establish, maintain, and translate the whole team in a locally rigid formation around the workplace performing a given task through centralized control laws. Finally, the treatment of several categories of obstacles is also included within the algorithm.

2. Vehicle Model

In this section, we derive a new kinematic model for the leader-follower based formation control of multiple vehicles. We will consider n ($n \in \mathbb{N}$) car-like vehicles in the Euclidean plane. We let $\mathcal{A}_1$ represent the leader and $\mathcal{A}_i$ for $i = 2, \ldots, n$ take the role of follower robots. With reference to Figure 1, adopted from [27], and for $i \in \{1, \ldots, n\}$, (x_i, y_i) represents the Cartesian coordinates and gives the reference point of each vehicle and θ_i gives the orientation of the ith car with respect to z_1-axis of the $z_1 z_2$-plane. Moreover, ϕ_i gives the ith vehicles steering angle with respect to its longitudinal axis. L_i represents the distance between the centers of the front and rear axles of the ith robot, and l_i is the length of each axle.

Next, to ensure that each vehicle safely steers past an obstacle, we adopt the nomenclature of [14] and construct circular regions that protect the robot. With reference to Figure 1, given the *clearance parameters* $\epsilon_1 > 0$ and $\epsilon_2 > 0$, we enclose each vehicle by a protective circular region centered at (x_i, y_i) with radius $r_i = (1/2)\sqrt{(L_i + 2\epsilon_1)^2 + (l_i + 2\epsilon_2)^2}$.

Furthermore, we assume no slippage condition of the rear and front wheels of the mobile robots of $\mathscr{A}_i$ when in contact with a rigid surface, that is, the lateral (or tangential) velocities of the wheels of the cars are assumed to be zero. We have no slippage (i.e., $\dot{x}_i \sin\theta_i - \dot{y}_i \cos\theta_i = 0$) and pure rolling (i.e., $\dot{x}_i \cos\theta_i + \dot{y}_i \sin\theta_i = v_i$) of the car-like mobile robot which generate the *nonholonomic constraints* on the system. These assumptions of no slippage and pure rolling of the car-like mobile robots are essential to generate the nonholonomic constraints on the system. The kinematics of the system which inherently capture these nonholonomic constraints, adopted from [14], are

$$\begin{aligned} \dot{x}_i &= v_i \cos\theta_i - \frac{L_i}{2}\omega_i \sin\theta_i, \\ \dot{y}_i &= v_i \sin\theta_i + \frac{L_i}{2}\omega_i \cos\theta_i, \\ \dot{\theta}_i &= \frac{v_i}{L_i}\tan\phi_i \coloneqq \omega_i, \\ \dot{v}_i &\coloneqq \sigma_{i1}, \\ \dot{\omega}_i &\coloneqq \sigma_{i2}, \end{aligned} \tag{1}$$

for $i \in \{1, \ldots, n\}$, and, without any loss of generality, we assume that $\phi_i = \theta_i$. In system (1), v_i and ω_i are, respectively, the instantaneous translational and rotational velocities, while σ_{i1} and σ_{i2} are the instantaneous translational and rotational accelerations of the ith robot.

Now, system (1) is a description of the instantaneous velocities and accelerations of $\mathscr{A}_i$. Let the vector $\mathbf{x}_i \coloneqq (x_i, y_i, \theta_i, v_i, \omega_i) \in \mathbb{R}^5$ refer to the position (x_i, y_i), orientation, θ_i, and the velocities (v_i, ω_i) of $\mathscr{A}_i$ at time $t \geq 0$.

Now, let

$$\begin{aligned} \mathbf{f}_i(\mathbf{x}_i) &= \left(f_{i1}(\mathbf{x}_i), f_{i2}(\mathbf{x}_i), f_{i3}(\mathbf{x}_i), 0, 0\right) \\ &\coloneqq \left(\dot{x}_i, \dot{y}_i, \dot{\theta}_i, 0, 0\right) \in \mathbb{R}^5, \end{aligned} \tag{2}$$

and $\mathbf{u}_i(t) \coloneqq (\sigma_{i1}(t), \sigma_{i2}(t)) \in \mathbb{R}^2$. Then system (1) can be written compactly as

$$\dot{\mathbf{x}}_i \coloneqq \mathbf{f}_i(\mathbf{x}_i) + \mathbf{B}_i \mathbf{u}_i(t), \tag{3}$$

where $\mathbf{B}_i$ is a 5×2 matrix of the form

$$\mathbf{B}_i = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{4}$$

Let $\mathbf{x} \coloneqq (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) \in \mathbb{R}^{5n}$ refer to the positions, orientations, and the velocities of all the vehicles in the C-ITS.

Let $\mathbf{f}(\mathbf{x}) \coloneqq (\mathbf{f}_1(\mathbf{x}), \mathbf{f}_2(\mathbf{x}), \ldots, \mathbf{f}_n(\mathbf{x})) \in \mathbb{R}^{5n}$ and $\mathbf{u}(t) \coloneqq (\mathbf{u}_1(t), \mathbf{u}_2(t), \ldots, \mathbf{u}_n(t)) \in \mathbb{R}^{2n}$. Then we have the following initial-value problem for $\mathscr{A}_i$:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{x}(t_0) &\coloneqq \mathbf{x}_0, \quad t_0 \geq 0, \end{aligned} \tag{5}$$

where if $\mathbf{0}$ is a 5×2 matrix of all zero entries,

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{B}_n \end{bmatrix}. \tag{6}$$

Now, assume that the final position of $\mathscr{A}_i$ is at the point $(x_i, y_i) = (p_{i1}, p_{i2})$ and final orientation at this point is $\theta_i = p_{i3}$. Its final instantaneous velocity vector is $(v_i, \omega_i) = (0, 0)$. Then it is clear that the points

$$\mathbf{x}_i^* \coloneqq (p_{i1}, p_{i2}, p_{i3}, 0, 0) \in \mathbb{R}^5 \tag{7}$$

are the components of the equilibrium point of system (5) in which we are interested; that is,

$$\mathbf{x}_e \coloneqq (\mathbf{x}_1^*, \mathbf{x}_2^*, \ldots, \mathbf{x}_n^*) \in \mathbb{R}^{5n}. \tag{8}$$

2.1. Leader-Follower Based Formation Scheme. Next we define two reference frames: the body frame which is fixed with the rotating body of the leader, $\mathscr{A}_1$, and a space frame, the inertial frame similar to one proposed in [16].

We assign a Cartesian coordinate system (X-Y) fixed on the leader body, as shown in Figure 2 adopted from [27], based on the concept of an instantaneous corotating frame of reference. Thus, when the leader $\mathscr{A}_1$ rotates, we have a rotation of the X-Y-axes.

To define the corotating frame of reference, first an origin is selected on the leader robot at (x_1, y_1). An axis of rotation is then set up, which is perpendicular to the plane of motion of the leader. Thus, at any selected moment t, the chosen rotating frame of reference rotates at an angular rate equal to the rate of rotation of the leader $\mathscr{A}_1$ about (x_1, y_1). Let r_{1k} represent the straight-line distance between the reference point of the leader and the kth follower, and α_{1k} represents the angle measured between the straight line joining the reference points of the leader and the kth follower and X-axis. Thus, given the leader's position and orientation, as long as (r_{1k}, α_{1k}), as shown in Figure 2, is fixed, the kth follower robot's position will be unique. We define the shape of the formation of the mobile robots as $\zeta = [\zeta_{12}, \zeta_{13}, \ldots, \zeta_{1n}]^T$, where $\zeta_{1k} = [r_{1k}, \alpha_{1k}]^T$ for $k \in \{2, \ldots, n\}$.

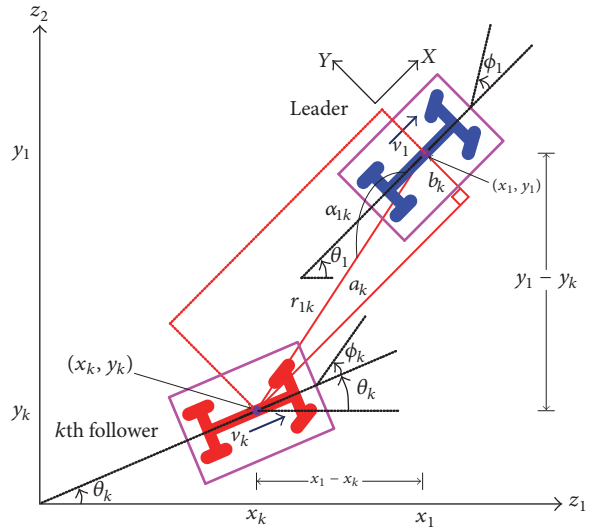


FIGURE 2: The proposed scheme utilizing a rotation of axes with the body frame fixed at the leader robot.

Definition 1. Let $2r_k < r_{1k}^d < 2r_k + \varsigma$, where, for $\varsigma > 0$ and $k \in \{2, \ldots, n\}$, the group of mobile robots make a $\zeta^d = [\zeta_{12}^d, \zeta_{13}^d, \ldots, \zeta_{1n}^d]^T$ formation, if $\exists T > 0, \forall t > T$:

$$\begin{aligned} r_{1k} &= r_{1k}^d, \\ \alpha_{1k} &= \alpha_{1k}^d. \end{aligned} \tag{9}$$

This gives then the polar coordinate representation of the follower's position relative to that of the leader. However, such representations using polar coordinates lead to certain singularities in the controllers [16]. To avoid such singular points, we consider the position of the kth follower by considering the relative distances of the kth follower from the leader $\mathscr{A}_1$ along the given X and Y directions. Thus, we have

$$\begin{aligned} A_k &= -(x_1 - x_k)\cos\theta_1 - (y_1 - y_k)\sin\theta_1, \\ B_k &= (x_1 - x_k)\sin\theta_1 - (y_1 - y_k)\cos\theta_1, \end{aligned} \tag{10}$$

for $k \in \{2, \ldots, n\}$, and A_k and B_k are the kth follower relative positions with respect to the X-Y coordinate system. If A_k and B_k are known and fixed, the follower's position will be distinct. Thus, to obtain a desired formation, one needs to know the design parameters a_k and b_k and the desired relative positions along the X-Y directions, such that the control objective would be to achieve $A_k \to a_k$ and $B_k \to b_k$. That is, $r_{1k} \to r_{1k}^d$ and $\alpha_{1k} \to \alpha_{1k}^d$, where $r_{1k}^d = \sqrt{a_k^2 + b_k^2}$ and $\alpha_{1k}^d = \tan(a_k/b_k)$, hence maintaining local rigidity of any formation.

Definition 2. We define a locally rigid formation as one which is maintained but allows for slight changes or distortions in the distances and angles between any two vehicles temporarily, while enroute to the target and task completion.

Remark 3 (formation initialization). Prescribed formations can be initialized by assigning appropriately the design parameters a_k and b_k. This will initiate the vehicles from their arbitrary initial positions to move into a desired formation pattern.

2.2. Formation Control Objective. Design acceleration-based control laws $(\sigma_{i1}, \sigma_{i2})$ which are functions of $\mathbf{x}_i := (x_i, y_i, \theta_i, v_i, \omega_i) \in \mathbb{R}^5$ for $i \in \{1, \ldots, n\}$, for each vehicle such that the C-ITS in a prescribed formation moves towards a predefined target of the leader in an obstacle cluttered environment. In addition, we also want the C-ITS to ensure obstacle and collision avoidance and maintain a prescribed locally rigid formation.

3. Artificial Potential Field Function

This section formulates collision-free trajectories of the vehicle system under kinodynamic constraints. We want to design the acceleration controllers, σ_{i1} and σ_{i2}, so that the team of vehicles moves safely towards the target of the leader while maintaining a prescribed locally rigid formation. We will design the attractive potential field functions which establish and translate formation and the repulsive potential field functions which ensure collision and obstacle avoidances.

In the following subsections, we will design these attractive and repulsive potential field functions which are subsequently summed to form the total potentials or artificial potential field of the system.

3.1. Attractive Potential Field Functions

3.1.1. Attraction to Target. A target is assigned to the leader. When the leader moves towards its defined target, the follower vehicles move with the leader, maintaining the formation. We want the leader $\mathscr{A}_1$ to start from an initial position, move towards a target, and finally converge at the center of the target.

Definition 4. The stationary target for the leader robot $\mathscr{A}_1$ is a circular disk with center (p_{11}, p_{12}) and radius rt.

The leader $\mathscr{A}_1$ will move towards its defined target, while the follower vehicles move with their leader maintaining a desired relative position, hence a locally rigid formation. For the attraction of $\mathscr{A}_1$ to its target, we consider an attractive potential function:

$$V_1(\mathbf{x}) = \frac{1}{2}\left[(x_1 - p_{11})^2 + (y_1 - p_{12})^2 + v_1^2 + \omega_1^2\right]. \tag{11}$$

The above function is a measure of its convergence to the target with the inclusion of the velocity components [14]. For the follower vehicle $\mathscr{A}_i$ for $i = 2, \ldots, n$ to maintain its desired relative position with respect to the leader, $\mathscr{A}_1$, we utilize the following potential function:

$$V_i(\mathbf{x}) = \frac{1}{2}\left[(A_i - a_i)^2 + (B_i - b_i)^2 + v_i^2 + \omega_i^2\right], \tag{12}$$

for $i = 2, \ldots, n$, where a_i and b_i will have unique values as dictated by the shape of the desired formation.

3.1.2. Auxiliary Function. To guarantee the convergence of the vehicles to their designated targets, we design an auxiliary function as

$$G_1(\mathbf{x}) = \frac{1}{2}\left[(x_1 - p_{11})^2 + (y_1 - p_{12})^2 + \rho_1(\theta_1 - p_{13})^2\right], \quad (13)$$

where p_{13} is the prescribed final orientation of the leader robot, $\mathscr{A}_1$, and

$$G_i(\mathbf{x}) = \frac{1}{2}\left[(A_i - a_i)^2 + (B_i - b_i)^2 + \rho_i(\theta_i - \theta_1)^2\right], \quad (14)$$

for $i = 2, \ldots, n$. The function ensures that the controllers become zero at the leader's target. The constant ρ_i is a binary constant denoted in (13) and (14) as the *angle-gain parameter* for θ_i, $i = 1, \ldots, n$. An *angle-gain parameter* will take a value of one only if a final predefined orientation is warranted; else it takes the default value of zero. This auxiliary function is then multiplied to the repulsive potential field functions.

3.2. Repulsive Potential Field Functions. We desire each vehicle to avoid all fixed and moving obstacles intersecting its path. Hence, we construct appropriate obstacle avoidance functions that measure the Euclidean distances between mobile robots and the obstacles on our roads. To obtain the desired avoidance, we generate repulsive potential field around the obstacles by designing a repulsive potential field function for each obstacle in accordance with LbCS. The repulsive potential fields function is an inverse function that encodes an avoidance function to the denominator and a control parameter in the numerator [14].

3.2.1. Fixed Obstacles. Let us fix $n \in \mathbb{N}$ solid obstacles on the roads, such as potholes, distance marker posts, or road hazard delineators. We assume that the lth obstacle is a circular disk with center (o_{l1}, o_{l2}) and radius ro_l. For the ith vehicle to avoid the lth obstacle, we consider an avoidance function:

$$\mathrm{FO}_{il}(\mathbf{x}) = \frac{1}{2}\left[(x_i - o_{l1})^2 + (y_i - o_{l2})^2 - (ro_l + r_i)^2\right], \quad (15)$$

where $i = 1, \ldots, n$ and $l = 1, \ldots, q$.

Consider, for example, the presence of three obstacles (i.e., $q = 3$) within the workspace, with $0 < z_1 < 70$ and $0 < z_2 < 70$. The total potentials, where $\alpha_{1l} > 0$ is a control or tuning parameter, which govern the motion of the leader $\mathscr{A}_1$ are

$$V_1(\mathbf{x}) + \sum_{l=1}^{3} \frac{\alpha_{1l}}{\mathrm{FO}_{1l}(\mathbf{x})}. \quad (16)$$

Figure 3 presents a three-dimensional view of the total potentials and Figure 4 presents the corresponding contour plot produced by (16).

The total potentials are generated for target attraction and avoidance of three stationary disk-shaped obstacles. For better visualization, the target of the leader is located at $(p_1, p_2) = (35, 35)$, and the disks are fixed at

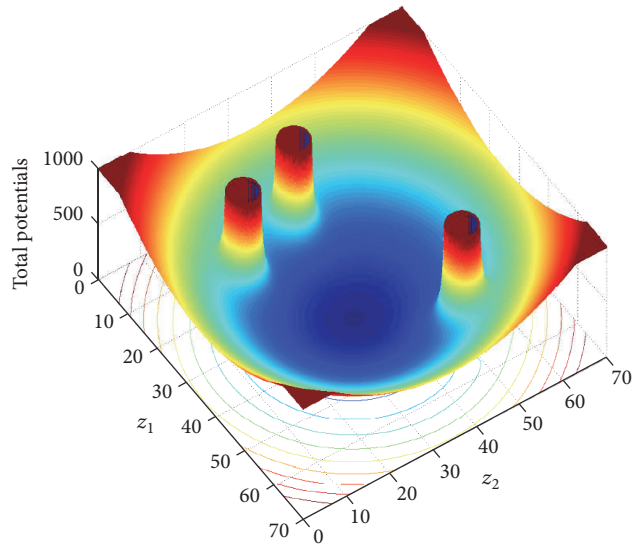


Figure 3: A three-dimensional view of the total potentials.

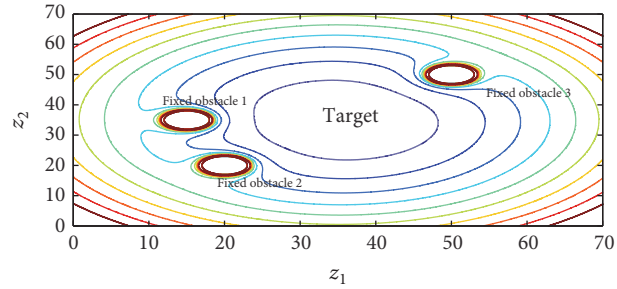


Figure 4: The contour plot of the total potential.

$(o_{11}, o_{12}) = (15, 35)$, $(o_{21}, o_{22}) = (20, 20)$, and $(o_{31}, o_{32}) = (50, 50)$, with radii of $ro_l = 3$, while $\alpha_{l1} = 1000$, for $l = 1, \ldots, 3$. Also, the velocity and angular components of the lead vehicle have been treated as constants ($v_1 = 0.5$, $\omega_1 = 0$, and $\theta_1 = 0$).

3.3. Boulevard Limitations

Definition 5. Consider a section of the road defined, for some $\eta_1 > 2r_i$, for $i = 1, \ldots, n$, as

$$WS = \left\{(z_1, z_2) \in \mathbb{R}^2 : 0 \le z_2 \le \eta_1\right\}. \quad (17)$$

The boundaries of the road section, illustrated in Figure 5, are defined as follows:

(a) Lower boundary: $B_1 = \{(z_1, z_2) \in \mathbb{R}^2 : z_2 = 0\}$

(b) Upper boundary: $B_2 = \{(z_1, z_2) \in \mathbb{R}^2 : z_2 = \eta_1\}$

We require the prescribed formation to stay within the boundaries of the road at all time $t \ge 0$. In our LbCS, these boundaries are considered as *fixed obstacles*. For the ith robot

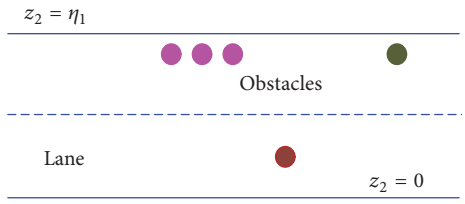


FIGURE 5: A schematic representation of the road with obstacles.

to avoid these, we define the following potential functions for the upper and lower boundaries, respectively:

$$W_{i1}(\mathbf{x}) = y_i - r_i, \tag{18a}$$

$$W_{i2}(\mathbf{x}) = \eta_1 - (y_i + r_i). \tag{18b}$$

Now, since $\eta_1 > 2r_i$, each of the functions is positive in WS, for $i = 1, \ldots, n$. As discussed before, these obstacle avoidance functions will be combined with appropriate tuning parameters to generate repulsive potential field functions for the boundaries of the workspace.

3.4. Final Orientation of Formation. Although the final position is reachable, it is virtually impossible to simultaneously harvest exact orientations via continuous feedback controllers of nonholonomic systems [14], a direct result of Brockett's theorem [28]. We utilize the concepts of ghost wall and *minimum distance technique* (MDT) from [14] to force final orientations of the vehicles. This is needed to confine the vehicles to within the lane markings and later for parking a vehicle into a parking bay. To begin, we construct ghost walls along the three sides of the final positions of the vehicles, treated as a parking bay, with the orientation depending on the desired final orientation of the vehicles. To avoid the ghost walls of each vehicle's parking bay, we utilize MDT, which gives the perpendicular distance between the closest point on each kth ghost wall and (x_i, y_i). The avoidance of these closet points on a given ghost wall at any time $t \geq 0$ essentially results in the avoidance of the entire wall by a vehicle.

Now let us consider the kth ghost wall in the z_1z_2-plane from the point (a_{k1}, b_{k1}) to the point (a_{k2}, b_{k2}). We assume that the point (x_i, y_i) is closest to it at the tangent line which passes through the point. From geometry, it is known that if (Lx_{ik}, Ly_{ik}) is the point of intersection of this tangent, then

$$\begin{aligned} Lx_{ik} &= a_{k1} + \lambda_{ik}(a_{k2} - a_{k1}), \\ Ly_{ik} &= b_{k1} + \lambda_{ik}(b_{k2} - b_{k1}), \end{aligned} \tag{19}$$

where

$$\begin{aligned} \lambda_{ik} &= (x_i - a_{k1})\, d_k + (y_i - b_{k1})\, r_k, \\ d_k &= \frac{a_{k2} - a_{k1}}{(a_{k2} - a_{k1})^2 + (b_{k2} - b_{k1})^2}, \\ r_k &= \frac{b_{k2} - b_{k1}}{(a_{k2} - a_{k1})^2 + (b_{k2} - b_{k1})^2}. \end{aligned} \tag{20}$$

If $\lambda_{ik} \geq 1$, then we let $\lambda_{ik} = 1$; if $\lambda_{ik} \leq 0$, then we let $\lambda_{ik} = 0$; otherwise we accept the value of λ_{ik} between 0 and 1, in which case there is a perpendicular line to the point (Lx_{ik}, Ly_{ik}) on the ghost wall from the center (x_i, y_i) of the ith vehicle at every time $t \geq 0$.

The leader will be avoiding the 1st, 2nd, and 3rd ghost walls, while the ith follower will be avoiding the $(3i-2)$, $(3i-1)$, and $(3i)$ ghost walls for $i = 2, \ldots, n$. Now, for the ith vehicle to avoid the closest point of each of the kth line segments, we consider an avoidance function

$$LS_{ik}(\mathbf{x}) = \frac{1}{2}\left[(x_i - Lx_{ik})^2 + (y_i - Ly_{ik})^2 - r_i^2\right], \tag{21}$$

for $k \in \{3i-2, 3i-1, 3i\}$ and $i = 1, \ldots, n$.

3.5. Moving Obstacles. To generate feasible trajectories, we consider moving obstacles in the workspace, in which the C-ITS has prior knowledge. Here, each vehicle has to be treated as a moving obstacle for all other vehicles on the road. The vehicles will have to travel towards their targets while avoiding another vehicle in their path. For the vehicle $\mathcal{A}_i$ to avoid the vehicle $\mathcal{A}_j$, via vehicle-to-vehicle (V-V) communication, we adopt an avoidance function:

$$\begin{aligned} \mathrm{MO}_{ij}(\mathbf{x}) &= \frac{1}{2}\left[(x_i - x_j)^2 + (y_i - y_j)^2 - (r_i)^2\right], \\ &\qquad \text{for } i, j = 1, \ldots, n \text{ with } i \neq j. \end{aligned} \tag{22}$$

3.5.1. Dynamic Constraints. Practically, the steering angle of the ith autonomous vehicle is limited due to mechanical singularities, while the translational speed is restricted due to safety reasons [14]. Subsequently, we have the following;

(i) $|v_i| < v_{\max}$, where $v_{\max}$ is the *maximal achievable speed* of the ith vehicle.

(ii) $|\omega_i| < v_{\max}/|\rho_{\min}|$, where $\rho_{\min} := L_i/\tan(\phi_{\max})$. This condition arises due to the boundedness of the steering angle ϕ_i. That is, $|\phi_i| \leq \phi_{\max} < \pi/2$, where $\phi_{\max}$ is the *maximal steering angle*.

Remark 6. For simplicity, the values of $v_{\max}$ and $\phi_{\max}$ will be kept the same for each vehicle.

As per the LbCS, for each dynamic constraint, we design a corresponding *artificial obstacle*. For example, we consider the artificial obstacle $\mathrm{AO}_i = \{v_i \in \mathbb{R} : v_i \leq -v_{\max} \text{ or } v_i \geq v_{\max}\}$ for the constraint tagged to v_i. We can create similar artificial obstacles for the other limitations as well. Hence, we consider the following avoidance functions:

$$\begin{aligned} U_{i1}(\mathbf{x}) &= \frac{1}{2}(v_{\max} - v_i)(v_{\max} + v_i), \\ U_{i2}(\mathbf{x}) &= \frac{1}{2}\left(\frac{v_{\max}}{|\rho_{\min}|} - \omega_i\right)\left(\frac{v_{\max}}{|\rho_{\min}|} + \omega_i\right), \\ &\qquad \text{for } i = 1, \ldots, n. \end{aligned} \tag{23}$$

4. Design of the Acceleration-Based Controllers

4.1. Lyapunov Function. We now construct the total potentials, that is, a Lyapunov function for system (1). First, for $i = 1, \ldots, n$, we introduce the following *control parameters* that we will use in the repulsive potential functions:

(i) $\alpha_{il} > 0, l = 1, \ldots, q$, for the collision avoidance of q disk-shaped obstacles.

(ii) $\beta_{is} > 0, s = 1, 2$, for the avoidance of the artificial obstacles from dynamic constraints.

(iii) $\eta_{ij} > 0, j = 1, \ldots, n, i \neq j$, for the collision avoidance between any two vehicles.

(iv) $\kappa_{ip} > 0, p = 1, \ldots, 2$, for the avoidance of the lane boundaries.

(v) $\gamma_{ik} > 0, k = 1, \ldots, 3n$, for the avoidance of the parking bays.

The above parameters are determined via heuristics, but there is also an opportunity to obtain optimal values through optimization techniques. Using these, we now construct the following Lyapunov function for system (1) with two components, namely, the attractive and repulsive potential field functions:

$$L_{(1)}(\mathbf{x}) = \sum_{i=1}^{n} \left[V_i(\mathbf{x}) + G_i(\mathbf{x}) \cdot \left(\sum_{l=1}^{q} \frac{\alpha_{il}}{\mathrm{FO}_{il}(\mathbf{x})} + \sum_{s=1}^{2} \frac{\beta_{is}}{U_{is}(\mathbf{x})} + \sum_{p=1}^{2} \frac{\kappa_{ip}}{W_{ip}(\mathbf{x})} \right) \right] + \sum_{i=1}^{n} G_i(\mathbf{x}) \left(\sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{\eta_{ij}}{\mathrm{MO}_{ij}(\mathbf{x})} + \sum_{k=1}^{3n} \frac{\gamma_{ik}}{LS_{ik}(\mathbf{x})} \right). \tag{24}$$

4.2. Nonlinear Acceleration Controllers. The process of designing the feedback controllers begins by noting that the functions f_{ik} to g_{ij} for $i = 1, \ldots, n$, $j = 1, 2$, and $k = 1, \ldots, 3$ are defined as in the Appendix (on suppressing $\mathbf{x}$).

Remark 7. The choice of total potential $L(\mathbf{x})$ given in (24) with its terms specified in (11)–(16), (18a), (18b), and (21)–(23) has the following properties:

(i) It attains a minimum value when the robots are at their desired locations.

(ii) It goes to infinity whenever one or more robots come in contact with an obstacle.

Remark 8. With the interrobot bounds (see (12) and (22)) in place, it is guaranteed that the robots reestablish the predetermined formation if the robot positions are slightly distorted with the encounter of obstacle(s) soon after the avoidance and before reaching the target.

So, we design the following theorem.

Theorem 9. *Consider n car-like mobile robots in the C-ITS, whose motion is governed by the ODEs described in system (1). The principal goal is to establish and control a prescribed formation, facilitate maneuvers of the vehicles within a constrained environment, and reach the target configuration while maintaining a desired formation. The subtasks include restrictions placed on the workspace, convergence to predefined targets, and consideration of kinodynamic constraints. Utilizing the attractive and repulsive potential field functions, the following continuous time-invariant acceleration control laws can be generated in accordance with the LbCS of system (1):*

$$\sigma_{i1} = -\frac{\left[\delta_{i1} v_i + f_{i1} \cos\theta_i + f_{i2} \sin\theta_i\right]}{g_{i1}},$$
$$\sigma_{i2} = -\frac{\left[\delta_{i2} \omega_i + (L_i/2)\left(f_{i2} \cos\theta_i - f_{i1} \sin\theta_i\right) + f_{i3}\right]}{g_{i2}}, \tag{25}$$

for $i = 1, \ldots, n$, where $\delta_{i1} > 0$ and $\delta_{i2} > 0$ are constants commonly known as convergence parameters.

5. Stability Analysis

We utilize Lyapunov's direct method to provide a mathematical proof of stability of system (1).

Theorem 10. *Let (p_{11}, p_{12}) be the position of the target of the leader and let p_{i3}, for $i = 1, \ldots, n$, be the prescribed final orientations of the robots. Let p_{i1} and p_{i2} satisfy*

$$a_i = -(p_{11} - p_{i1}) \cos\theta_1 - (p_{12} - p_{i2}) \sin\theta_1,$$
$$b_i = (p_{11} - p_{i1}) \sin\theta_1 - (p_{12} - p_{i2}) \cos\theta_1, \tag{26}$$

for any given a_i and b_i, for $i = 2, \ldots, n$. If $\mathbf{x}_e \in \mathbb{R}^{5n}$ as defined in (8) is an equilibrium point for (1), then $\mathbf{x}_e \in D(L_{(1)}(\mathbf{x}))$ is a stable equilibrium point of system (1).

Proof. One can easily verify the following, for $i \in \{1, \ldots, n\}$:

(1) $L_{(1)}(\mathbf{x})$ is defined as continuous and positive over the domain $D(L_{(1)}(\mathbf{x})) = \{\mathbf{x} \in \mathbb{R}^{5n} : \mathrm{FO}_{il}(\mathbf{x}) > 0,\ l = 1, \ldots, q;\ \mathrm{MO}_{ij}(\mathbf{x}) > 0,\ j = 1, \ldots, n,\ j \neq i;\ W_{ip}(\mathbf{x}) > 0,\ p = 1, \ldots, 4;\ LS_{ik}(\mathbf{x}) > 0,\ k = 1, \ldots, 3n;\ U_{is}(\mathbf{x}) > 0,\ s = 1, 2\}$.

(2) $L_{(1)}(\mathbf{x}_e) = 0$; $\dot{L}_{(1)}(\mathbf{x}_e) = 0$.

(3) $L_{(1)}(\mathbf{x}) > 0\ \forall \mathbf{x} \in D(L_{(1)}(\mathbf{x}))/\mathbf{x}_e$.

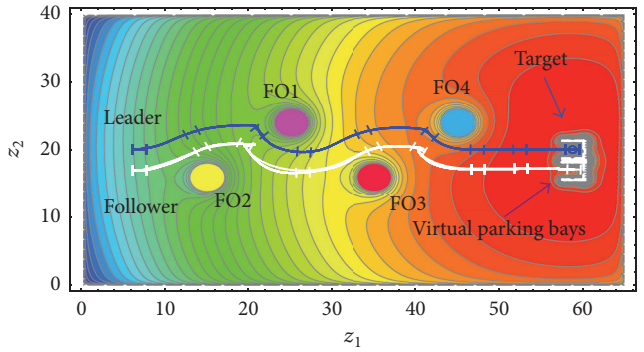
(4) $\dot{L}_{(1)}(\mathbf{x}) = -\sum_{i=1}^{n} (\delta_{i1} v_i^2 + \delta_{i2} \omega_i^2) \leq 0, \forall \mathbf{x} \in D(L_{(1)}(\mathbf{x}))$.

(5) $L_{(1)}(\mathbf{x}) \in C^1(D(L_{(1)}(\mathbf{x})))$.

Hence, $L_{(1)}(\mathbf{x})$ is classified as a Lyapunov function for system (1) and $\mathbf{x}_e$ is a stable equilibrium point in the sense of Lyapunov. □

6. Simulation Results

In this section, we illustrate the effectiveness of the proposed continuous controllers by simulating a number of virtual scenarios for the C-ITS on our roads.



FO: Fixed obstacle

Figure 6: The evolution of C-ITS trajectories and the contour plot in the presence of obstacles in Scenario 1.

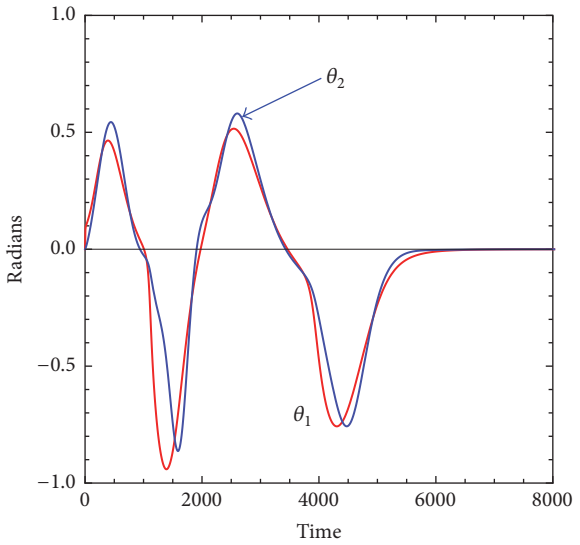


Figure 7: Orientations of $\mathscr{A}_1$ and $\mathscr{A}_2$ for Scenario 1.

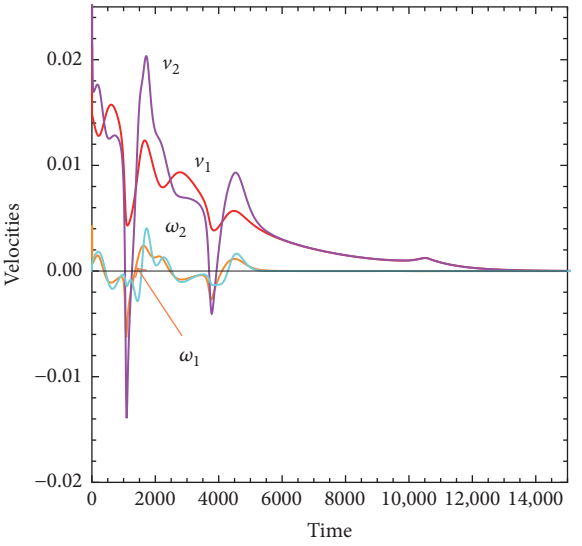


Figure 8: Translational and rotational velocities for Scenario 1.

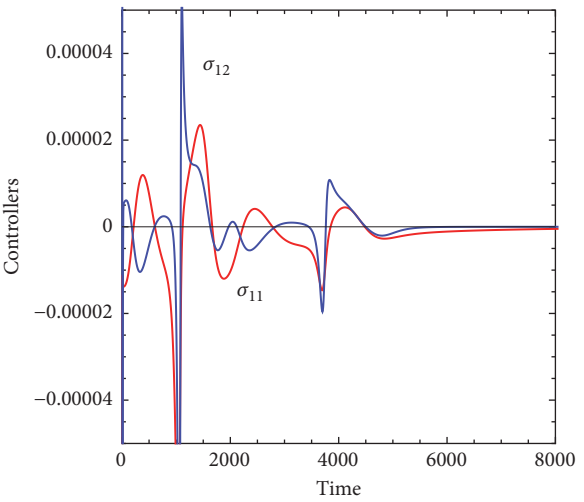


Figure 9: Accelerations σ_{11} and σ_{12} of $\mathscr{A}_1$ for Scenario 1.

6.1. Scenario 1: Line Formation in the Presence of Obstacles. We consider the motion of two automated vehicles in a *line formation* in the presence of obstacles. The follower vehicle is prescribed a position relative to the leader $\mathscr{A}_1$ as seen in Figure 6. While the leader moves towards its target, the follower maintains a desired distance and orientation relative to the leader, therefore maintaining a locally rigid formation throughout the journey.

Assuming that the appropriate units have been accounted for, Table 1 provides the corresponding initial and final configurations of the two vehicles and other parameters required to simulate Scenario 1.

Figure 6 also depicts the contour plot of the potential fields of the system enroute to the final destination. Figures 7 and 8 show the orientations and velocities of $\mathscr{A}_1$ and $\mathscr{A}_2$, respectively. Figure 9 depicts the time evolution of the nonlinear controllers of the leader $\mathscr{A}_1$, and Figure 10 compares the relative distance r_{12} to the desired relative distance r_{12}^d. It is evident that the formation is slightly distorted when the pair of cooperative cars encounter an obstacle; however, the desired formation is reestablished before the pair reach the final destination.

6.2. Scenario 2: Effect of Noise on Line Formation. To evaluate the robustness of the proposed scheme, we look at the effect of noise on the formation of the C-ITS. It is sufficient to include the noise parameters in the components A_k and B_k which define the follower vehicles' relative position to the leader vehicle, with respect to the X-Y coordinate system, similar to the one proposed in [29]. Thus we have

$$\begin{aligned} A_k &= -(x_1 - x_k)\cos\theta_1 - (y_1 - y_k)\sin\theta_1 + \xi\gamma_k(t), \\ B_k &= (x_1 - x_k)\sin\theta_1 - (y_1 - y_k)\cos\theta_1 + \xi\nu_k(t). \end{aligned} \tag{27}$$

The terms $\xi\gamma_k(t)$ and $\xi\nu_k(t)$ are the small disturbances, where $\xi \in [0, 1]$ is the noise level, while $\gamma_k(t)$ and $\nu_k(t)$ are randomized time-dependent variables such that $\gamma_k(t) \in [-1, 1]$

Table 1: Numerical values of initial and final states, constraints, and parameters for Scenario 1.

	Initial configuration
Rectangular positions	$(x_1, y_1) = (7, 20)$ and $(x_2, y_2) = (7, 17)$
Velocities	$v_i = 0.5$ and $\omega_i = 0$, for $i = 1, 2$
Angular positions	$\theta_i = 0$, for $i = 1, 2$
Parking bays	$(a_{11}, b_{11}) = (57.5, 21.3)$ and $(a_{12}, b_{12}) = (60.1, 21.3)$ $(a_{31}, b_{31}) = (57.5, 18.7)$ and $(a_{32}, b_{32}) = (60.1, 18.7)$ $(a_{41}, b_{41}) = (57.5, 18.2)$ and $(a_{42}, b_{42}) = (60.1, 18.2)$ $(a_{61}, b_{61}) = (57.5, 15.6)$ and $(a_{62}, b_{62}) = (60.1, 15.6)$
	Constraints and parameters
Dimensions of robots	$L_i = 1.6$ and $l_i = 1.2$ for $i = 1, 2$
Leader's target	$(p_{11}, p_{12}) = (57, 20)$; $rt = 0.5$
Final orientations	$p_{13} = p_{23} = 0$
Max. translational velocity	$v_{\max} = 5$
Max. steering angle	$\phi_{\max} = \frac{\pi}{2}$
Clearance parameters	$\epsilon_1 = 0.1$ and $\epsilon_2 = 0.05$
Fixed obstacles	$(o_{11}, o_{12}) = (15, 16)$, $(o_{21}, o_{22}) = (25, 24)$, and $(o_{31}, o_{32}) = (35, 16)$ $(o_{41}, o_{42}) = (45, 24)$ and $ro_l = 2$ for $l = 2, \ldots, 4$
	Control and convergence parameters
Collision avoidance	$\eta_{ij} = 0.001$, for $i, j = 1, 2$, $j \neq i$ $\kappa_{ip} = 0.1$, for $i = 1, 2$, $p = 1, \ldots, 2$ $\gamma_{ik} = 0.1$, for $i = 1, 2$, $k = 1, \ldots, 3n$ $\alpha_{11} = 1$, $\alpha_{12} = \alpha_{14} = 0.1$, and $\alpha_{13} = 2$ $\alpha_{2l} = 0.01$ for $l = 1, \ldots, 4$
Dynamics constraints	$\beta_{is} = 0.01$, for $i, s = 1, 2$
Convergence	$\delta_{11} = 3000$, $\delta_{12} = 100$, $\delta_{21} = 10$, and $\delta_{22} = 100$

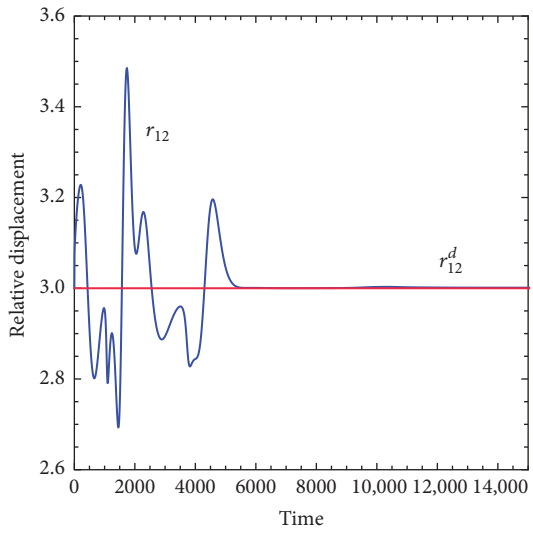


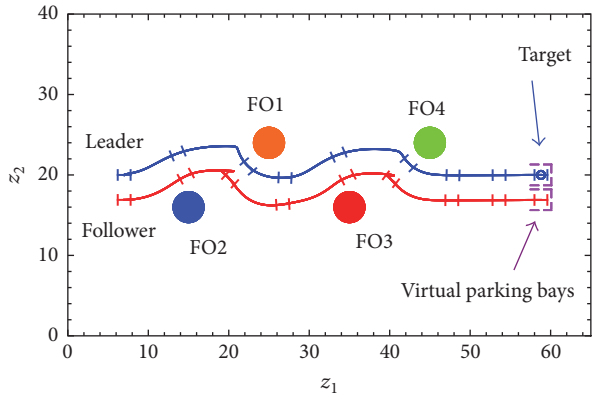
Figure 10: The relative distance of the follower to the leader vehicle compared to the desired relative distance in Scenario 1.

and $v_k(t) \in [-1, 1]$. Figure 11 shows the trajectories and the control signals under the influence of small disturbances, $\xi \in [0, 1]$. It is observed that the pair of cooperative cars maintain the line formation throughout their road journey even under the influence of the noise terms. There are slight distortions in their formation when the pair of cooperative vehicles encounter obstacles but these distortions are temporary. Also, the disturbances in the controllers at $t = 11{,}000$ units are a result of the parking bays. The two cars align themselves to achieve the desired orientations inside the parking bays. The final orientations are forced as a result of the repulsive potentials created by the walls of the parking bays.

6.3. Scenario 3: Diamond Formation. In this scenario, we have considered the leader vehicle at the center of a Diamond Formation with the followers positioned at each vertex (see Figure 12). The figure shows the formation maneuvered from an initial state to a predefined final state, with collision and obstacle avoidance.

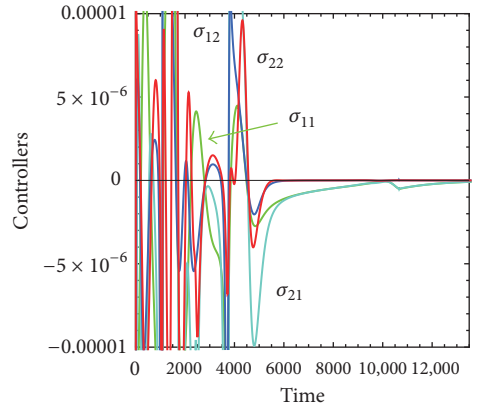
Figure 12 also depicts the contour plot of the potential fields and the corresponding collision-free path over the defined workspace, $0 < z_1 < 40$, containing the obstacles on the road. Figures 13 and 14 depict the time evolution of the nonlinear controllers of the leader and its follower vehicles. Assuming that the appropriate units have been accounted for, Table 2 (if different from Table 1) provides the corresponding initial and final configurations of the 5-car C-ITS and other parameters required to simulate Scenario 3. The coordinates of the parking bays can be obtained from Figure 12.

Clearly the translational and rotational accelerations of the vehicles decrease as the formation approaches a fixed obstacle and increase once it is able to evade it. Moreover, Figures 15 and 16 compare the relative distance A_k to the

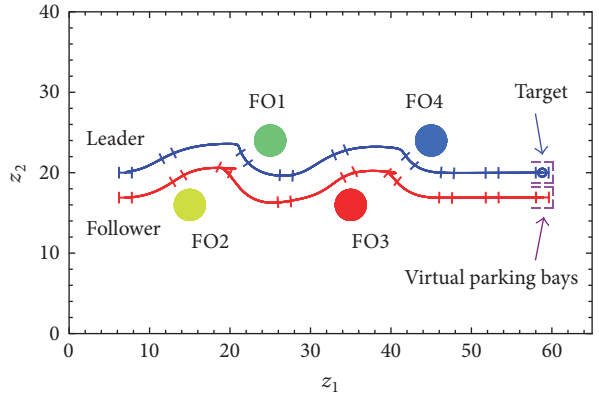


FO: Fixed obstacle

(a) Trajectories for $\xi = 0.1$

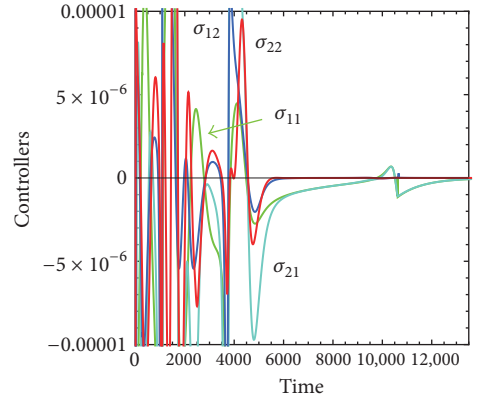


(b) Control signals, σ_{i1} and σ_{i2}, for $\xi = 0.1$

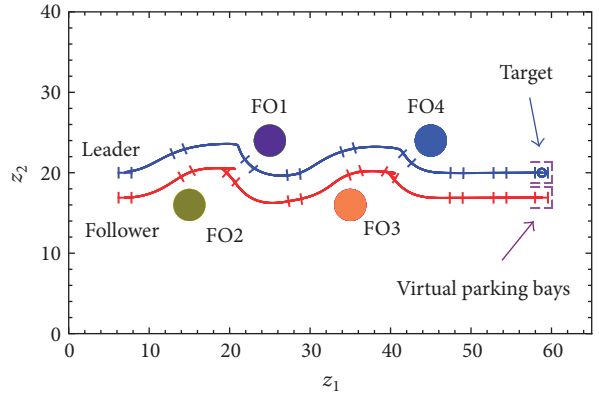


FO: Fixed obstacle

(c) Trajectories for $\xi = 0.2$

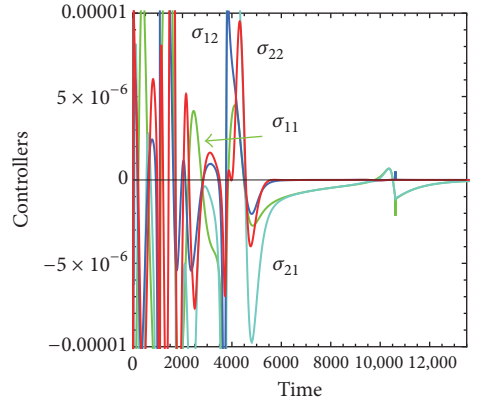


(d) Control signals, σ_{i1} and σ_{i2}, for $\xi = 0.2$



FO: Fixed obstacle

(e) Trajectories for $\xi = 0.5$



(f) Control signals, σ_{i1} and σ_{i2}, for $\xi = 0.5$

FIGURE 11: Trajectories of $\mathscr{A}_1$ and $\mathscr{A}_2$ and the evolution of control signals under various influences of noise.

desired relative distance a_k and compares B_k to the desired relative distance b_k, respectively. It is evident from the behavior shown in the figures that the formation gets slightly distorted when the C-ITS encounters an obstacle; however, the desired formation is reestablished before the leader $\mathscr{A}_1$ reaches its designated target, thus ensuring a locally rigid formation along the trajectory of the C-ITS.

7. Conclusion

This paper presents a set of nonlinear control laws using the LbCS to achieve a novel technique of extracting a locally rigid formation of a C-ITS made up of multiple car-like autonomous robots. A leader-following strategy is proposed to operate within the control scheme for the maintenance

Table 2: Numerical values of initial and final states, constraints, and parameters for Scenario 3.

	Initial configuration
Rectangular positions	$(x_1, y_1) = (7, 20)$, $(x_2, y_2) = (4, 20)$, $(x_3, y_3) = (7, 23)$ $(x_4, y_4) = (10, 20)$, and $(x_5, y_5) = (7, 17)$ $(a_2, b_2) = (3, 0)$, $(a_3, b_3) = (0, -3)$ $(a_4, b_4) = (-3, 0)$, and $(a_5, b_5) = (0, 3)$
Velocity and angular position	$v_i = 0.5$, $\omega_i = 0$, and $\theta_i = 0$ for $i = 1, \ldots, 5$
	Constraints and parameters
Leader's target	$(p_{11}, p_{12}) = (57, 37)$; $rt = 0.5$
Final orientations	$p_{i3} = 0$ for $i = 1, \ldots, 5$
	Control and convergence parameters
Obstacle avoidance	$\alpha_{il} = 10$, for $i = 1, \ldots, 5$, $l = 1, \ldots, 3$ $\kappa_{ip} = 0.1$, for $i = 1, \ldots, 5$, $p = 1, \ldots, 2$ $\gamma_{ik} = 1.2 \times 10^{-4}$, for $i = 1, \ldots, 5$, $k = 1, \ldots, 3i$, $k = 1, \ldots, 3n$
Collision avoidance	$\eta_{ij} = 0.01$, for $i, j = 1, \ldots, 5$, $j \neq i$
Dynamics constraints	$\beta_{is} = 0.001$, for $i = 1, \ldots, 5$, $s = 1, 2$
Convergence	δ_{11} =20,500, δ_{12} =10,500, $\delta_{i1} = 60$, and $\delta_{i2} = 60$ for $i = 1, \ldots, 5$

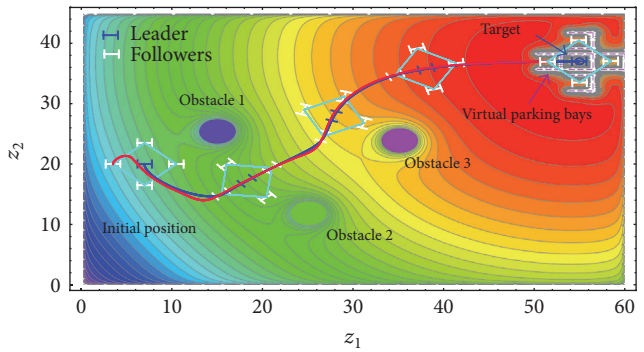


Figure 12: The evolution of the 5-robot C-ITS and the contour plot in the presence of obstacles in Scenario 3.

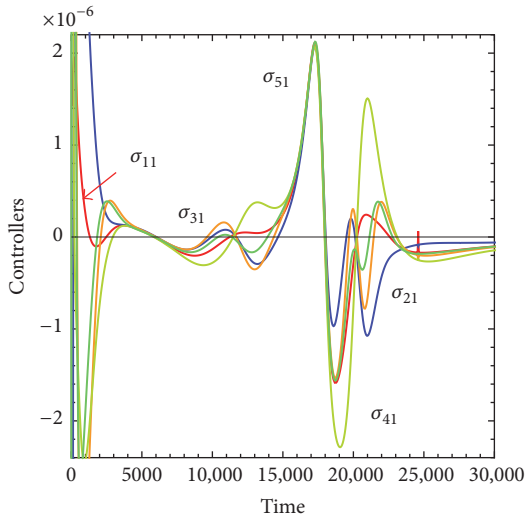


Figure 13: Translational accelerations, σ_{1i} for Scenario 3.

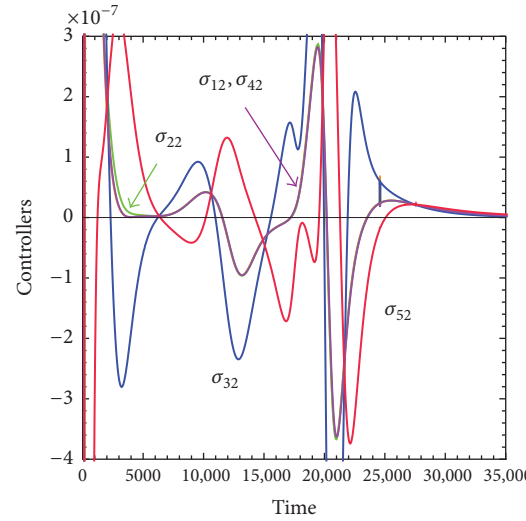


Figure 14: Rotational accelerations, σ_{2i} for Scenario 3.

of the locally rigid formation for a C-ITS which navigates in a constrained environment. This leader-follower scheme uses a Cartesian coordinate system fixed on the leader's body based on the concept of an instantaneous corotating frame of reference to uniquely assign a position to each follower. The advantage of such an approach is the complete avoidance of singularities inherent to the polar coordinate representations.

The new time-invariant acceleration-based controllers produce feasible trajectories and ensured a nice convergence of the system to its equilibrium state while satisfying the necessary kinematic and dynamic constraints. The scheme enables the vehicles to follow a predetermined leader while maintaining a locally rigid formation. The robustness of the proposed scheme is established by considering the effect of noise on the formation, while stability of the system is guaranteed using the direct method of Lyapunov. The assumptions of no slippage and pure rolling of the car-like mobile robots are also considered to generate the nonholonomic constraints

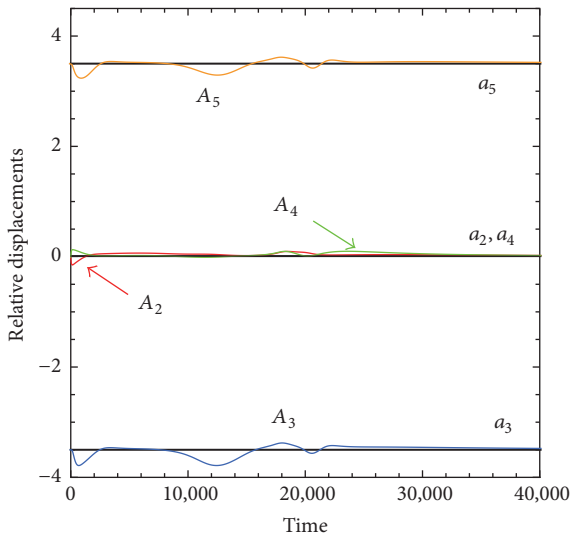


FIGURE 15: Relative distance error, A_k to a_k for Scenario 3.

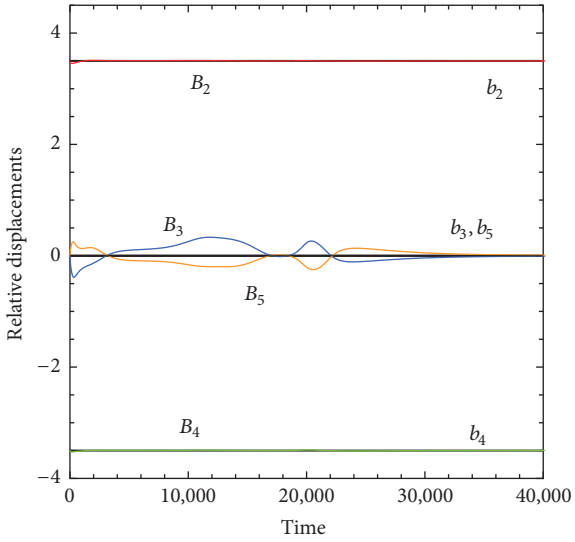


FIGURE 16: Relative distance error, B_k to b_k for Scenario 3.

on the given system. Further, the LbCS utilizes the *minimum distance technique* and parking bays to guarantee desired parking maneuvers and establish feasible prescribed posture of each vehicle in its designated parking bay.

This paper is a theoretical exposition into the applicability of an amalgamation of the new artificial potential field method, LbCS, and the leader-follower scheme. We restrict ourselves to showing the effectiveness of the control laws using computer-based simulations of interesting scenarios and numerical proofs. We shall be undertaking experiential design of the above system to see its effectiveness in the future. Future research will also address irregular shaped multimodal obstacles in partially known or completely known environments with the adaptive usage of sensing zones and relevant parameters. The optimization process of parameters within the LbCS still remains an open problem that warrants a careful thought.

Appendix

$$f_{11} = \left[1 + \sum_{l=1}^{q} \frac{\alpha_{1l}}{\mathrm{FO}_{1l}} + \sum_{s=1}^{2} \frac{\beta_{1s}}{U_{1s}} + \sum_{p=1}^{2} \frac{\kappa_{1p}}{W_{1p}} + \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{\eta_{1j}}{\mathrm{MO}_{1j}} + \sum_{k=1}^{3n} \frac{\gamma_{1k}}{LS_{1k}}\right](x_1 - p_{11}) + \sum_{r=2}^{n}\left[1 + \sum_{l=1}^{q} \frac{\alpha_{1l}}{\mathrm{FO}_{1l}} + \sum_{s=1}^{2} \frac{\beta_{1s}}{U_{1s}}\right]\left[-(A_r - a_r)\cos\theta_1 + (B_r - b_r)\sin\theta_1\right] + \sum_{r=2}^{n}\left[\sum_{p=1}^{2} \frac{\kappa_{1p}}{W_{1p}} + \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{\eta_{1j}}{\mathrm{MO}_{1j}} + \sum_{k=1}^{3n} \frac{\gamma_{1k}}{LS_{1k}}\right]\left[-(A_r - a_r)\cos\theta_1 + (B_r - b_r)\sin\theta_1\right] - G_1 \sum_{k=1}^{3} \frac{\gamma_{1k}}{LS_{1k}^2(\mathbf{x})}\left\{\left[1 - (a_{k2} - a_{k1})\, d_k\right](x_1 - Lx_{1k}) - (b_{k2} - b_{k1})\, d_k\,(y_1 - Ly_{1k})\right\} - G_1\left[\sum_{l=1}^{q} \frac{\alpha_{1l}}{\mathrm{FO}_{1l}^2}(x_1 - o_{l1}) + 2\sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{\eta_{1j}}{\mathrm{MO}_{1j}^2}(x_1 - x_j)\right],$$

$$f_{12} = \left[1 + \sum_{l=1}^{q} \frac{\alpha_{1l}}{\mathrm{FO}_{1l}} + \sum_{s=1}^{2} \frac{\beta_{1s}}{U_{1s}} + \sum_{p=1}^{2} \frac{\kappa_{1p}}{W_{1p}} + \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{\eta_{1j}}{\mathrm{MO}_{1j}} + \sum_{k=1}^{3n} \frac{\gamma_{1k}}{LS_{1k}}\right](y_1 - p_{12}) + \sum_{h=2}^{n}\left[1 + \sum_{l=1}^{q} \frac{\alpha_{1l}}{\mathrm{FO}_{1l}} + \sum_{s=1}^{2} \frac{\beta_{1s}}{U_{1s}}\right]\left[-(A_h - a_h)\sin\theta_1 - (B_h - b_h)\cos\theta_1\right] + \sum_{h=2}^{n}\left[\sum_{p=1}^{2} \frac{\kappa_{1p}}{W_{1p}} + \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{\eta_{1j}}{\mathrm{MO}_{1j}} + \sum_{k=1}^{3n} \frac{\gamma_{1k}}{LS_{1k}}\right]\left[-(A_h - a_h)\sin\theta_1 - (B_h - b_h)\cos\theta_1\right] - G_1\left[\sum_{l=1}^{q} \frac{\alpha_{1l}}{\mathrm{FO}_{1l}^2}(y_1\right.$$

$$
\begin{aligned}
&- o_{l2}) + 2\sum_{\substack{j=1\\ j\neq i}}^{n} \frac{\eta_{1j}}{\mathrm{MO}_{1j}^2}\left(y_1 - y_j\right)\Bigg] - \frac{\kappa_{12}}{W_{12}^2} + \frac{\kappa_{14}}{W_{14}^2} \\
&- G_1 \sum_{k=1}^{3} \frac{\gamma_{1k}}{LS_{1k}^2(\mathbf{x})} \left\{\left[1 - \left(b_{k2} - b_{k1}\right) r_k\right]\left(y_1 - Ly_{1k}\right)\right. \\
&\left. - \left(a_{k2} - a_{k1}\right) r_k \left(x_1 - Lx_{1k}\right)\right\}, \\
f_{13} = &\left[\sum_{l=1}^{q} \frac{\alpha_{1l}}{\mathrm{FO}_{1l}} + \sum_{s=1}^{2} \frac{\beta_{1s}}{U_{1s}} + \sum_{p=1}^{2} \frac{\kappa_{1p}}{W_{1p}} + \sum_{\substack{j=1\\ j\neq i}}^{n} \frac{\eta_{1j}}{\mathrm{MO}_{1j}}\right. \\
&\left. + \sum_{k=1}^{3n} \frac{\gamma_{1k}}{LS_{1k}}\right] \rho_1 \left(\theta_1 - p_{13}\right) - \sum_{i=2}^{n} \left[\sum_{l=1}^{q} \frac{\alpha_{1l}}{\mathrm{FO}_{1l}} + \sum_{s=1}^{2} \frac{\beta_{1s}}{U_{1s}}\right. \\
&\left. + \sum_{p=1}^{2} \frac{\kappa_{1p}}{W_{1p}} + \sum_{\substack{j=1\\ j\neq i}}^{n} \frac{\eta_{1j}}{\mathrm{MO}_{1j}} + \sum_{k=1}^{3n} \frac{\gamma_{1k}}{LS_{1k}}\right] \rho_i \left(\theta_i - \theta_1\right), \\
g_{11} &= 1 + G_1 \frac{\beta_{11}}{U_{11}^2}, \\
g_{12} &= 1 + G_1 \frac{\beta_{12}}{U_{12}^2},
\end{aligned}
\tag{A.1}
$$

and, for $i = 2, \ldots, n$,

$$
\begin{aligned}
f_{i1} = &\left[1 + \sum_{l=1}^{q} \frac{\alpha_{il}}{\mathrm{FO}_{il}} + \sum_{s=1}^{2} \frac{\beta_{is}}{U_{is}} + \sum_{p=1}^{2} \frac{\kappa_{ip}}{W_{ip}} + \sum_{\substack{j=1\\ j\neq i}}^{n} \frac{\eta_{ij}}{\mathrm{MO}_{ij}}\right. \\
&\left. + \sum_{k=3i-2}^{3i} \frac{\gamma_{ik}}{LS_{ik}}\right] \left(A_i - a_i\right) \cos\theta_1 - \left[1 + \sum_{l=1}^{q} \frac{\alpha_{il}}{\mathrm{FO}_{il}}\right. \\
&\left. + \sum_{s=1}^{2} \frac{\beta_{is}}{U_{is}} + \sum_{p=1}^{2} \frac{\kappa_{ip}}{W_{ip}} + \sum_{\substack{j=1\\ j\neq i}}^{n} \frac{\eta_{ij}}{\mathrm{MO}_{ij}} + \sum_{k=3i-2}^{3i} \frac{\gamma_{ik}}{LS_{ik}}\right] \left(B_i\right. \\
&\left. - b_i\right) \sin\theta_1 - G_i \sum_{k=3i-2}^{3i} \frac{\gamma_{ik}}{LS_{ik}^2(\mathbf{x})} \left\{\left[1\right.\right. \\
&\left.\left. - \left(a_{k2} - a_{k1}\right) d_k\right]\left(x_i - Lx_{ik}\right) - \left(b_{k2}\right.\right. \\
&\left.\left. - b_{k1}\right) d_k \left(y_i - Ly_{ik}\right)\right\} \\
&- G_i \left[\sum_{l=1}^{q} \frac{\alpha_{il}}{\mathrm{FO}_{il}^2}\left(x_i - o_{l1}\right) + 2\sum_{\substack{j=1\\ j\neq i}}^{n} \frac{\eta_{ij}}{\mathrm{MO}_{ij}^2}\left(x_i - x_j\right)\right], \\
f_{i2} = &\left[1 + \sum_{l=1}^{q} \frac{\alpha_{il}}{\mathrm{FO}_{il}} + \sum_{s=1}^{2} \frac{\beta_{is}}{U_{is}} + \sum_{p=1}^{2} \frac{\kappa_{ip}}{W_{ip}} + \sum_{\substack{j=1\\ j\neq i}}^{n} \frac{\eta_{ij}}{\mathrm{MO}_{ij}}\right. \\
&\left. + \sum_{k=1}^{3n} \frac{\gamma_{ik}}{LS_{ik}}\right] \left(A_i - a_i\right) \sin\theta_1 + \left[1 + \sum_{l=1}^{q} \frac{\alpha_{il}}{\mathrm{FO}_{il}}\right. \\
&\left. + \sum_{s=1}^{2} \frac{\beta_{is}}{U_{is}} + \sum_{p=1}^{2} \frac{\kappa_{ip}}{W_{ip}} + \sum_{\substack{j=1\\ j\neq i}}^{n} \frac{\eta_{ij}}{\mathrm{MO}_{ij}} + \sum_{k=1}^{3n} \frac{\gamma_{ik}}{LS_{ik}}\right] \left(B_i\right. \\
&\left. - b_i\right) \cos\theta_1 - G_i \sum_{k=3i-2}^{3i} \frac{\gamma_{ik}}{LS_{ik}^2(\mathbf{x})} \left\{\left[1\right.\right. \\
&\left.\left. - \left(b_{k2} - b_{k1}\right) r_k\right]\left(y_i - Ly_{ik}\right) - \left(a_{k2} - a_{k1}\right) r_k \left(x_i\right.\right. \\
&\left.\left. - Lx_{ik}\right)\right\} - G_i \left[\sum_{l=1}^{q} \frac{\alpha_{il}}{\mathrm{FO}_{il}^2}\left(y_i - o_{l1}\right)\right. \\
&\left. + 2\sum_{\substack{j=1\\ j\neq i}}^{n} \frac{\eta_{ij}}{\mathrm{MO}_{ij}^2}\left(y_i - y_j\right)\right] - \frac{\kappa_{i2}}{W_{i2}^2} + \frac{\kappa_{i4}}{W_{i4}^2}, \\
f_{i3} = &\left[\sum_{l=1}^{q} \frac{\alpha_{il}}{\mathrm{FO}_{il}} + \sum_{s=1}^{2} \frac{\beta_{is}}{U_{is}} + \sum_{p=1}^{2} \frac{\kappa_{ip}}{W_{ip}} + \sum_{\substack{j=1\\ j\neq i}}^{n} \frac{\eta_{ij}}{\mathrm{MO}_{ij}}\right. \\
&\left. + \sum_{k=1}^{3n} \frac{\gamma_{ik}}{LS_{ik}}\right] \rho_i \left(\theta_i - \theta_1\right), \\
g_{i1} &= 1 + G_i \frac{\beta_{i1}}{U_{i1}^2}, \\
g_{i2} &= 1 + G_i \frac{\beta_{i2}}{U_{i2}^2}.
\end{aligned}
\tag{A.2}
$$

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. Festag, "Cooperative intelligent transport systems standards in Europe," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 166–172, 2014.

[2] E. Ben Hamida, H. Noura, and W. Znaidi, "Security of cooperative intelligent transport systems: Standards, threats analysis

and cryptographic countermeasures," *Electronics*, vol. 4, no. 3, pp. 380–423, 2015.

[3] T. Balch and M. Hybinette, "Social potentials for scalable multi-robot formations," *Proceedings-IEEE International Conference on Robotics and Automation*, vol. 1, pp. 73–80, 2000.

[4] C. De La Cruz and R. Carelli, "Dynamic model based formation control and obstacle avoidance of multi-robot systems," *Robotica*, vol. 26, no. 3, pp. 345–356, 2008.

[5] F. Arrichiello, *Coordination Control of Multiple Mobile Robots [Ph.D. thesis]*, Universita Degli Studi Di Cassino, Cassino, Italy, November 2006.

[6] T. H. A. Van Den Broek, N. Van De Wouw, and H. Nijmeijer, "Formation control of unicycle mobile robots: a virtual structure approach," in *Proceedings of the 48th IEEE Conference on Decision and Control held jointly with 2009 28th Chinese Control Conference, CDC/CCC 2009*, pp. 8328–8333, Shanghai, China, December 2009.

[7] H. Yamaguchi, "A distributed motion coordination strategy for multiple nonholonomic mobile robots in cooperative hunting operations," *Robotics and Autonomous Systems*, vol. 43, no. 4, pp. 257–282, 2003.

[8] Y. H. Esin and M. Unel, "Formation control if nonholonomic mobile robots using implicit polynomials and elliptic fourier descriptors," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 18, no. 5, pp. 765-80, 2010.

[9] R. Fierro, A. K. Das, V. Kumar, and J. P. Ostrowski, "Hybrid control of formations of robots," in *Proceedings of the IEEE Conference Robotics and Automation*, vol. 1, pp. 374–379, Seoul, Korea, May 2001.

[10] W. L. Seng, J. C. Barca, and Y. A. Şekercioğlu, "Distributed formation control of networked mobile robots in environments with obstacles," *Robotica*, vol. 12, pp. 1–13, 2014.

[11] F. Fahimi, "Full formation control for autonomous helicopter groups," *Robotica*, vol. 26, no. 2, pp. 143–156, 2008.

[12] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[13] R. Falconi, L. Sabattini, C. Secchi, C. Fantuzzi, and C. Melchiorri, "Edge-weighted consensus-based formation control strategy with collision avoidance," *Robotica*, vol. 33, no. 2, pp. 332–347, 2015.

[14] B. Sharma, *New Directions in the Applications of the Lyapunov-based Control Scheme to the Findpath Problem [Ph.D. thesis]*, University of the South Pacific, Suva, Fiji Islands, July 2008.

[15] E. Yang, D. Gu, and H. Hu, "Nonsingular formation control of cooperative mobile robots via feedback linearization," in *Proceedings of the IEEE IRS/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, pp. 3652–3657, August 2005.

[16] W. Kang, N. Xi, J. Tan, and Y. Wang, "Formation control of multiple autonomous robots:theory and experimentation," *Intelligent Automation & Soft Computing*, vol. 10, no. 2, pp. 1–17, 2004.

[17] A. Bazoula, M. S. Djouadi, and H. Maaref, "Formation control of multi-robots via fuzzy logic technique," *International Journal of Computers, Communications & Control*, vol. 3, pp. 179–184, 2008.

[18] N. H. M. Li and H. H. T. Liu, "Formation UAV flight control using virtual structure and motion synchronization," in *Proceedings of the 2008 American Control Conference (ACC '08)*, pp. 1782–1787, Washington, Wash, USA, June 2008.

[19] W. Ren and R. W. Beard, "Virtual structure based spacecraft formation control with formation feedback," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, California, Calif, USA, August 2002.

[20] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," in *Proceedings of the 2001 IEEE International Conference on Robotics and Automation (ICRA '01)*, pp. 3961–3966, Seoul, Republic of Korea, May 2001.

[21] N. Sorensen and W. Ren, "A unified formation control scheme with a single or multiple leaders," in *Proceedings of the 2007 American Control Conference (ACC '07)*, pp. 5412–5418, New York, NY, USA, July 2007.

[22] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "Leader-follower formation control of nonholonomic mobile robots with input constraints," *Automatica*, vol. 44, no. 5, pp. 1343–1349, 2008.

[23] W. J. Dong, Y. Guo, and J. A. Farrell, "Formation control of nonholonomic mobile robots," in *Proceedings of the 2006 American Control Conference*, pp. 5902-5607, Minneapolis, MN, USA, June 2006.

[24] K. Raghuwaiya and S. Singh, "Formation types of multiple steerable 1-trailer mobile robots via split/rejoin maneuvers," *New Zealand Journal of Mathematics*, vol. 43, pp. 7–21, 2013.

[25] B. Sharma, J. Vanualailai, and A. Prasad, "Formation control of a swarm of mobile manipulators," *Rocky Mountain Journal of Mathematics*, vol. 41, no. 3, pp. 909–940, 2011.

[26] B. Sharma, J. Vanualailai, and A. Prasad, "A $d\phi$ strategy: facilitating dual-formation control of a virtually connected team," *Journal of Advanced Transportation*, vol. 2017, Article ID 9213805, 17 pages, 2017.

[27] K. Raghuwaiya, B. Sharma, and J. Vanualailai, *Advanced Computer and Communication Engineering Technology: Lecture Notes in Electrical Engineering*, Cooperative Control of Multi Robot Systems with a low-degree formation, Springer, 2015.

[28] R. W. Brockett, *Differential Geometric Control Theory*, Asymptotic Stability and Feedback Stabilisation, Springer, 1983.

[29] A. Prasad, B. Sharma, and J. Vanualailai, "A new stabilizing solution for motion planning and control of multiple robots," *Robotica*, vol. 34, no. 5, pp. 1071–1089, 2016.

Hindawi
Journal of Advanced Transportation
Volume 2018, Article ID 6392697, 10 pages
https://doi.org/10.1155/2018/6392697

Research Article

Global and Local Path Planning Study in a ROS-Based Research Platform for Autonomous Vehicles

Pablo Marin-Plaza, Ahmed Hussein, David Martin, and Arturo de la Escalera

Universidad Carlos III de Madrid, Leganes, Spain

Correspondence should be addressed to Pablo Marin-Plaza; pamarinp@ing.uc3m.es

Received 17 November 2017; Revised 16 January 2018; Accepted 17 January 2018; Published 22 February 2018

Academic Editor: Aboelmaged Noureldin

The aim of this work is to integrate and analyze the performance of a path planning method based on Time Elastic Bands (TEB) in real research platform based on Ackermann model. Moreover, it will be proved that all modules related to the navigation can coexist and work together to achieve the goal point without any collision. The study is done by analyzing the trajectory generated from global and local planners. The software prototyping tool is Robot Operating System (ROS) from Open Source Robotics Foundation and the research platform is the iCab (Intelligent Campus Automobile) from University Carlos III. This work has been validated from a test inside the campus where the iCab has performed the navigation between the starting point and the goal point without any collision. During the experiment, we proved the low sensitivity of the TEB method to variations of the vehicle model configuration and constraints.

1. Introduction

Autonomous vehicles base their decisions on planner modules that create the collision-free waypoints in the path to reach the destination point. These modules are capable of finding the optimal solution minimizing the computational time and distance covered by the vehicle avoiding static and dynamic obstacles. Some research groups test these modules on real vehicles such as PROUD in Parma in 2013 [1] or Karlsruhe Institute of Technology (KIT) driving around the cities with advanced technology in perception, navigation, and decision making [2]. In all cases, the navigation module is divided into a global planner and a local planner, where the first one finds the optimal path with a prior knowledge of the environment and static obstacles, and the second one recalculates the path to avoid dynamic obstacles.

The main contribution of this work is to demonstrate the integration of using TEB local planner in an Ackermann model with real test. The platform used is the iCab (Intelligent Campus Automobile) which works with the software prototyping tool ROS. Additionally, it has been discovered that the capacity of the vehicle to follow the local plan with accuracy is not mandatory. The vehicle is able to reach the goal with modest errors when following the local plan. This characteristic is really important in real life when the vehicle model depends on multiple factors as the pressure of the tires, the number of passengers, and the battery of the vehicle. When this mathematical model varies from the nominal values, the vehicle will not reach point by point the provided local plan, so TEB local planner is more robust to face changes in the configuration of the mathematical model.

In order to prove that the TEB local planner is able to work along the other processes involved in the vehicle, it is necessary to solve several problems related to the localization in the environment (initial localization), the ego-motion of the vehicle (odometry), environment understanding (perception and mapping), which path should the vehicle take (planning), and movement (low level control). Hence, this document gathers all the solutions for each task for navigation and then tests the Dijkstra method for global planning and the Time Elastic Bands method used for local planning.

This document is divided into six sections. Section 2 describes current planning methods used in the autonomous vehicles research field. Section 3 describes the research

platform and the modules related to the navigation. Section 4 describes the experimental setup. Section 5 summarizes the results. Finally, the last section contains the conclusions and the future work.

2. Path Planning

The path planning problem is a well-known NP-hardness [3] where the complexity increases with the degrees of freedom of the vehicle. In this work, the ground vehicle is reduced to a 2D space of a single plane X-Y where the Z component is neglected and the vehicle is restricted by some constraints due to the Ackermann configuration [4]. The aim of solving this NP-hardness is not to find one solution that connects the start point and the goal point, but the optimal solution with the minimum distance and the smoothest maneuvers and without hitting any known obstacles. Usually, this module is divided into the global planner, which uses a priori information of the environment to create the best possible path, if any, and the local planner, which recalculates the initial plan to avoid possible dynamic obstacles. The generated plan is a collection of waypoints for global, $P_i = (x_i, y_i)^T$, and local $P_i = (x_i, y_i, \theta_i)^T$, $\in \Re^2 \times S^1$, where S is the navigation plane where the vehicle is able to move.

2.1. Global Planner. As said before, the global planner requires a map of the environment to calculate the best route. Depending on the analysis of the map, some methods are based on Roadmaps like Silhouette [5] proposed by Canny in 1987 or Voronoi [6] in 2007. Some of them solve the problem by assigning a value to each region of the roadmap in order to find the path with minimum cost. Some examples are the Dijkstra [7] algorithm, Best First [8], and A^* [9]. Another approach is by dividing the map into small regions (cells) called cell decomposition as mentioned in [10]. A similar approach by using potential fields is described in [11], the most extended algorithm used few years ago rapidly exploring random trees [12] or the new approach based on neural networks [13]. Some solutions combine the aforementioned algorithms improving the outcome at the cost of high computational power.

So, for this work and for testing purposes, the selected method is the Dijkstra algorithm because of the simplicity for debugging and its good performance on the experiments. It is necessary to remark that the goal of this paper is not to find the best and fastest solution, but it is to prove that all modules of navigation are working together and the vehicle is able to navigate from one point to another.

Dijkstra uses the roadmap approach to convert the problem into a graphic search method using the information of a grid cell map. This method starts with a set of candidate nodes where the vehicle is able to navigate (free space) assigning a cost value to each of them. From the starting point, this value is increased by the necessary number of nodes to pass through to reach each node. For example, in Figure 1, the free space around the starting point takes the value of one unit and for the second generation of neighbors takes the value of two units and so on. For each cell with value, a checked cell is assigned and the method continues with the next generation of neighbors until the goal point is reached. The minimum value of the sum of all nodes from the starting point and the end point is the shortest path. After the success of finding the global path from the start to the goal, all the selected nodes are translated into positions in the reference axes as the form $P_i = (x_i, y_i)^T$. The global planner divides the map into nodes for each free cell but the outcome is not smooth and some points are not compliant with the vehicle geometry and kinematics.

2.2. Local Planner. In order to transform the global path into suitable waypoints, the local planner creates new waypoints taking into consideration the dynamic obstacles and the vehicle constraints. So, to recalculate the path at a specific rate, the map is reduced to the surroundings of the vehicle and is updated as the vehicle is moving around. It is not possible to use the whole map because the sensors are unable to update the map in all regions and a large number of cells would raise the computational cost. Therefore, with the updated local map and the global waypoints, the local planning generates avoidance strategies for dynamic obstacles and tries to match the trajectory as much as possible to the provided waypoints from the global planner. Different approaches are based on trajectory generation using Clothoids lines [16], Bezier lines [17], arcs and segments [18], or splines [19]. All of them create intermediate waypoints following the generated trajectory. The selected method for local planning is Time Elastic Bands [15, 20] which consists of deforming the initial global plan considering the kinematic model of the vehicle and updating the local path based on dynamic obstacles or the possible deviation from the path. This method has been selected because of the integration of the trajectory planning and obstacle detection with avoidance at low computational cost fulfilling the requirements of the local planner in the research platform iCab.

Time Elastic Bands (TEB) create a sequence of intermediate vehicle poses $p_i = (xi, yi, \theta_i)^T$ modifying the initial global plan. It requires the velocity and acceleration limits of the vehicle, the security distance of the obstacles and the geometric, and kinematic and dynamic constraints of the vehicle. All of this configuration generates a set of commands for speed (v) and steering angle (δ), as $\text{cmd}_i = (v_i, \delta_i)^T$, required to achieve the intermediate waypoints, while the vehicle is moving. Figure 2 shows a robot moving from the starting point to the goal point passing through four waypoints in the smoothest way possible between obstacles. One of the drawbacks of this method is the influence of the dynamic obstacles direction in the generation of the recalculated local path. For example, when a pedestrian is crossing from east to west the trajectory generated which connects a point from the south to the north (Figure 3), the recalculation of the new trajectory will still be on the left of the obstacle but further. The solution is the creation of three elastic bands: one updated each iteration, one is recalculated from the original path, and the last one is created from scratch each time. The selected one is the shortest of the three that is compliant with all requirements.

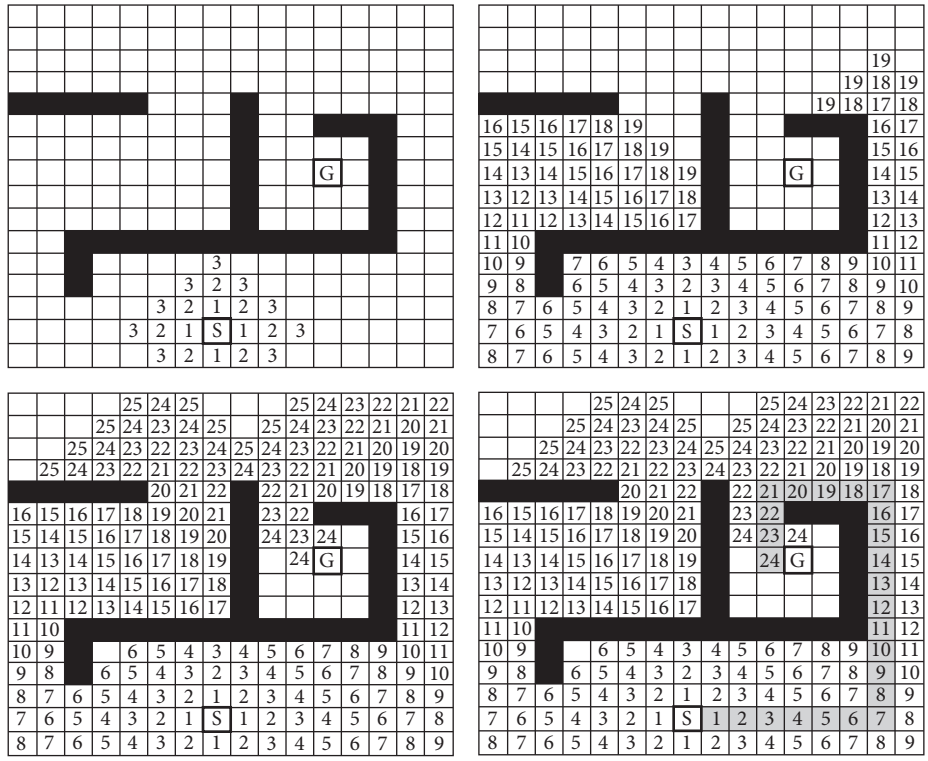


Figure 1: Dijkstra method to find the optimal path [14].

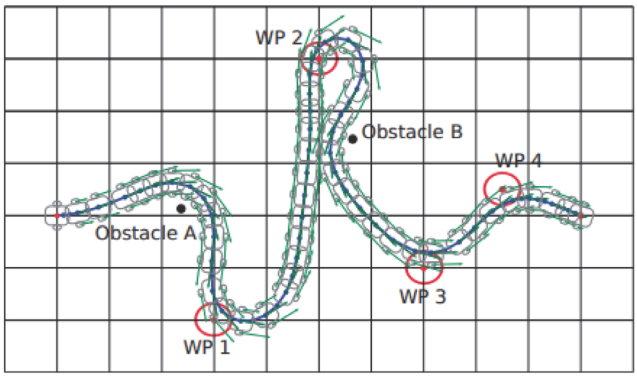


Figure 2: Description of Time Elastic Bands from [15].

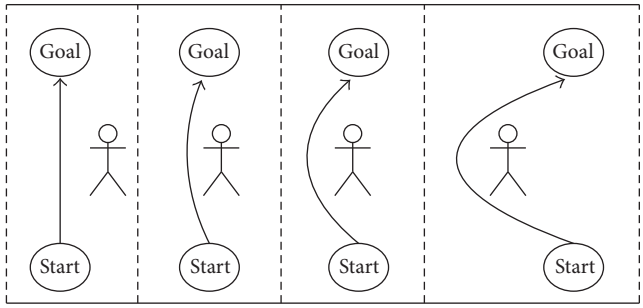


Figure 3: Pedestrian crossing from east to west of the trajectory generated.

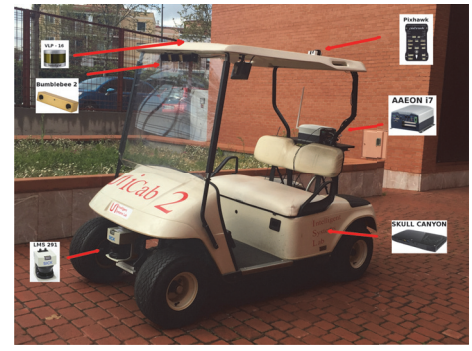


Figure 4: iCab with sensors.

3. Platform Description

The research platform is a modified golf cart equipped with several sensors such as wheel encoders, a Lidar with 16 laser planes on the top, a single plane laser located in the front, a stereo camera located in the front, and a GPS-IMU-Compass located on the top, as shown in Figure 4. The movement of the vehicle is done by a 36 V DC motor for the linear acceleration and another 36 V DC motor for the steering wheel. More information and tools are available in [21–23].

Additionally, the vehicle includes two computers to gather information from the sensors, process it, and perform the navigation decisions. The main computer is in charge of the wheel encoders, images, laser, imu, GPS, and compass

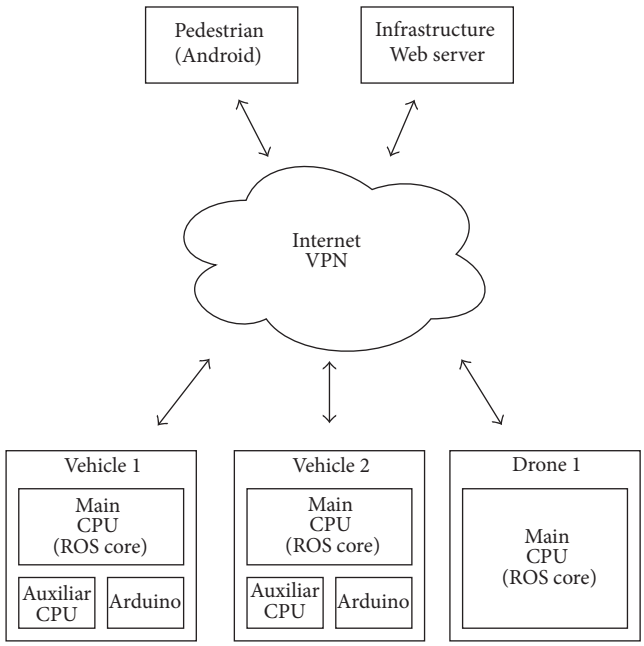


Figure 5: System communications.

and generates the maps. Therefore, the system provides wheel odometry, obstacle detection, and maps. The second computer processes the point cloud generated from the Lidar and computes the Lidar odometry. The software used, as aforementioned said, is ROS. This tool allows the vehicle to share and synchronize messages between nodes in the same computer and, additionally, with the computers and microcontrollers in the vehicle by the network using ROS core master. Furthermore, using a master discovery tool for all the ROS core masters in the network (one core master for each vehicle), the systems are able to share messages between vehicles, infrastructures, and pedestrians [24]. The use of one core master for each vehicle allows the inner communications even when there is no Internet connection, as shown in Figure 5.

A brief graphical description of the nodes and connections is shown in Figure 6. From left to right, the nodes are as follows:

(i) *Global map* is responsible for loading the global grid map for the campus as *navigation_msgs/OccupancyGrid.msg*. The selected grid size has a resolution of 0.156 m used in the analysis of the previously recorded point cloud obtained with the Lidar (a priori knowledge).

(ii) *Global planner* uses the Dijkstra algorithm to create the/*global_plan* to reach the goal point using the global map. This path message is a standard type *navigation_msgs/Path.msg* which contains the waypoints without the orientation.

(iii) *Lidar odometry* and *wheel odometry* are in charge of creating the ego-motion measurements from the starting point of the vehicle. The messages used are the type of *navigation_msgs/Odometry.msg*.

(iv) *Odometry manager* is responsible for the fusion and adding the initial localization for the conversion between global and local odometry.

(v) *Costmap 2D* uses the laser sensor information to create a local Costmap in front of the vehicle. The Costmap values represent the proximity of the vehicle to an obstacle using the geometry of the vehicle. The resulting map has the obstacles inflated by a security perimeter where the vehicle should not allow entering. Additionally, this map is used for TEB local planner in order to modify the local path based on the distance from the vehicle to the obstacles.

(vi) *TEB local planner* needs a global plan, Costmap, and vehicle odometry. It publishes the local plan (at a specific rate) with a configurable lookahead distance, and additionally, it publishes the control commands *ackermann_cmd* with the type *ackermannDriveStamped.msg*. The selected lookahead distance is 10 m because it is suitable to avoid big obstacles like other vehicles or groups of students.

(vii) *Obstacle manager* is responsible for sending a warning message to the movement manager for possible threats like pedestrians or obstacles in front of the vehicle. The type of this message is custom and has a list of obstacles with the information of distance and area for each obstacle. This node also returns the distance to the closest obstacle to take into consideration for emergency brake.

(viii) *Task manager* is in charge of selecting the task for the autonomous vehicle. For other projects, it is possible to configure different behaviors like platooning mode, drive between boundaries, or manual mode. This node receives information of the possible threats in the environment in order to stop the vehicle if necessary. This package publishes the reference velocity and reference steering angle.

(ix) *PID velocity* is responsible for creating the PWM necessary to the rear wheels DC motor. It takes as input the current velocity. The type of message is standard as *std_msgs/Float64.msg*.

(x) *Movement manager* is a driver in charge of checking all limits and possible wrong velocity messages. It sends the desired acceleration to the DC motor and the selected angle for the steering angle.

Transformation Trees (tf). One of the most important aspects to consider when working with vehicle odometry is the frame of reference and the coordinates. This node is in charge of transforming the movement of the vehicle from its ego-motion local reference to the global reference adding the initial position and orientation. Therefore, global and local path need the transformation between local coordinates and global coordinates. The *odometry manager* module is in charge of fusing the odometries and publishing the transformation between global and local coordinates using the initial localization in the map of the vehicle. Figure 7 shows all the frames involved in the architecture for one vehicle. The

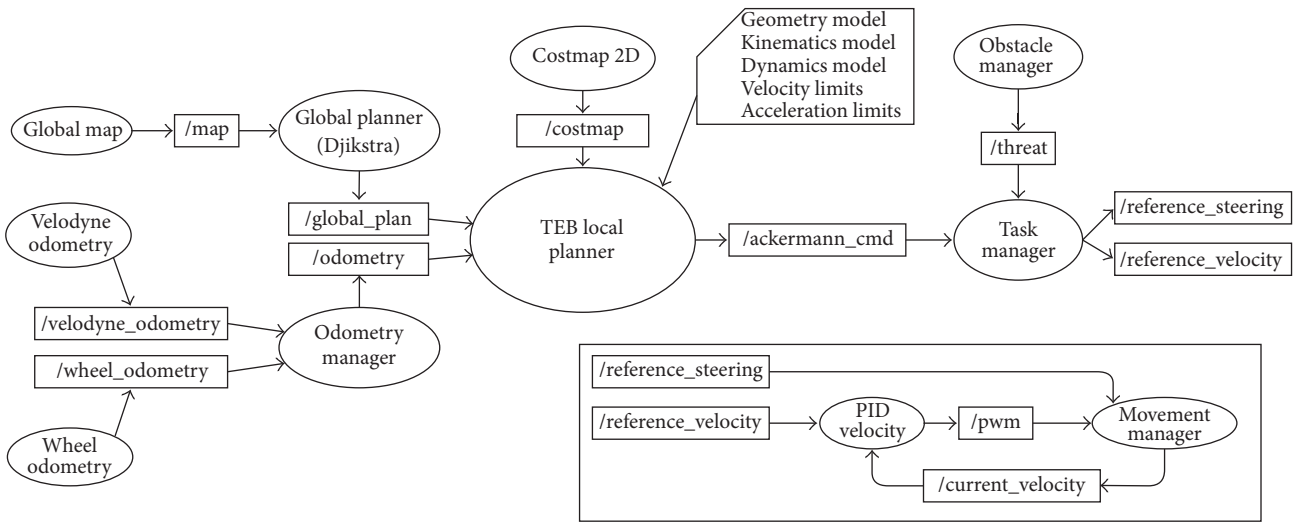


Figure 6: Software nodes sharing messages by topics in the partial software architecture.

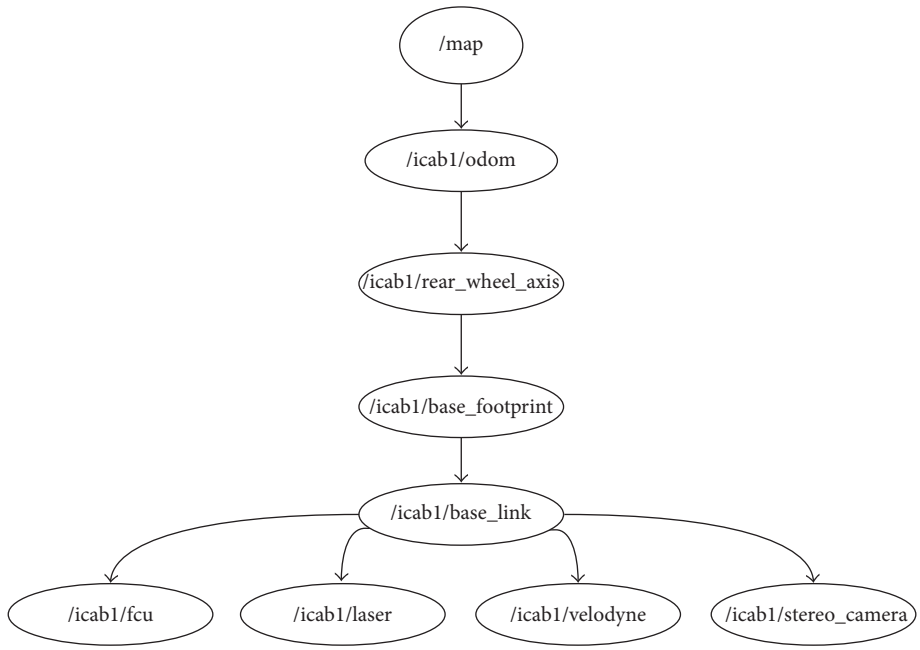


Figure 7: Transformation Tree for iCab1 showing all the sensors.

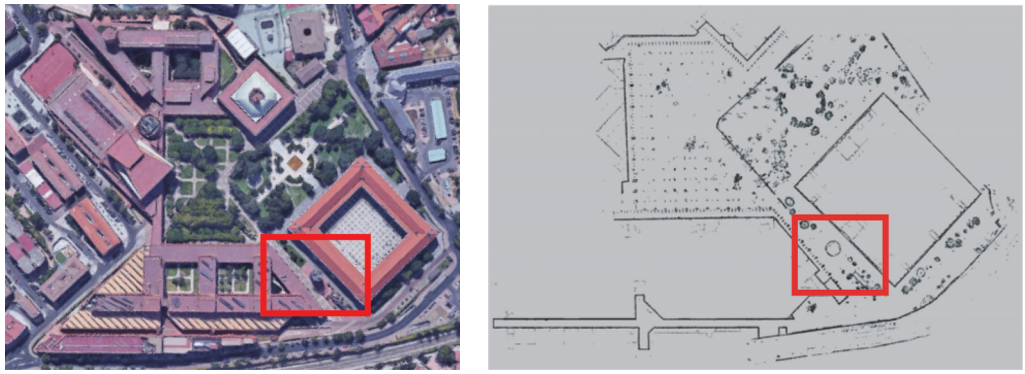
appropriate configuration of these transformations allows the system to collaborate with other coexisting heterogeneous vehicles under a vehicle namespace such as in this case icab1.

4. Experimental Setup

Aiming to test the path planner in the real platform, a simple test has been prepared where the vehicle navigates from a starting point and finishes in a goal point. The selected location takes place in a zone where a circular building is occluding the direct path trajectory. The data is recorded into a bag file, while the vehicle performs the experiment for posterior offline analysis. In Figure 8, the global map used to calculate the global plan is shown and the experiment zone is located inside of the red square.

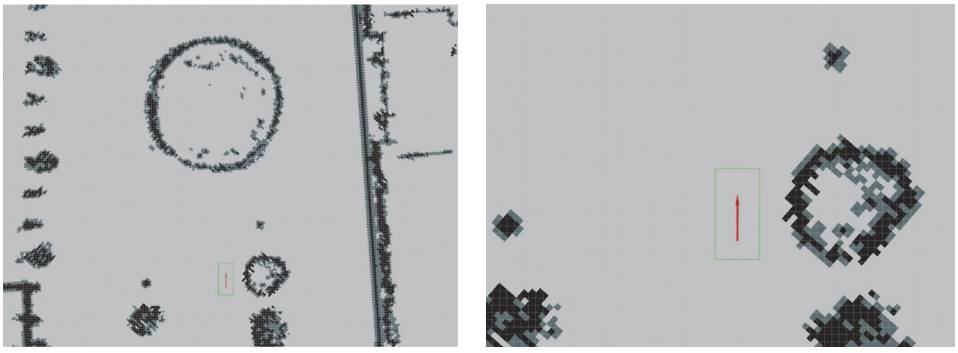
This map has been generated using the Lidar 3D points with the Lidar odometry and mapping [25] adapted to work with the correct axes reference. Afterwards, the point cloud has been processed to extract the floor at a specific height. The last step is to project this point cloud to the 2D space to have all the static obstacles (notice that the cup of the trees is taken into account as an obstacle due to the possible collision with the leaves and the top of the vehicle).

The representation of the vehicle, the maps, and the movements are easily analyzed using the RVIZ tool. Figure 9 describes the localization of the vehicle in the environment



(a) (b)

Figure 8: Satellite image of the Campus granted by Google Maps on (a). Campus gridmap on (b), the experiment is located inside of the red square.



(a) (b)

Figure 9: On (a), starting point of the vehicle oriented to the local odometry. On (b), zoom in the details of the vehicle geometry.

where the green rectangular shape represents the geometry and the red arrow represents the orientation.

After introducing the goal point, the representation of the lines for the paths is shown in Figure 10 where the blue line is the global path generated from the global planning using the current localization and the goal point. The input goal point is generated from the RVIZ visualizer using *move_base_msgs/MoveBaseGoal.msg* and clicking on the screen. The orange line is the local path generated from the TEB local planner and each cycle of the node is recalculated.

5. Results

All trajectories generated from the TEB local planner are evaluated offline by the data recorded, while the experiment is running.

5.1. Evaluation Metrics. The evaluation metrics used in this section are the Euclidean distance (DGL) between the Global Plan Waypoints (GPW) and Local Plan Waypoints (LPW) as shown in (1). The algorithm to find the best local plan is constantly updating it in order to avoid obstacles in the route or because the vehicle is not able to follow the previous local path.

$$\mathrm{DGL}_i = \sqrt{(\mathrm{GPW}_i - \mathrm{LPW}_i)^2} \quad \forall i \in Q. \tag{1}$$

5.2. Data Analysis. The experiment has been done by the lookahead distance configured to 10 m for the TEB algorithm because if an obstacle is above the global plan (a dynamic obstacle or a static obstacle which are not on the global map), this lookahead distance should be big enough to find another path even if it is far away from the global plan. Figure 11 represents the global plan generated with Dijkstra method on blue. In Figure 12, each line represents the local plan generated from the position of the vehicle while moving. The representation of multiple lines is due to the recalculation of the local trajectories. It is possible to appreciate the details of some lines that do not end into the goal point because of the lookahead distance configuration. Figure 13 represents the position of the vehicle at each moment in red dots. It is possible to appreciate all the graphs combined in the Figure 14 and with the details of the first three meters of the movement in Figure 15. Each red dot is the current odometry of the vehicle, the big blue line is the Dijkstra global plan, and each

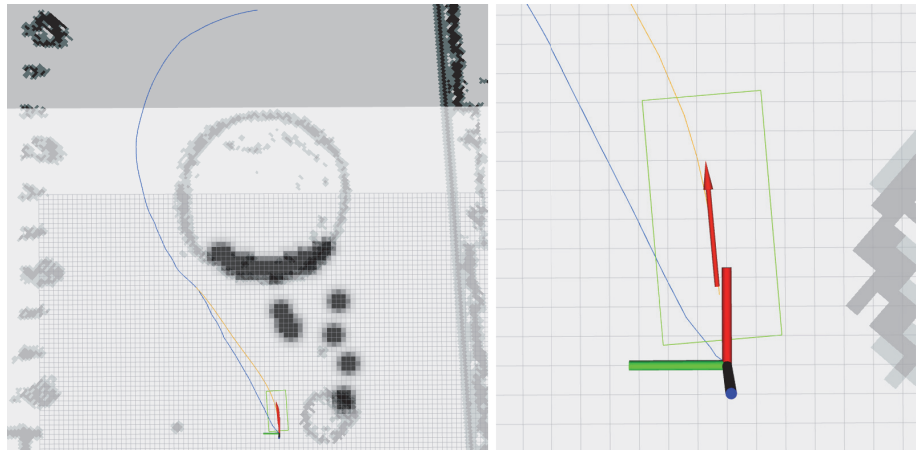


Figure 10: Global plan in blue. Local plan in orange. Overlay of global map and local Costmap 2D.

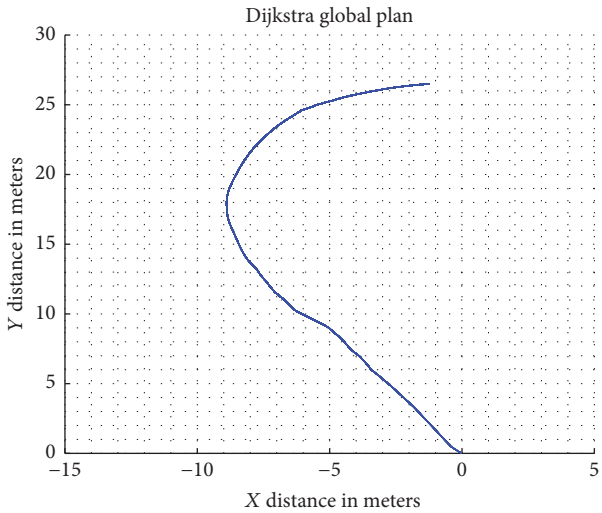


Figure 11: Global path.

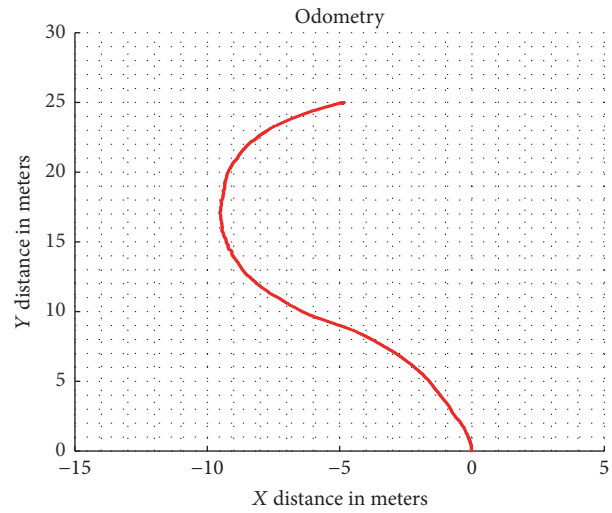


Figure 13: Odometry points.

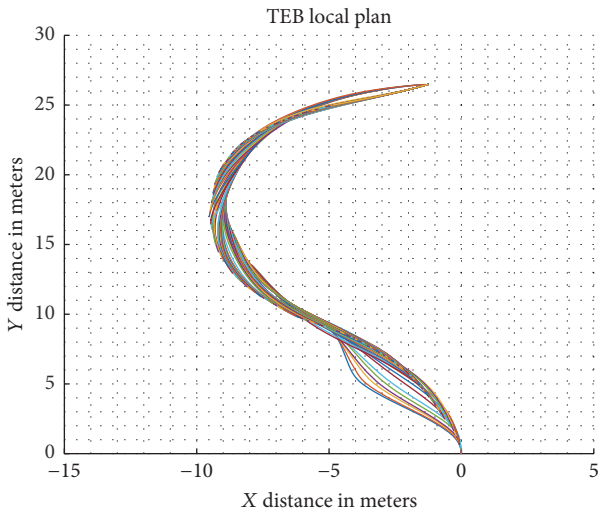


Figure 12: Local path.

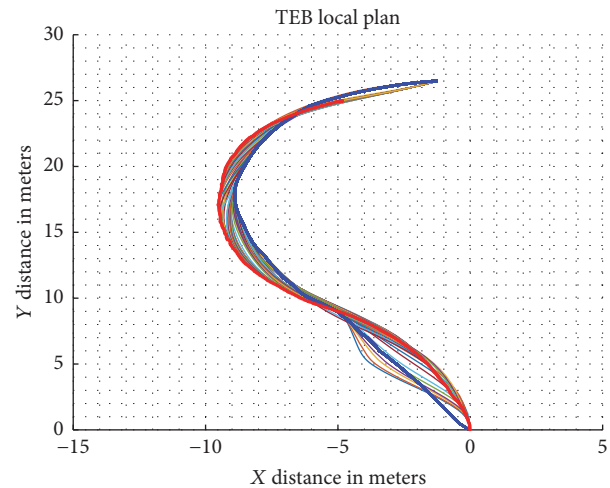


Figure 14: Combined.

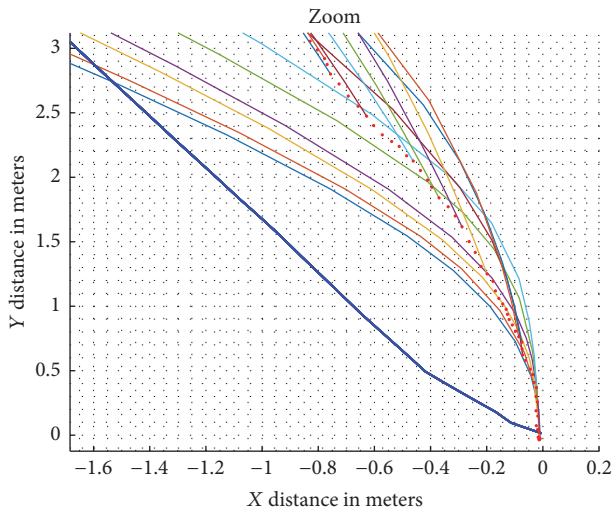


Figure 15: Combined zoom.

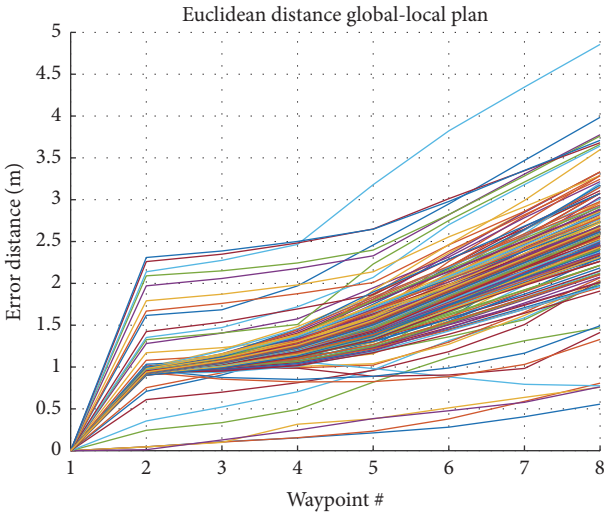


Figure 16: Error local plan, global plan.

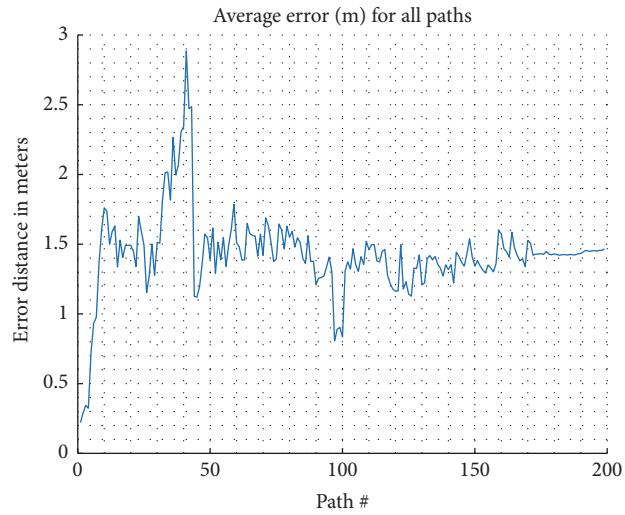


Figure 17: Average error.

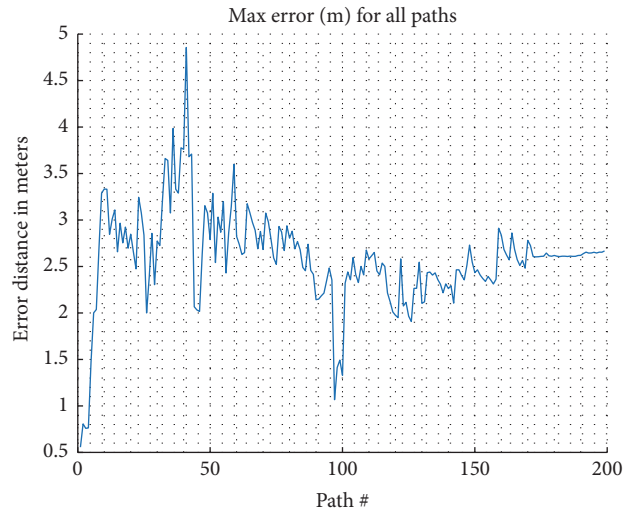


Figure 18: Max error.

thin line in multicolor corresponds to each TEB local plan. It is possible to appreciate the recalculations of the local plan in some dots where the vehicle is not able to follow the proposed trajectory (multicolor lines). Even if the vehicle does not follow perfectly the path due to the mathematical model variations configured into the TEB local planner parameters such as steering angle velocity, it seems that the header of the vehicle is different than in the calculated plan which follows into a wrong steering angle velocity configuration.

Euclidean Distance between Local and Global Path. The error between the global path and the local path determines how accurate the recalculation of the trajectory is. In Figure 16, each line represents the distance error for all local trajectories planned, while the vehicle was moving.

In Figure 16, the Euclidean distance between the first points of the local trajectory and the first points of the global trajectory from the position of the vehicle is described for all paths generated. The reason for this first point is because, after the recalculation of the new local path, the old path is not useful anymore. For the analysis, the seven first points have been selected for each local plan. Regarding this graph, it is possible to notice the incremental growth of the Euclidean distance point by point. The reason for this growth is because the vehicle is not able to follow properly the generated local path. Figure 17 is the result of computing the mean of the seven first values for every path. After the initial rise, there is a tendency around 1.42 m. The maximum Euclidean distance error for all paths generated between local path and global path is detailed in Figure 18. Due to the avoidance of an obstacle at the beginning, the local path deviates further from the global path.

Speed and Steering Angle. For the analysis of the control commands of the speed and steering angle granted by the TEB local planner module, Figure 19 shows the values in

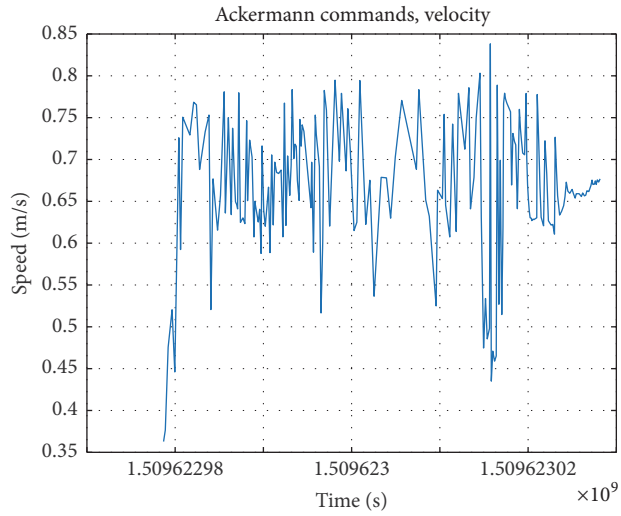


Figure 19: Speed.

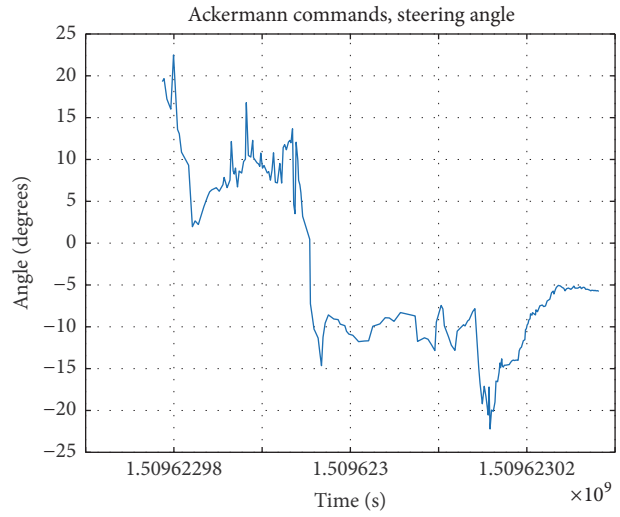


Figure 20: Steering.

real time provided to the vehicle. For the steering angle in Figure 20, it is possible to appreciate the variations of the commands each time the local plan is recalculated. At the beginning of the experiment, the steering angle provided is positive (turn to the left) until the vehicle reaches the position in the world where the circular building is left behind and starts to turn to the right with a negative angle.

6. Conclusions and Future Work

The first conclusion of this work determines that the use of Time Elastic Bands is a good choice even if the model for the kinematics of the vehicle is not precisely adjusted or when the vehicle is out of the path. This algorithm is useful in both of these events and when a dynamic obstacle is in front of the vehicle because of the continuous update of the local path. The second conclusion is that this method has been tested in a real vehicle where the proper configuration of the kinematics, dynamics, and geometry model could differ from the real model. This issue generates trajectories slightly different because the expected movement of the vehicle and the real movement of the vehicle are lightly different. Moreover, the TEB local planner generates the commands of velocities and steering angles that allow the vehicle to reach the goal.

For future work, more experiments are planned such as the comparison between this method and the Dynamic Window Approach and changing the control over the line generated with different path followers such as Pure Pursuit, Optimal Path Controller based on LQR, or Follow the Carrot. More complex scenarios will be added where pedestrians intersect with the trajectory generated and the vehicle should act accordingly, avoiding the pedestrian over the correct side in order to test the aforementioned solution for the elastic bands where a pedestrian is crossing in front of the vehicle.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is supported by Madrid Community Project SEGVAUTO-TRIES (S2013-MIT-2713) and by the Spanish Government CICYT Projects (TRA2013-48314-C3-1-R, TRA2015-63708-R, and TRA2016-78886-C3-1-R).

Supplementary Materials

The supplementary material of this work consists of a video of the real performance of the vehicle in the place where the experiment was done. In the video, it is possible to appreciate the visualizer RVIZ from ROS with two more windows. The top right window displays the left camera of the stereo vision system mounted on the vehicle. Below this window, the laser readings converted into a local map is placed. The background is composed of the global Costmap 2D and the overlay of the local Costmap 2D. On this map, the black squares correspond to obstacles and the color lines describe the trajectories. The blue line represents the global path generated by the Dijkstra algorithm and the orange represents the local path generated by the Time Elastic Bands algorithm. The green square on the bottom of the video represents the dimensions of the vehicle. As soon as the vehicle moves, the red arrows represent the movement and orientation of the vehicle and all of them are superimposed on each other because of the update of the odometry. The vehicle is able to navigate from one point to another without any collision. *(Supplementary Materials)*

References

[1] A. Broggi, M. Buzzoni, S. Debattisti et al., "Extensive tests of autonomous driving technologies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1403–1415, 2013.

[2] J. Ziegler, P. Bender, M. Schreiber et al., "Making bertha drive-an autonomous journey on a historic route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014.

[3] B. Chen and G. Quan, "NP-hard problems of learning from examples," in *Proceedings of the 5th International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 182–186, IEEE, October 2008.

[4] M. Khristamto, A. Praptijanto, and S. Kaleg, "Measuring geometric and kinematic properties to design steering axis to angle turn of the electric golf car," *Energy Procedia*, vol. 68, pp. 463–470, 2015.

[5] J. Canny, "A new algebraic method for robot motion planning and real geometry," in *28th Annual Symposium on Foundations of Computer Science*, pp. 39–48, IEEE, 1987.

[6] P. Bhattacharya and M. L. Gavrilova, "Voronoi diagram in optimal path planning," in *Proceedings of the 4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD '07)*, pp. 38–47, July 2007.

[7] S. Skiena, "Dijkstra's algorithm," in *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*, pp. 225–227, Addison-Wesley, Boston, Mass, USA, 1990.

[8] R. Dechter and J. Pearl, "Generalized best-first search strategies and the optimality of A^*," *Journal of the ACM (JACM)*, vol. 32, no. 3, pp. 505–536, 1985.

[9] D. Gelperin, "On the optimality of A^*," *Artificial Intelligence*, vol. 8, no. 1, pp. 69–76, 1977.

[10] M. Šeda, "Roadmap methods vs. cell decomposition in robot motion planning," in *Proceedings of the 6th WSEAS International Conference on Signal Processing, Robotics and Automation*, pp. 127–132, World Scientific and Engineering Academy and Society (WSEAS), 2007.

[11] P. Vadakkepat, K. C. Tan, and W. Ming-Liang, "Evolutionary artificial potential fields and their application in real time robot path planning," in *Proceedings of the Congress on Evolutionary Computation (CEC 00)*, vol. 1, pp. 256–263, IEEE, July 2000.

[12] S. M. LaValle, *Rapidly-exploring random trees: A new tool for path planning*, 1998.

[13] S. X. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 1, pp. 718–725, 2004.

[14] M. Ben-Ari and F. Mondada, "Mapping-based navigation," in *Elements of Robotics*, pp. 165–178, Springer, 2018.

[15] C. Rosmann, W. Feiten, T. Wosch, F. Hoffmann, and T. Bertram, "Efficient trajectory optimization using a sparse model," in *Proceedings of the 2013 6th European Conference on Mobile Robots, ECMR 2013*, pp. 138–143, IEEE, September 2013.

[16] S. Gim, L. Adouane, S. Lee, and J.-P. Dérutin, "Clothoids composition method for smooth path generation of car-like vehicle navigation," *Journal of Intelligent & Robotic Systems*, pp. 1–18, 2017.

[17] J. P. Rastelli, R. Lattarulo, and F. Nashashibi, "Dynamic trajectory generation using continuous-curvature algorithms for door to door assistance vehicles," in *Proceedings of the 25th IEEE Intelligent Vehicles Symposium, IV 2014*, pp. 510–515, IEEE, June 2014.

[18] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.

[19] A. Piazzi, C. G. Lo Bianco, M. Bertozzi, A. Fascioli, and A. Broggi, "Quintic G_2-splines for the iterative steering of vision-based autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 27–36, 2002.

[20] C. Rösmann, W. Feiten, T. Wosch, F. Hoffmann, and T. Bertram, "Trajectory modification considering dynamic constraints of autonomous robots," in *Proceedings of the 7th German Conference on Robotics; ROBOTIK 2012*, pp. 1–6, VDE, 2012.

[21] P. Marin-Plaza, J. Beltran, A. Hussein et al., "Stereo vision-based local occupancy grid map for autonomous navigation in ros," in *Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, vol. 3, pp. 703–708, 2016.

[22] R. Moliner and R. Tanda, "Tool for robust tuning of PI/PID controllers with two degree of freedom," *RIAI: Revista Iberoamericana de Automatica e Informatica Industrial*, vol. 13, no. 1, pp. 22–31, 2016.

[23] R. Arnanz, F. J. García, and L. J. Miguel, "Methods for induction motor control: State of art," *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial*, vol. 13, no. 4, pp. 381–392, 2016.

[24] A. Kokuti, A. Hussein, P. Marin-Plaza, A. De La Escalera, and F. Garcia, "V2X communications architecture for off-road autonomous vehicles," in *Proceedings of the 2017 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pp. 69–74, IEEE, Vienna, Austria, June 2017.

[25] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Robotics: Science and Systems*, vol. 2, 2014.

Hindawi
Journal of Advanced Transportation
Volume 2018, Article ID 9702730, 11 pages
https://doi.org/10.1155/2018/9702730

Research Article

Investigations on Driver Unique Identification from Smartphone's GPS Data Alone

Arijit Chowdhury,[1] **Tapas Chakravarty**,[1] **Avik Ghose**,[1] **Tanushree Banerjee**,[1] **and P. Balamuralidhar**[2]

[1] *Tata Consultancy Services, Building 1B, Ecospace, Plot IIF/12, New Town, Rajarhat, Kolkata, West Bengal 700156, India*
[2] *Tata Consultancy Services, Whitefield, Bangalore, India*

Correspondence should be addressed to Tapas Chakravarty; tapas.ieee@gmail.com

Received 28 August 2017; Revised 27 December 2017; Accepted 15 January 2018; Published 8 February 2018

Academic Editor: David Martín

Driver identification is an emerging area of interest in vehicle telematics, automobile control, and insurance. Recent body of works indicates that it may be possible to uniquely identify a driver using multiple dedicated sensors. In this paper, we present an approach for driver identification using smartphone GPS data alone. For our experiments, we collected data from 38 drivers for two months. We quantified the driver's natural style by extracting a set of 137 statistical features from data generated for each completed trip. The analysis shows that, for the "driver identification" problem, an average accuracy of 82.3% is achieved for driver groups of 4-5 drivers. This is comparable to the state of the arts where mostly a multisensor approach has been taken. Further, it is shown that certain behavioral attributes like high driving skill impact identification accuracy. We observe that Random Forest classifier offers the best results. These results have great implications for various stakeholders since the proposed method can identify a driver based on his/her naturalistic driving style which is quantified in terms of statistical parameters extracted from only GPS data.

1. Introduction

Studies on human behavior are gaining importance and a large number of theories are originating on this topic [1]. The area of computational social science is emerging as an important field of study. "Driver profiling" is one such aspect contributing to human behavior models. In recent times, a number of studies have been carried out to extract meaningful and actionable insights from driving data. The primary objective of these studies are to derive an estimate of risk involved in the very nature of driving a land vehicle. Wahlström et al. [2] provide us with a very detailed review of the work done (smartphone based telematics) in the last ten-year span. A large body of the works referred by Wahlström et al. relates to the development of industrial applications like UBI (Usage Based Insurance) and connected car scenarios. However, a larger question remains to be answered: Will the estimation of driving risk be sufficient to model the driving behavior? In an earlier work, Fuller [3] proposed that the drivers try to estimate and maintain a level of task difficulty. Fuller [4] had proposed a TCI (Tasks Capability Interface) model in this regard. This model proposes that the driver is continuously reacting to the demand of the driving task and his/her assessment of own capability. Considering that Fuller's "driver model" is comprehensive, we are then led to believe that each individual will demonstrate uniqueness in driving pattern leading to the possibility of fingerprinting.

Keeping a focus on driver's operative characterization, Lin et al. [5] evaluate different methods such as evaluation of driver's real-time behavior, vehicle state, or monitoring facial expressions. Lin et al. also mentioned that such behavior needs to be classified, before identification. However, the issues related to driver fingerprinting were addressed by Enev et al. [6]. In their paper aptly titled "Automobile Driver Fingerprinting," Enev et al. investigate "the potential to identify individuals" by analyzing their natural driving styles. Here [6], authors demonstrate that, even with a restricted set of sensors, the drivers can be uniquely identified with

an accuracy of 87% (99% with top 5 sensors) using just 15 minutes of on-road data. This is a significant advancement in comparison to the work done by Zhang et al. [7] who utilized simulated data [multisession, 20 male drivers] and demonstrated 85% accuracy by using Hidden Markov Model (HMM). Enev et al. also hypothesized that given an availability of enough longitudinal data, "everyone can be distinguished." In another related work on driver identification, Hallac et al. [8] demonstrated that there are unique patterns in individual driving styles which can be detected even for a short drive. Hallac et al. experimentally demonstrated that the vehicle turn signature is often well suited for detecting individual style. In this case, authors designed an experiment consisting of driving a car (several drivers) along a road segment of 150-foot radius. They monitored 12 sensor readings, other than GPS (Global Positioning System). The result shows an average prediction accuracy of 76.9% for two-driver classification and 50.1% for five-driver classification. Authors also mentioned that a fusion approach of prediction models can lead to an enhanced classifier. In a 2015 survey article, Engelbrecht et al. [9] have presented valuable insight in driving behavior analysis. Engelbrecht et al. draw our attention to the fact that smartphone based sensing links the behavior to an individual rather than a vehicle. Further, Engelbrecht et al. note that there are differences between systems which detect driving maneuvers only and those which classify driver behavior. In order to predict future state of maneuvers, it is first necessary to deduce a person's naturalistic driving style. Only then an anomalous behavior can be accurately recognized. In an earlier work, Themann et al. [10] noted that there is a strong need to identify driver's unique preferences and incorporate such models in adaptive cruise control systems (anticipating driving style) so that a largely automated driving can fulfill its promise of improving fuel efficiency.

It is to be noted that there are different objectives for modeling specific driving maneuvers vis-à-vis modeling general driving behavior. The risky sudden maneuvers like hard brake happen at an operational level, in a timeframe of millisec, whereas lane change, turn, and stop are more tactical, on a timescale of seconds. The long term goals of the driver have longer time frames and are strategic in nature. Abuali and Abou-Zeid [11] illustrate the above stated viewpoint with reference to the proposition by Laapotti et al. [12] that there exists a behavioral level on top of the above three levels of hierarchical control model; this layer describes the life skill and general goal of the individual. Thus taking a cue from the above statement, we may predict that identifying a driver's propensities can lead us to understand the person's exercise of choice in different scenarios, whenever she/he is faced with self-assessment of task difficulty and own capability.

Smartphone continues to be a favorite platform for sensing a driver's real-time maneuvers and deriving driving profile. Vlahogianni and Barmpounakis [13] investigate smartphone based analytics for driving behavior assessment. As mentioned by Vlahogianni and Barmpounakis, there is little knowledge about reliability of such smartphone based sensing except for the fact that GPS (in a phone) and accelerometer continue to remain a popular choice. In a recent work Tanprasert et al. have proposed driver identification in real-time using accelerometer and GPS data using unsupervised anomaly detection and neural networks [14]. Acceleration variation is also investigated for the purpose of driver identification in [15] by Phumphuang et al. They also used PCA (Principle Component Analysis) for dimensionality reduction. In another recent work, Junior et al. [16] investigate driver behavior profiling using different Android phone based sensors and different types of machine learning algorithms. In this case, 4 car types and 13 minutes of average drive time are investigated with primary sensors being accelerometer, magnetometer, and gyroscope. In conclusion, authors in [16] mention that Random Forest algorithm "is the best performing MLA with 28 out of 35 best assemblies."

Present authors have worked extensively to improve the reliability of smartphone based GPS measurements [17, 18]. In a recent work [19], we have presented a driver behavior analysis platform where nearly 50000 Kms worth of driving data, obtained from 38 drivers (in their natural environment), is analyzed and statistically modeled with 2D factor analysis, with the factors being aggression and skill. In this work, we utilize the same dataset and investigate much granularity to detect variability in natural driving styles of the individual. We have considered those GPS data as valid where the horizontal accuracy measure is reported to be less than 16 m. A few incomplete datasets are removed for analysis purpose. While majority of collected GPS data (speed, heading, etc.) obtained are found to be quite accurate, we have also applied filters on raw GPS speed measurements (as outlined in [17, 18]) in order to eliminate spurious values. It is often seen that sudden brakes and acceleration lead to momentary loss of data which can lead to wrong estimation of acceleration/deceleration at that instance. However, such instances are few and these have been corrected. It is also felt that a few anomalous events will not affect our proposed aggregate statistical model. In this aggregate model, driver behavior is quantified based on the statistics associated with the completed trip rather than individual events. Also, we have not performed behavior identification with considerations to different road segments separately. Even though we have considered lateral acceleration (derived from GPS data) as an important indicator of driver behavior, our proposed model does not delve deep into specific horizontal curves. Authors in [20] have shown that, along horizontal curves, the perception of the road geometry by drivers affects road safety. Vaiana et al. [20] conducted experiments with 35 participants driving around overall 86 curves with different radii and GPS data was analyzed to understand the driving behavior around the curves. Thus, it is felt that future works may specifically investigate this aspect.

The key contributions of this paper are as follows.

We investigate whether unique identification of driver is possible by using only GPS data measurements. This entails an investigation into the level of accuracy achievable under such circumstances. Our objective is to identify the driver's natural style and distinguish drivers based on their naturalistic driving styles. Towards that goal, feature computation and classification have been performed. Drivers are grouped in groups of 4-5 drivers, based on the proximity

Table 1: Basic statistics and travelled distance for a representative sample of drivers.

Driver ID	Evening trip	Day trip	Morning trip	Distance travelled (Km)	Average speed (m/s)	Total trip duration (minutes)	Total number of trips
D001	121	26	49	3632	11.85	4638	241
D002	20	5	10	365	8.76	902	37
D003	26	1	28	719	8.92	1738	58
D004	30	7	5	884	12.50	1227	48
D005	41	13	26	1057	10.30	2082	103

of regular driving locations as well as trip timings. Such a method is assumed to normalize the environment in which the drivers operate. Then driver identification is studied using different algorithms and accuracy is validated using *k-fold cross-validation*. The results are reported on test sets for different groups. The presented approach enables us to map a journey with the most likely driver taking only GPS data into consideration.

The rest of the paper is organized as follows: in Section 2, we describe our experiment details and the data set generated. Section 3 describes our algorithms and methods applied. In Section 4 we illustrate our results and provide analysis of the same. Finally, we summarize our conclusions in Section 5 of the paper.

2. Experimental Setup and Data Collection

2.1. Experimental Setup. For this study, we collected data from a total of 38 drivers. Majority of the drivers recorded at least 2 trips on weekdays within an observation window of two months. It is to be noted that the authors of this paper played no role in selecting drivers as well as the driving location. However, the subjects form a peer group based on geography and it is assumed that they went about their normal lifestyles, usually travelling from home to work and back, with one or two additional trips. Overall, we have around 4000 individual trips. Our method relies solely on data gathered from smartphone's GPS modules without any knowledge of local road and traffic conditions. Overall a total of 3927 trips are considered from the collected data. Data were collected on weekdays with one or more trips per day. In total, 1233 hours of driving data is captured and the overall car journey length is found to be 50740 Km, covering various regions in the USA. Although the driver's geographical location is available from latitude and longitude, we do not have access to any personally identifiable data such as name, age, gender, car type, and home address for privacy protection. Many users enrolled in this program and were given sequential IDs (D001–D085); we only conducted our study based on 38 drivers who had considerable number of complete trips. As can be observed, the trips are found to be well distributed across different times of the day. A representative sample is presented in Table 1.

Taking long distance trips and short distance trips separated across time in a day creates diversity in runtime condition of the vehicle and thus enriches the data set. Driver D055 was the least frequent traveller with overall 12 trips only. That data is utilized to verify the effectiveness of our algorithm when applied on a relatively low data volume. Although a high identification accuracy for D055 is not expected, we explored the possibility that inclusion of a lower volume data set does not affect the overall result.

2.2. Data Collection and Preprocessing. As mentioned, our primary data collection module is a GPS receiver. The participant driver downloads an app for GPS logging on her smartphone. Each participant is instructed to drive naturally according to his/her preference. Data is collected only when that participant is driving his/her own car (a single vehicle per participant). Smartphone is kept inside car and held fixed with respect to car's body. Since only GPS reading is used, the orientation of smartphone is not important for our work.

In order to evaluate data quality, the knowledge of measurement Precision is of utmost importance [25]. Such Precision is dependent on the quality of GPS receiver of the given smartphone. The integrity and availability of smartphone's GPS signal get affected by events like urban tunneling, sudden change in vehicle speed and direction, and so forth. Handel et al. [25] discuss GPS data integrity enhancement and monitoring in detail. Authors demonstrate that, in order to ensure robust calculation of different figures of merit, "data cleansing and integrity monitoring are much needed." Handel suggests two techniques, namely, second-by-second data and whole trip data. It is stated that a direct differentiation of the GPS speed data will amplify high frequency noises and outliers. A better option will be to clean the speed measurements, by fitting a polynomial model.

For our purpose, we have followed the principle outlined by Handel et al. [25]. To begin with, we eliminate all measurements which offer horizontal accuracy metric of greater than 16 m. Further, we attempt to estimate the true speed (at a given instance) by building a relationship with two immediate past measurements. The method is similar to moving average filter except that the averaging is done with different weight coefficients. The method is illustrated in detail in [17] and is not repeated here. Moreover, it is observed that there are only few missing data points. Since we use aggregate model where each completed trip is taken as a set of data values, an outlier detection algorithm eliminates those which display marked deviation in the statistical properties associated with both lateral and longitudinal acceleration. This aspect is explained in detail in [19]. It is to be noted that the proposed model

is quite unlike an event based method. In the present case, a handful of anomalous events are treated as outliers so that we can deduce the natural driving style of the driver.

Thus, the data recorded consists of attributes like timestamp, altitude, course, horizontal accuracy, latitude, longitude, and speed in m/s. All data are collected at 1 Hz rate. The data consists of the following parameters.

(i) Driver ID: a unique ID that distinguishes a driver.

(ii) GPS data: all attributes like speed, location, course, and horizontal accuracy.

Let us assume that v_1, v_2 to v_n be the consecutive speed measurement samples, at time t_1, t_2 to t_n. Similarly, we get θ_1, θ_2 to θ_n as consecutive course (heading) measurements. Let us also assume that a_1, a_2 to a_n be the consecutive discrete acceleration samples (derived from speed measurements) at time t_1, t_2 to t_n, where $\Delta t = t_n - t_{n-1}$ for uniform sampling rate. Then, from these primary GPS measurements, few secondary data are computed. These are as follows:

(i) Longitudinal acceleration:

$$\text{acc}_{\text{Long}} = \frac{(v_n - v_{n-1})}{(t_n - t_{n-1})}. \tag{1}$$

(ii) Angular speed: rate of change of course:

$$\omega = \frac{\partial\theta}{\partial t} * \frac{\pi}{180}. \tag{2}$$

(iii) Lateral acceleration [26]:

$$\text{acc}_{\text{Lat}} = \frac{v^2}{R} = v\omega. \tag{3}$$

(iv) Jerk: rate of change of acceleration (m/s^3):

$$J_i = \frac{(a_{i+1} - a_i)}{\Delta t} \quad \forall 1 \le i \le 3. \tag{4}$$

(v) Jerk energy [27] for the sth window (using (4)):

$$\text{JE}_s = J_{s1}^2 + J_{s2}^2 + J_{s3}^2, \tag{5}$$

Here, "jerk energy (JE)" is computed with 50% overlap and sliding window (4 seconds) based method. In addition, we compute 1st and 2nd derivative with respect to time for speed, acceleration (both longitudinal and lateral), jerk, jerk energy, and angular speed. From the above list, all redundant computations are removed. Thus, the primary measured values and the secondary computed values constitute the trip level data for further classification. All these data are denoted as $d \in D$; where D is the set of all data.

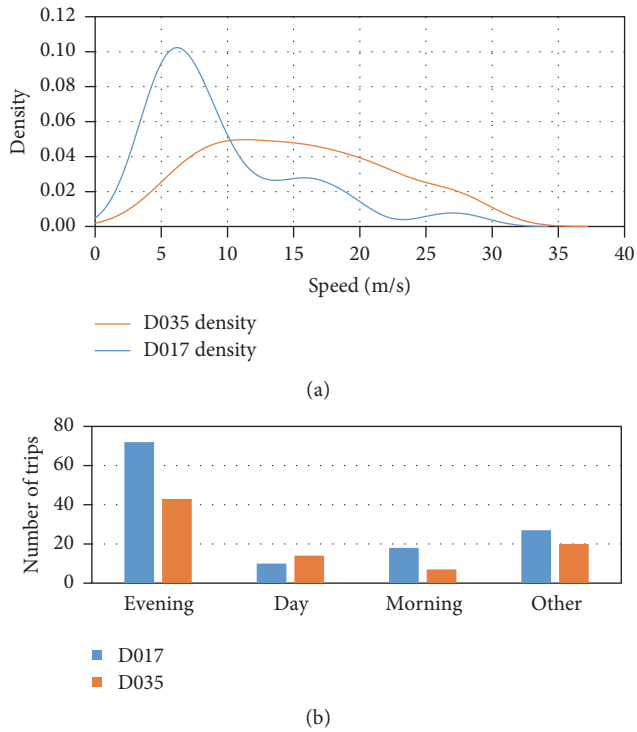


Figure 1: *Distribution of important parameters of journey for 2 drivers.* (a) Distribution of average speed. (b) Trips occurring at various time of day.

3. Method of Analysis

3.1. Basic Analysis and Feature Extraction. It is observed that the overall dataset is quite diverse, accommodating scenarios like driving on weekdays (or weekends), different time slots of the day, and different localities. Basic statistics derived from the driving data shows variation across drivers. Figure 1(a) shows the variation in the average speed (for each completed trip) for two drivers, namely, D017 and D035. For each driver, multiple trips are available and from every trip the average speed value is computed. Thus for each driver a set of average speeds (i.e., average speed per trip) is obtained. From this set, an empirical distribution of average speed can be computed. Also D017 and D035 have trips spread throughout different times of a day, as evident from Figure 1(b).

We have explored the dataset to obtain basic distributions of the total trip duration as well as the total distance covered by the participant drivers. This is shown in Figure 2. It can be seen that the typical distance covered by a driver is less than 1000 Km while typical driving time is approximately 20 hours. In order to statistically explore the dataset, we extracted multiple features from the driving data. We consider GPS measured "Speed" and "Heading" as primary data. From "Speed", we compute secondary data, namely, jerk, jerk energy, lateral acceleration, angular speed, and longitudinal acceleration. The positive and negative acceleration (both longitudinal and lateral) are segregated and separately treated. Next, the 1st and 2nd derivative (with respect to time) for all the above stated data are calculated; these also form secondary data. It is to

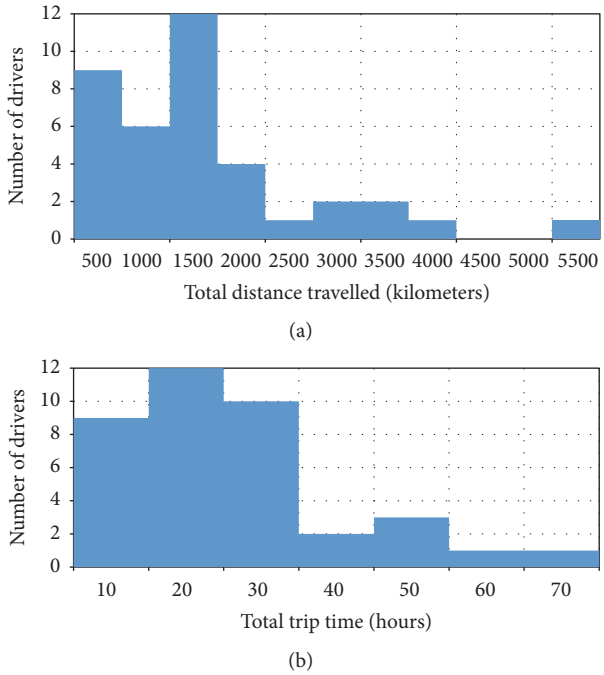


FIGURE 2: *Distribution of total journey distance and time for the participants*. (a) Histogram of total distance travelled by the drivers. (b) Histogram of total trip duration for the drivers.

be noted that, for "Heading," we take only the derivatives as secondary data. Subsequently the statistical features of the primary and secondary data, corresponding to every completed trip, are extracted. These features are mean, median, skewness, kurtosis, standard deviation, max, min, 97.5th percentile, 1st and 3rd quartiles (Q1, Q3), IQR (interquartile range: Q3–Q1), and 2.5th percentile. Table 2 summarizes the list of features. Thus, the variations in speed and acceleration are taken to be the major indicators of how the car is driven.

Some examples of features are median of speed, Q1 of 1st derivative of lateral acceleration, kurtosis of 2nd derivative of jerk energy, and 2.5th percentile of 2nd derivative of jerk energy. Our choice of suitability of the "feature" rests on the requirement that it should have good individual predictive ability. Also, the set of selected features should have low correlation amongst themselves and high correlation with class (driver in our case) [28]. As the next step, "analysis of variance" (ANOVA) is performed on the entire feature set in order to identify the statistically invariant features across all the drivers. For example, skewness of acceleration is seen to be very less close to 0 and does not vary with trip or driver. These are omitted. After such omission, we obtain a set of 137 features that is considered suitable for further analysis. From here onwards, these 137 features constitute a "set of all features" hitherto named as "global" feature set. It is to be noted that the proposed approach is dependent on the completion of the trip thereby excluding those applications where real-time identification of the driver is required. That is why dimensionality reduction methods like PCA are not considered. Also, the car type for each user is not known except for the fact that these are owned by the participant drivers. Hence, ANOVA cannot be performed with respect to car type. Throughout this study, we assume that the observed variations are due to differences in driving styles only. However, if the knowledge on the dynamics of the car as well as the road condition was available, it could have enhanced the quality of interpretation.

3.2. Classification for Driver Identification. Driver identification is approached as a classification problem, where each driver represents a class. From the trip data corresponding to a driver (belonging to a group of 4-5 drivers), the features are computed and classification is performed. Here, the classification accuracy with respect to a driver is measured by the percentage of correctly classified trips for that driver in a group. Similarly, the group level accuracy is measured by the percentage of correctly classified trips corresponding to that group. A driver misclassification reduces the overall accuracy.

To check the applicability of the global feature set, we performed Random Forest [29] based evaluation and observed that the accuracy obtained is good for the purpose of behavior identification. Throughout this study, for the purpose of classification, we have used only Random Forest with 100 trees and a batch size of 100. Maximum depth of the tree is set to unlimited. Table 3 gives the confusion matrix for a group of four drivers (a group consisting of drivers D001, D002, D006, and D009). Metrics like Precision, Recall, and $F1$ score are used for measuring performance of classifiers [30].

Overall accuracy obtained for this group is 77.7%, where accuracy is defined as the ratio of total number of correctly identified trips over the total number of trips. Within a group, the identification accuracy of individual driver is denoted by the computed value of *Recall*, percentage of his/her own trip identified correctly. Additionally, we compute *Precision* which is defined as how many trips (ratio), identified as belonging to a particular driver, actually belongs to him or her. For the purpose of multiclass classification, we computed the macro average (averaging the evaluation measures) performance measure [31]. Macro averages of Precision, Recall, and $F1$ score are 0.79, 0.76, and 0.78, respectively. We also observed that the other techniques like KNN (k-Nearest Neighbor) and SVM (Support Vector Machine) show lower accuracy as compared to Random Forest. {SVM is used with polynomial kernel and tolerance parameter is set to 0.001, iterations continue till convergence happens while KNN is used with number of nearest neighbors set to 8 and linear search} The models are then validated using k-fold cross-validation technique, where $k = 10$ is used. There is no algorithm that performs better in general; thus there is a need to choose an appropriate algorithm that outperforms others for the proposed problem definition. Random Forest outperformed SVM and KNN each time in our study. Similar observations have been identified by other authors [8, 16]. In [8], Hallac et al. mention that as the driver pool increases, the alternative approaches using Multinomial Logistic Regression and Support Vector Machines "dropped off significantly" as compared to Random Forest. In [16], Junior et al. compared Artificial Neural Networks, Support Vector Machines, Random Forest, and Bayesian Network towards driver behavior profiling. Junior et al. conclude

Table 2: Summary of features for driver identification.

Data	Feature
Speed, jerk, jerk energy, lateral acceleration, angular speed, and longitudinal acceleration	Mean, median, skewness, kurtosis, standard deviation, max, min, 97.5th percentile, 1st and 3rd quartiles (Q1, Q3), IQR (interquartile range: Q3–Q1), and 2.5th percentile
Positive lateral acceleration, negative longitudinal acceleration, positive lateral acceleration, and negative longitudinal acceleration	
All 1st and 2nd derivatives of data in rows 1 and 2	

Table 3: Accuracy and confusion matrix on a group (G4) of drivers using global feature set.

Driver ID	Predicted				Classification accuracy (Recall%)
	D001	D002	D006	D009	
Actual					
D001	31	1	7	0	79.5
D002	1	15	0	10	57.7
D006	11	0	37	0	77.1
D009	1	1	1	32	91.4
Precision (%)	70.5	88.2	82.2	76.2	

that Random Forest outperforms others for majority of assemblies.

In our application, the final accuracy is validated on "test data" (data from new journey of the drivers). It is found that the accuracy remains similar to what was obtained in k-fold cross-validation. Hence, for the rest of this paper, we present our evaluation using the global feature set together with Random Forest.

4. Results and Analysis

4.1. Driver Identification Results. For the purpose of driver identification, we segregated the drivers into different natural groups and analyzed the trips for each group. While grouping the drivers, we fundamentally looked at their regions of operation as well as the trip timings. Mostly, the drivers are grouped according to geographical similarity. They drive in the same area of a city with comparable trip timings; that is, the drivers operating in similar locality with similar departure time are grouped together. Figure 3 displays a map outlining the traversed regions for some of the groups. It needs to be mentioned that a few drivers fall in multiple groups as they have overlapping route locations and/or timings.

The blue marks in Figure 3 show the traversed routes by the participants. The red colored boundaries denote the zone covered by all the drivers belonging to the specific group. The underlying premise is that all the drivers in one group will face very similar external perturbations that may affect their driving styles. Therefore, the distinctiveness in their driving data can be attributed to their own propensities only.

In this section, we present the results obtained for each group of drivers as well as the accuracy obtained for individual drivers. These results are summarized in Table 4.

The detailed results for G4—the group consisting of drivers D001, D002, D006, and D009—have been presented in Table 3. Across all the groups, G1 to G9, we obtain an average overall accuracy of 82.3%. This in turn confirms the strength of the selected feature set as well as the classification

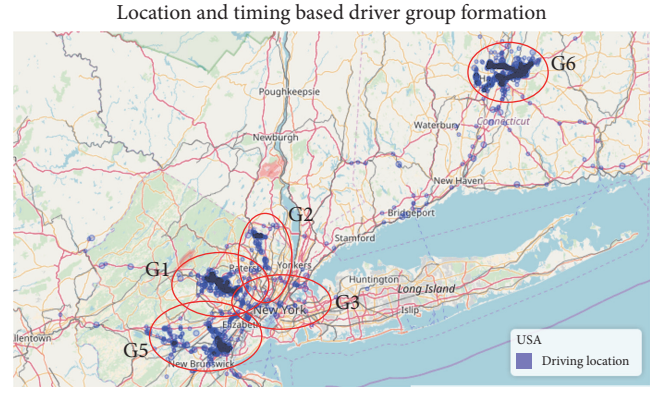


Figure 3: Map displaying geographical locations for the group of drivers [typical journey localities for each group of drivers are circled].

method chosen for the purpose of identification. At group level, the least average accuracy obtained is 74%, being that for group 9. The average accuracy in group 9 is affected by one driver, D055. For driver D055, none of his trips are classified correctly (total 9 trips are considered). This happened due to the limited availability of training data. Although the group level average accuracy is deemed to be high (>78% except for G9); the same cannot be unequivocally stated for individual driver identification. Within a group, both Recall and Precision are observed to vary, with the minimum value being 0.25 (for G9). However, for all such cases, the available training data is much smaller than others.

It is seen that the proposed method works quite well when a group of limited size is formed. Cross-validation results for a sample group are shown in Tables 5 and 6 for 10-fold cross-validation and tested on new data.

We have placed D021 in three groups (G1, G3, and G5) while D014 is placed in two groups (G1 and G2). This is because the routes traversed by the drivers in these groups lie in reasonable proximity and the routes undertaken by D021

Table 4: Summary of driver identification scores for all groups.

Group	Precision	Recall	$F1$ score	True negative rate	Overall accuracy
G1	0.81	0.79	0.8	0.93	0.82
G2	0.86	0.74	0.79	0.95	0.84
G3	0.80	0.79	0.79	0.93	0.8
G4	0.79	0.76	0.77	0.94	0.78
G5	0.81	0.78	0.79	0.94	0.79
G6	0.86	0.81	0.83	0.93	0.83
G7	0.86	0.90	0.88	0.97	0.9
G8	0.9	0.89	0.89	0.97	0.91
G9	0.45	0.63	0.87	0.89	0.74

Table 5: Accuracy and confusion matrix on group 6 (G6) of drivers with all features.

Driver ID	Predicted				Recall (%)
	D0068	D0069	D0064	D0066	
Actual					
D0068	38	2	1	0	92.6
D0069	4	22	1	0	81.4
D0064	3	4	13	0	65
D0066	1	1	0	13	86.6
Precision (%)	82.6	75.8	86.6	100	

Table 6: Accuracy and confusion matrix on group 6 (G6) of drivers for 10-fold cross-validation.

Driver ID	Predicted				Recall (%)
	D0068	D0069	D0064	D0066	
Actual					
D0068	42	0	22	8	58.3
D0069	0	42	2	0	95.4
D0064	8	0	118	18	81.9
D0066	3	0	25	70	71.4
Precision (%)	79.2	100	70.6	72.9	

and D014 fall in any one of these overlapping groups. However, for the purpose of identification, we have considered all the trips undertaken by the said drivers (like 14 trips for D021 and 11 trips for D014) in each of their overlapping groups; else a group wise route segregation would have resulted in a substantially reduced data for the given drivers in these groups.

It is also of interest to evaluate the identification accuracy when all the 38 drivers are placed in a single group. Towards this objective, we apply Random Forest on the entire dataset. It is seen that 25 drivers displayed accuracy better than 40% with 16 of them crossing 50% accuracy level. The median accuracy obtained is 0.46 with standard deviation being 0.22. Figure 4 represents the confusion matrix in terms of a heatmap which shows how many (percentage) trips of a driver are correctly classified and how many of them are mapped into other drivers.

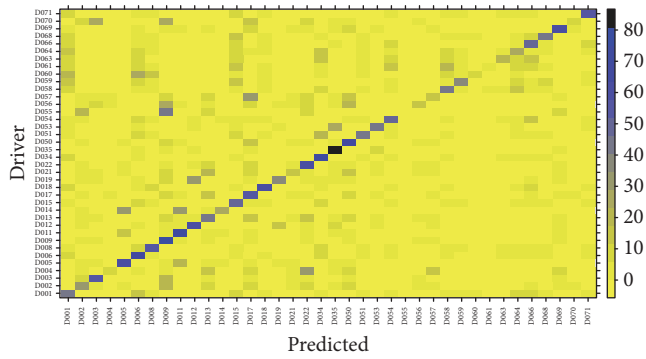


Figure 4: Accuracy and classification obtained for all 38 drivers taken together in a single group.

4.2. Effect of Skill and Aggression on Driver Identification Accuracy. In our previous work [19], we have quantified each driver in terms of 2D factor analysis—Aggression and Skill. High acceleration/deceleration maneuvers induce risk in driving. These patterns get reflected as heavy tail of the acceleration distribution obtained from trip level data. Since kurtosis is a good measure of sharp peak and heavy tail, it is chosen as a suitable feature to be extracted from the acceleration profile. Also, the combination of kurtosis and skewness squared can be used as an identifier of underlying distribution. For a skilled driver, the basic nature of acceleration will not significantly vary between trips. In our study, it is found that skewness of acceleration profile is close to 0 (pdf of acceleration is symmetric). Therefore, only the variation in kurtosis over all trips of a driver indicates variation in underlying distribution of acceleration. The inverse of variation in kurtosis is quantified as a measure of skill and the median value of kurtosis is used as an indicator of aggression [19]. Similarly, we consider the median and standard deviation corresponding to lateral acceleration. Then for a group of drivers, these measures are normalized to obtain skill and aggression score. Finally, we form five clusters of drivers with 50% (19) of participants treated as "normal" drivers. From this analysis, we get 7 drivers in the cluster denoted as "high skill and low aggression" and 8 drivers in "low skill and high aggression" cluster. It is be noted that such clustering is only relative; if the population changes, some drivers may relocate to adjacent clusters.

Each driver, based on all the available trips, is assigned a point in two-dimensional space as aggression and skill. We now investigate whether relative skill and/or aggression can affect identification accuracy. It is to be noted that the skill-aggression scores for each driver are derived from the same GPS data which is used for identification. We have taken a snapshot of drivers for analysis. In Table 7, we display skill-aggression score [19] for all 38 drivers. Please note that Table 7 is sorted according to skill score, from highest relative skill to lowest, with the scores being bounded to ±4: skill score of +4 is termed as most skillful (or most aggressive) and

Table 7: Aggression and skill score for drivers.

Driver	Aggression	Skill	Associated group
D018	−1.06	3.87	3
D053	−1.62	1.76	5
D066	−0.98	1.65	6
D071	−1.6	1.58	9
D058	−0.41	1.39	1
D019	0.46	0.9	3
D008	−1.06	0.76	7, 8
D054	−0.53	0.41	7, 8
D068	−0.85	0.35	6
D035	−0.94	0.31	Ungrouped
D012	−0.09	0.23	2
D055	0.33	0.21	9
D006	−0.61	0.1	4
D015	−0.4	0.02	Ungrouped
D059	−1.11	−0.05	8
D063	−0.49	−0.15	9
D064	−0.43	−0.19	6
D069	−0.31	−0.2	6
D003	−0.24	−0.22	Ungrouped
D060	−0.25	−0.28	Ungrouped
D009	−0.05	−0.32	4
D001	−0.82	−0.37	4
D013	0.35	−0.44	2
D057	1.25	−0.47	Ungrouped
D002	0.73	−0.52	4
D034	0.64	−0.55	1
D017	0.61	−0.59	3
D070	3.18	−0.6	7, 9
D056	0.87	−0.69	5
D061	−0.91	−0.69	8
D011	0.46	−0.77	Ungrouped
D022	0.83	−0.82	2
D004	0.92	−0.84	7
D005	1.4	−0.85	Ungrouped
D021	0.22	−0.94	1, 3, 5
D014	2.01	−0.97	1, 2
D050	0.64	−0.97	5
D051	−0.09	−1.03	5

−4 depicting the least skill (or minimum aggression). These scores are so normalized that an average driver will get a score of (0, 0).

We take the case of four drivers, namely, D058, D018, D066, and D071, who display high relative skills and moderate (to low) aggression. In the present case, these drivers got grouped into G1 (Table 8), G3 (Table 9), G6 (Table 10), and G9 (Table 11), respectively.

From the results presented in Tables 8 to 11, we observe that the identification accuracy *(% Recall)* for each of these high skilled drivers exceeds 80%, which is deemed very impressive. Does it therefore mean that certain behavioral

Table 8: Confusion matrix on group 1 (G1) of drivers.

Driver ID	Predicted				Recall (%)	Skill rank
	D014	D021	D034	*D058*		
Actual						
D014	6	5	0	0	54.4	4
D021	1	13	0	0	92.8	3
D034	0	1	27	3	87.1	2
D058	0	1	5	26	*81.2*	*1*
Precision (%)	85.7	65	84.3	*89.6*		

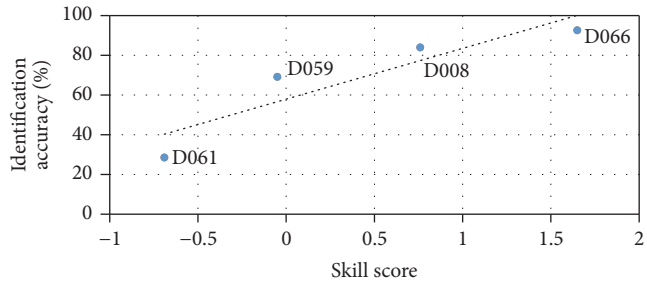


Figure 5: Identification accuracy obtained in a group with identical aggression displayed by the drivers.

attributes (like higher skill and low aggression) ensure significantly better identification in a group? Let us examine this further by forming arbitrary groups, that is, groups formed solely based on their relative standing in terms of aggression and skill.

We investigated the above hypothesis by forming a group of four drivers (D066, D008, D0059, and D061) who display similar aggression levels but widely separated in skill scores (refer to Table 7). Figure 5 shows the results. It is seen that, for such a group, the driver with highest skill factor can be identified as accurately as 93%. There is also a monotonic increase in identification accuracy with respect to increasing skill factor.

In order to test the hypothesis further, we conducted an investigation by forming a group with the four most skillful drivers, namely, D018, D071, D066, and D053. They are also closely spaced in 2D-aggression/skill plane, meaning thereby that the key attributes of their driving styles are also similar. The result is shown in Figure 6. The achieved accuracy for each of them is very high (>80%)

Considering these results in conjunction with the ones obtained for the same drivers in their natural groupings, one may tend to infer that the drivers displaying very high skill factor (along with moderate aggression) are accurately identifiable, irrespective of the group composition of similar size. However the effect of larger group size on the stated hypothesis needs further study.

In a similar vein, we investigate a group of drivers displaying lowest skills as well as being closely spaced in the 2D-aggression/skill plane. These are D004, D022, D050, and D056. The result is shown in Figure 7.

Comparing the results obtained from Figure 7 with their natural groups, it is found that, for the drivers with lower

Table 9: Confusion matrix on group 3 (G3) of drivers.

Driver ID	Predicted				Recall (%)	Skill rank
	D017	*D018*	D019	D021		
Actual						
D017	21	0	0	3	87.5	3
D018	0	13	0	0	*100*	*1*
D019	1	2	9	2	64.2	2
D021	4	0	1	9	64.2	4
Precision (%)	80.7	*86.6*	90	64.2		

Table 10: Confusion matrix on group 6 (G6) of drivers.

Driver ID	Predicted				Recall (%)	Skill rank
	D064	*D066*	D068	D069		
Actual						
D064	38	2	1	0	92.6	3
D066	4	22	1	0	*81.4*	*1*
D068	3	4	13	0	65	2
D069	1	1	0	13	86.6	4
Precision (%)	82.6	*75.8*	86.6	100		

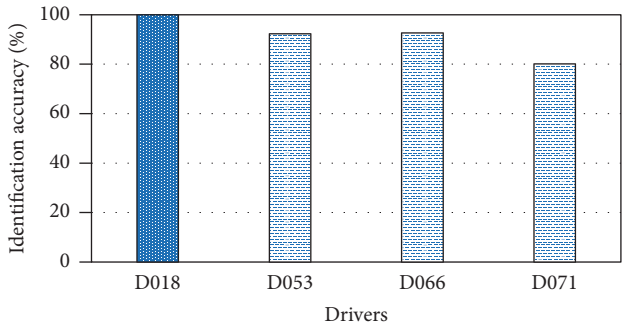


Figure 6: Identification accuracy obtained in a group of drivers with very high skills.

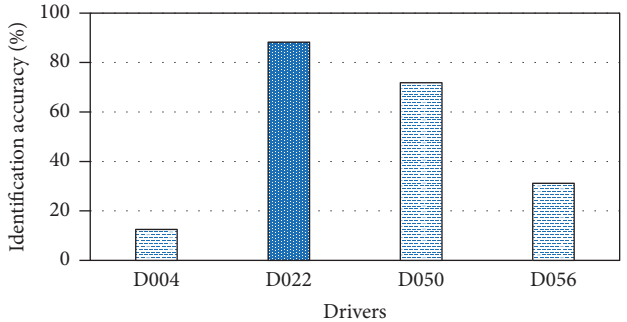


Figure 7: Identification accuracy obtained in a group of drivers with low skills and similar aggression.

skills, one may not be able to predict a given range of accuracy measures. These drivers are inconsistent in their driving styles. For all such cases, group composition does matter. Therefore we can state that the skill and aggression factors as seen in driving styles have considerable effect on driver identification accuracy.

4.3. Discussion. The objective of the presented work is to investigate whether we are able to uniquely identify a driver (amongst a group of drivers) solely based on the GPS data covering the entire trip. Our study shows that, even with such minimal sensing, it is possible to identify a driver with an accuracy of nearly 78%. The proposed method is not envisioned as a replacement for biometric authentication. The purpose of this study is to accurately identify a driver based on the statistical features of GPS data alone in contrast with the multisensor approach taken by previous authors. It is seen that, for a small group of drivers, an automated driver identification routine is feasible even with access to GPS data alone. At the same time, the proposed method throws some challenges. For example, since trip based feature set is only considered, its applicability to real-time monitoring scenarios remains doubtful.

We now compare our results with relevant prior works. Driver identification [21] by Burton et al. achieves 95% confidence interval on 10 drivers using a simulated data set. Fung et al. [22] achieve 61% accuracy for older (70+ years) drivers using only acceleration and speed. Moreira-Matias and Farah [23] achieve 88% accuracy on a large multisensory data obtained from 217 drivers. Moreira-Matias and Farah [23] also develop a generic machine learning methodology and feature reduction process. Martínez et al. [24] give accuracy above 80% with multisensor data on 11 drivers. A comparative analysis of our work with others is given in Table 12.

From Table 12, it is seen that the accuracy attained through our method is quite comparable to the published works. While the results are very promising, it is felt that additional dedicated sensors like brake pedal and steering

Table 11: Confusion matrix on group 9 (G9) of drivers.

Driver ID	Predicted				Recall (%)	Skill rank
	D055	D063	D070	*D071*		
Actual						
D055	0	0	2	0	0	2
D063	0	8	1	3	66.6	3
D070	0	0	1	0	100	4
D071	0	4	0	20	*83.3*	*1*
Precision (%)	NA	66.6	25	*86.9*		

Table 12: Comparison of the proposed method with other existing methods.

Method	Sensors	Data	Obtained accuracy
Our method	GPS only	38 drivers, 2 months	82.3%
Enev et al. [6]	Brake pedal	15 drivers	87%
Zhang et al. [7]	Multisensory	20 drivers, simulated driving	85%
Hallac et al. [8]	12 sensor readings, other than GPS	10 cars, 64 drivers	76% (for 2 drivers) 50% (for 5 drivers)
Burton et al. [21]	Multisensor, brake pedal, gas pedal, speed, etc.	10 drivers, simulated driving	95% confidence interval
Fung et al. [22]	Location and speed	14-old-age drivers	30–61%
Moreira-Matias and Farah [23]	Multisensor data	Dataset of 217 families collected over one year	88%
Martínez et al. [24]	Multisensor data	11 different drivers 25 Km (40 min.) data	Above 80%

wheel will certainly increase the identification accuracy to a large extent. Our data set does not contain any information about the traffic and weather conditions, when the trips were undertaken. Such additional information could have been utilized for natural group formation possibly leading to superior results.

5. Conclusions

In this paper, we present the results of our study on driver identification approach using only smartphone's GPS. Data collected from 38 drivers for a duration of two months and covering a total journey length of 50000 Km have been analyzed. This study demonstrates that a set of 137 features, extracted from the GPS data corresponding to the completed trips, can be used to identify the driver of a vehicle. For the purpose of driver identification, we have segregated the drivers into natural groups of four to five drivers in each group where route proximity is used as the deciding factor for such segregation. The investigations have offered important insights regarding driving behavior and its role vis-à-vis our ability to identify the driver accurately. We have observed that certain behavioral attributes (like higher skill, low aggression) ensure significantly better identification accuracy. The analysis presented in this paper shows that, for the "driver identification" problem, an average accuracy of 82.3% is achieved, where the drivers are grouped into multiple groups (each group has 4-5 drivers). Thus using only GPS, a good accuracy is obtainable and GPS can serve as a backbone of driver identification system. Further improvements can be achieved by using other sensors with GPS.

We conclude by stating that it is possible to identify drivers with reasonably high accuracy even when only the smartphone GPS is used. At the same time, it is to be noted that much more investigations are needed in the future in order to identify a driver with near certainty. It is to be seen whether the additional sensory information available in a vehicle's ECU (Electronic Control Unit) can be fused with this GPS data so that this accuracy is greatly enhanced.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] D. J. Watts, "Should social science be more solution-oriented?" *Nature Human Behaviour*, vol. 1, no. 1, p. 0015, 2017.

[2] J. Wahlström, I. Skog, and P. Handel, "Smartphone-Based Vehicle Telematics: A Ten-Year Anniversary," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[3] R. Fuller, "Towards a general theory of driver behaviour," *Accident Analysis & Prevention*, vol. 37, no. 3, pp. 461–472, 2005.

[4] J. Fuller, "Psychology and the highway engineer," *Human Factors for Highway Engineers*, pp. 1–10, 2002.

[5] N. Lin, C. Zong, M. Tomizuka, P. Song, Z. Zhang, and G. Li, "An overview on study of identification of driver behavior characteristics for automotive control," *Mathematical Problems in Engineering*, vol. 2014, Article ID 569109, 2014.

[6] M. Enev, A. Takakuwa, K. Koscher, and T. Kohno, "Automobile Driver Fingerprinting," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 1, 2016.

[7] X. Zhang, X. Zhao, and J. Rong, "A study of individual characteristics of driving behavior based on hidden markov model," *Sensors & Transducers Journal*, vol. 167, no. 3, pp. 194–202, 2014.

[8] D. Hallac, A. Sharang, R. Stahlmann et al., "Driver identification using automobile sensor data from a single turn," in *Proceedings of the 19th IEEE International Conference on Intelligent Transportation Systems, ITSC 2016*, pp. 953–958, Brazil, November 2016.

[9] J. Engelbrecht, M. J. Booysen, G.-J. Van Rooyen, and F. J. Bruwer, "Survey of smartphone-based sensing in vehicles for intelligent transportation system applications," *IET Intelligent Transport Systems*, vol. 9, no. 10, pp. 924–935, 2015.

[10] P. Themann, J. Bock, and L. Eckstein, "Optimisation of energy efficiency based on average driving behaviour and driver's preferences for automated driving," *IET Intelligent Transport Systems*, vol. 9, no. 1, pp. 50–58, 2015.

[11] N. Abuali and H. Abou-Zeid, "Driver behavior modeling: Developments and future directions," *International Journal of Vehicular Technology*, vol. 2016, Article ID 6952791, 2016.

[12] S. Laapotti, E. Keskinen, M. Hatakka et al., "Driving circumstances and accidents among novice drivers," *Traffic Injury Prevention*, vol. 7, no. 3, pp. 232–237, 2006.

[13] E. I. Vlahogianni and E. N. Barmpounakis, "Driving analytics using smartphones: Algorithms, comparisons and challenges," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 196–206, 2017.

[14] T. Tanprasert, C. Saiprasert, and S. Thajchayapong, "Combining Unsupervised Anomaly Detection and Neural Networks for Driver Identification," *Journal of Advanced Transportation*, vol. 2017, pp. 1–13, 2017.

[15] P. Phumphuang, P. Wuttidittachotti, and C. Saiprasert, "Driver identification using variance of the acceleration data," in *Proceedings of the 19th International Computer Science and Engineering Conference, ICSEC 2015*, Thailand, November 2015.

[16] J. F. Junior, E. Carvalho, B. V. Ferreira et al., "Driver behavior profiling: an investigation with different smartphone sensors and machine learning," *PLoS ONE*, vol. 12, no. 4, Article ID e0174959, 2017.

[17] A. Chowdhury, T. Chakravarty, and P. Balamuralidhar, "Estimating true speed of moving vehicle using smartphone-based GPS measurement," in *Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2014*, pp. 3348–3353, usa, October 2014.

[18] A. Chowdhury, A. Ghose, T. Chakravarty, and P. Balamuralidhar, "An improved fusion algorithm for estimating speed from smartphone's Ins/Gps sensors," *Next Generation Sensors and Systems*, vol. 16, pp. 235–256, 2015.

[19] T. Banerjee, A. Chowdhury, and T. Chakravarty, "Mydrive: Drive behavior analytics method and platform," in *Proceedings of the 3rd International Workshop on Physical Analytics, WPA 2016*, pp. 7–12, Singapore.

[20] R. Vaiana, T. Iuele, V. Gallelli, and D. Rogano, "Demanded versus assumed friction along horizontal curves: An on-the-road experimental investigation," *Journal of Transportation Safety & Security*, pp. 1–27, 2017.

[21] A. Burton, T. Parikh, S. Mascarenhas et al., "Driver identification and authentication with active behavior modeling," in *Proceedings of the 12th International Conference on Network and Service Management, CNSM 2016 and Workshops, 3rd International Workshop on Management of SDN and NFV, ManSDN/NFV 2016 and International Workshop on Green ICT and Smart Networking, GISN 2016*, pp. 388–393, Canada, November 2016.

[22] N. C. Fung, B. Wallace, A. D. C. Chan et al., "Driver identification using vehicle acceleration and deceleration events from naturalistic driving of older drivers," in *Proceedings of the 12th IEEE International Symposium on Medical Measurements and Applications, MeMeA 2017*, pp. 33–38, USA, May 2017.

[23] L. Moreira-Matias and H. Farah, "On Developing a Driver Identification Methodology Using In-Vehicle Data Recorders," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2387–2396, 2017.

[24] M. V. Martínez, J. Echanobe, and I. Del Campo, "Driver identification and impostor detection based on driving behavior signals," in *Proceedings of the 19th IEEE International Conference on Intelligent Transportation Systems, ITSC 2016*, pp. 372–378, Brazil, November 2016.

[25] P. Handel, I. Skog, J. Wahlstrom et al., "Insurance telematics: Opportunities and challenges with the smartphone solution," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 57–70, 2014.

[26] R. Vaiana, T. Iuele, V. Astarita et al., "Driving behavior and traffic safety: an acceleration-based safety evaluation procedure for smartphones," *Modern Applied Science (MAS)*, vol. 8, no. 1, pp. 88–96, 2014.

[27] T. Chakravarty, A. Chowdhury, A. Ghose, C. Bhaumik, and P. Balamuralidhar, "Statistical analysis of road-vehicle-driver interaction as an enabler to designing behavioral models," *International Journal of Modeling, Simulation, and Scientific Computing*, no. 2, 2014.

[28] A. Hall Mark and L. A. Smith, "Practical feature subset selection for machine learning," in *Computer Science '98 Proceedings of the 21st Australasian Computer Science Conference ACSC'98, Perth*, C. McDonald, Ed., pp. 181–191, Springer, Berlin, 1998.

[29] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.

[30] D. M. W. Powers, "Evaluation: from precision, recall and F-factor to ROC, informedness, markedness & correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[31] V. Van Asch, "Macro-and micro-averaged evaluation measures," Tech. Rep., 2013, http://www.cnts.ua.ac.be/~vincent/pdf/microaverage.pdf.

Hindawi
Journal of Advanced Transportation
Volume 2018, Article ID 4106086, 14 pages
https://doi.org/10.1155/2018/4106086

Research Article

Traffic State Estimation Using Connected Vehicles and Stationary Detectors

Ellen F. Grumert [1,2] **and Andreas Tapani** [1]

[1] *Swedish National Road and Transport Research Institute (VTI), 581 95 Linköping, Sweden*
[2] *Department of Science and Technology, Linköping University, 601 74 Norrköping, Sweden*

Correspondence should be addressed to Ellen F. Grumert; ellen.grumert@vti.se

Received 29 September 2017; Accepted 2 December 2017; Published 10 January 2018

Academic Editor: Fernando García

Real-time traffic state estimation is of importance for efficient traffic management. This is especially the case for traffic management systems that require fast detection of changes in the traffic conditions in order to apply an effective control measure. In this paper, we propose a method for estimating the traffic state and speed and density, by using connected vehicles combined with stationary detectors. The aim is to allow fast and accurate estimation of changes in the traffic conditions. The proposed method does only require information about the speed and the position of connected vehicles and can make use of sparsely located stationary detectors to limit the dependence on the infrastructure equipment. An evaluation of the proposed method is carried out by microscopic traffic simulation. The traffic state estimated using the proposed method is compared to the true simulated traffic state. Further, the density estimates are compared to density estimates from one detector-based method, one combined method, and one connected-vehicle-based method. The results of the study show that the proposed method is a promising alternative for estimating the traffic state in traffic management applications.

1. Introduction

Density, speed, and flow are important measures for describing the characteristics of the traffic on a road segment. The density of traffic is defined as the number of vehicles located on a road segment. The speed can be defined either as the time mean speed, which is the mean speed of all vehicles passing a specific location within a given time interval, or as space mean speed, which is the mean speed of all vehicles travelling over a road segment at a certain point in time. Finally, the flow is defined as the number of vehicles passing a specific location within a given time interval. The three measures are commonly referred to as the traffic state. The traffic state is of special interest for traffic management systems based on automatic control. Examples of such systems are variable speed limit systems and ramp metering. The purpose of the traffic management system is to improve the traffic conditions during, for example, congested periods and incidents. The systems should be able to detect abrupt changes in the traffic state, such as lower speeds and flows and higher densities, and use this information as input to apply a suitable control strategy. Thus, representative real-time estimation of the traffic state is of importance.

The traditional way to measure and estimate the traffic state is by the use of stationary equipment, for example, loop detectors and radars. Due to recent development in vehicle technology, different types of connected vehicles are being introduced and the expectation is that in 2020 75% of newly produced vehicles will be equipped with technology that enables the possibility to connect to the surroundings [1]. The connected vehicles facilitate communication between vehicles and between vehicles and the infrastructure. This allows for frequent updates of individual vehicle measures such as their speed and position. Hence, it is possible to use connected vehicles in combination with stationary detectors, or as a standalone data source, to estimate the traffic state. This can also result in improved spatial estimates instead of the traditionally used point estimates.

In this study, we propose a method for estimating the traffic state based on vehicle-to-infrastructure communication. The required information is speed and positioning measurements from connected vehicles in combination with counts from stationary detectors. By assuming that the connected vehicles have the same distribution of speed as regular vehicles, the speed is estimated as an average of the speeds of the connected vehicles. The only connected vehicle data needed to estimate the density is information about the current road segment of the connected vehicles. This makes the method robust with respect to errors in the positioning data. Each connected vehicle continuously communicates its location, which is used to estimate the total number of connected vehicles on a specific road segment. Further, the number of connected vehicles passing stationary detectors is, together with the total number of passing vehicles, used to estimate the penetration rate of connected vehicles. Thereby, the total number of vehicles located on a segment can be estimated.

Our hypotheses are that the proposed method will result in (1) density and speed estimates that can capture the current traffic conditions on the road, (2) a possibility to use more sparsely placed stationary detectors without considerably reducing the performance of the density estimation, given that the share of connected vehicles is assumed to be approximately the same over the road stretch, and (3) precise and fast detection of changes in the traffic state, especially for higher penetration rates of connected vehicles. In real applications, the traffic state estimation is one component of a larger model system, often including both data assimilation and fusion techniques. The aim of this paper is to find a straightforward approach to estimate the density and speed by the use of connected vehicles and investigate how well the traffic state estimation can capture the actual traffic situation as a first step towards using the proposed method as such a component in traffic management applications.

To study these hypotheses, the proposed method is evaluated by the use of microscopic traffic simulation. The density estimates are compared to one detector-based method, one combined method, and one connected-vehicle-based method. The comparisons are done in order to investigate if the proposed method gives estimates that are comparable to estimates of existing methods. A simulation scenario with an incident is analyzed to study how well the proposed method can capture abrupt changes in the traffic state. Two different distances between the detectors are applied to examine how the method performs with sparsely placed detectors. To isolate the effects related to the method, we use a simple design of the road network and assume that no measurement errors exist in detector and connected vehicle data.

The remainder of the paper is organized as follows. In Section 2, an overview of traffic state estimation methods is given with focus on density estimation. The proposed method for estimating the traffic state based on connected vehicles in combination with stationary detectors is presented in Section 3. The simulation setup and the evaluation method are described in Section 4, including an overview of the methods used for comparison. In Section 5, the performance of the proposed method is presented and compared to other methods. Finally, conclusions from the study and directions for further research are discussed in Section 6.

2. Traffic State Estimation Using Connected Vehicles

The traditional way to estimate the traffic state is by the use of stationary detectors such as loop detectors and radar detectors, as described by Kurkjian et al. [2], Coifman [3], and Singh and Li [4]. This is limiting the estimation to specific points in space, and the conditions in between detectors remain unknown. Hence, the estimation will be a good representation of the traffic state on the road section only under steady-state conditions, that is, when there is no change in the traffic conditions in space and time. This is usually not the case, particularly not for bottlenecks or during incidents, and therefore the density estimated with these methods will most probably deviate from the true density on the section. Hence, to give enough information about the traffic conditions on a longer road section, the detector-based method does require densely placed detectors (see, e.g., the method proposed by Singh and Li [4]). Data assimilation and fusion techniques including a traffic model are common methods to get a picture of the traffic state also in between the detectors. A number of studies using different underlying traffic models and different filtering approaches exist in the literature (see, e.g., Kurkjian et al. [2], Muñoz et al. [5], Wang and Papageorgiou [6], Mihaylova et al. [7], Singh and Li [4], and Duret et al. [8]). Methods have also been proposed, where no underlying model and no filtering are needed. See, for example, Coifman [3], where reidentification of vehicles and the vehicle conservation law is used to estimate the density between two detectors. Darwish and Bakar [9] conclude that methods using different types of stationary detectors can estimate the traffic state accurately but they are often expensive to install and maintain and are limited to small areas. Also, the information is usually transmitted with delay, since it has to be processed through a traffic information center.

Lately, when more data sources have become available, connected vehicles have been used as input to the filtering approaches in order to update the modeled traffic state. See, for example, Herrera and Bayen [10], Work et al. [11], Yuan et al. [12], Seo et al. [13], Astarita et al. [14], and Bekiaris-Liberis et al. [15]. Other traffic state estimation methods making use of connected vehicle data without an underlying traffic model are presented by Herring et al. [16], Herrera et al. [17], Van Lint and Hoogendoorn [18], Qiu et al. [19], Ma et al. [20], Bhaskar et al. [21], Zhang et al. [22], Seo et al. [23], and Montero et al. [24].

When speed measurements from connected vehicles are available, the speed can be estimated by calculating an average of the speeds of the connected vehicles. This requires that the connected vehicles have the same distribution of speeds as regular vehicles, similar to what has been done in the works of Astarita et al. [14] and Bekiaris-Liberis et al. [15]. Otherwise, the speed estimate would be biased towards the average speed of the connected vehicles.

The density estimate using connected vehicles requires some more calculations. One way of estimating the density

is by using connected vehicles together with traditional stationary detectors, here referred to as combined methods. For the combined methods, a weighted estimate based on both traditional detector measurements and connected vehicle measurements of, for example, speed, travel time, and/or location is used. Examples are the methods presented by Astarita et al. [14], Qiu et al. [19], Ma et al. [20], Bhaskar et al. [21], Zhang et al. [22], and Bekiaris-Liberis et al. [15]. The method by Qiu et al. [19], which was later extended by Ma et al. [20], detects the number of vehicles located within a segment. The density estimate is calculated by counting the number of vehicles that have passed the detector upstream of the segment at the times when a connected vehicle enters and exits a segment. According to Qiu et al. [19], the accuracy of the density estimates is better than when only stationary detector data is used to estimate density. Zhang et al. [22] use probe vehicle data and detector stations in order to estimate the space mean speed, and not density, on a road stretch. Astarita et al. [14] and Bekiaris-Liberis et al. [15] develop macroscopic cell transmission type models for the dynamics of the percentage of connected vehicles along the considered road. It is assumed that the connected vehicles move with the same average speed as the nonconnected vehicles, and hence no modeling of the speed dynamics is needed. In the work of Astarita et al. [14], the density is estimated by the percentage of connected vehicles based on counts of connected vehicles moving from one segment to another in the network and inflows measured at the ramps and at the boundaries of the network. Similarly, Bekiaris-Liberis et al. [15] use the penetration rate of connected vehicles, together with measurements of speed from the connected vehicles, the boundary flow, and the ramp flows measured through stationary detectors, to estimate the density. Also the combined methods do often require densely placed detectors to get good estimates. The methods are often based on retrospective measurements, such as travel time at an earlier point in time (see, e.g., Qiu et al. [19] and Ma et al. [20]). As a result, the density estimate might not reflect the current situation.

For the connected-vehicle-based methods, the connected vehicles are used to capture the surrounding traffic conditions at every point in space, and the methods are therefore not limited to fixed locations. Recent studies by Seo et al. [13, 23] investigate how the gap to a leading vehicle can be used to estimate the density on a road segment. The same method has been applied to an urban area in Montero et al. [24]. Further, the method is extended to also include measurements of the gap to the following vehicle. Another method is making use of vehicle spacings and speed as input data for estimating density [12]. This method requires a numerical model of the relationship between the traffic states to describe the changes in speed, flow, and density. Finally, Seo and Kusakabe [25] propose a method based on the number of vehicles located in between two connected vehicles to estimate the traffic conditions on the road. For methods using only connected vehicle data, the measurements used to estimate density are local, only including the connected vehicle and its surroundings, which might not necessarily reflect the density on a larger section of the road. Also, the methods often require identification of current lane, speed of the vehicle, distance to vehicle in front, and so forth.

To conclude, detector-based density estimation techniques make use of measurements from stationary detectors, usually consisting of flow and speed, to estimate density. However, the density estimates using stationary detectors are based on point estimates and are therefore limited due to the fact that the conditions in between detectors are not known. Therefore, the methods require densely spaced detectors to give density estimates that correspond well with the traffic conditions on the road. The detector-based method can be improved by including connected vehicle data. For combined methods, the need for continuous updates from connected vehicles can be limited; that is, it is enough with low frequency data communicated from the connected vehicles. However, the combined methods do usually still require densely spaced detectors and the information is sometimes based on retrospective connected vehicle measurements. The density estimates using only connected vehicle data are based on local density estimates including precise estimates of the density surrounding the connected vehicle. The connected vehicle density estimates are transmitted for further processing and converted to a density estimate of a larger area by including estimates from many connected vehicles. Hence, a representative density estimate can often only be reached with a high connected vehicle penetration rate or for a high flow level. Further, the connected vehicles are assumed to be able to continuously transmit information about their location, speed, gap to proceeding vehicle, and so forth.

3. A Method for Estimating the Traffic State by Using Connected Vehicle and Detector Data

We propose a combined method for estimating the traffic state on the road. The connected vehicle data is based on vehicle-to-infrastructure communication. The method is straightforward and it is possible to estimate the speed and density accurately based on limited information from the connected vehicles. Measurements from sparsely placed stationary detectors can be used without reducing the performance of the estimates. The purpose of the method is to get fast and representative traffic state estimates that can also be used to identify changes in the traffic conditions. Before introducing the method, a few essential assumptions are given:

(i) The connected vehicles are able to report their position, including information about their current road segment and speed, with a frequency of 1 Hz.

(ii) The stationary detectors are able to count and report the total number of vehicles, N, and the total number of connected vehicles, M, passing the detector within a given aggregation time period, T.

(iii) The connected vehicles are assumed to have the same distribution of speeds as the nonconnected vehicles.

Hence, neither the equipment used for communication of information for the connected vehicle nor the type of stationary detector (radar, Bluetooth, loop, etc.) is defined and

Table 1: Required measurements for estimating density based on CCV method, SD method, CC method, and GAP method.

Method	Connected vehicle data			Stationary detector data		
	Time at detector	Position	Measurement time stamp	Count of vehicles	Count of connected vehicles	Speed
CCV		X		X	X	
SD				X		X
CC	X			X		
GAP		X	X			

may vary as long as they fulfil the requirements presented above.

The traffic state estimation consists of two parts, a speed estimate and a density estimate for each segment on the considered road stretch. The speed estimates are based on simple calculations. It is assumed that the connected vehicles have the same distribution of speeds as the nonconnected vehicles. By communication of the individual speed of each connected vehicle i located on segment k at time t, the average speed of connected vehicles, $v_k^i(t)$, can be calculated. Then, the average speeds at time t are averaged over the aggregation time period, T, to get the final speed estimate,

$$\hat{v}_k(T) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{n_k}\sum_{i=1}^{n_k} v_k^i(t), \tag{1}$$

where n_k is the total number of connected vehicles on road segment k.

The density estimates make use of the position of each connected vehicle to get the total number of connected vehicles at each road segment, k, and for each time step, t. The average number of connected vehicles, $\overline{PV}_k(T)$, in segment k and for the aggregation time period, T, is used to estimate the total density. The stationary detector data is used to estimate the penetration rate on segment k as the number of connected vehicles, M_k, divided by the total number of vehicles, N_k. The penetration rate at the detector station located just upstream of segment k is used as input for the density estimate at segment k. The density estimate becomes

$$\hat{\rho}_k(T, X_k) = \frac{\overline{PV}_k(T)}{X_k}\frac{1}{(M_k/N_k)}. \tag{2}$$

The new speed and density estimates are becoming available after the latest aggregation time period T, based on the measurements within the same aggregation time period and, hence, the estimates are varying with time. The temporal indices in the density estimates have been suppressed to increase readability. The method is hereafter referred to as the Count Connected Vehicle (CCV) method. By assuming that the penetration rate is constant over a longer road section, detectors can be sparsely placed in order to have as little requirements on detectors as possible. In this case, each estimate of the penetration rate is applied to many segments before a new estimate of the penetration rate becomes available.

The information from the connected vehicles and the detectors is collected at each time step and communicated to a central unit, where it is being processed, resulting in time-dependent speed and density estimates. Finally, the estimates are aggregated over the aggregation time period T. Figure 1 gives an illustration of the process. The local units are the individual connected vehicles and the detectors. The central unit can, for example, be a roadside unit or a traffic management center used for further processing of data.

4. Evaluation Method

The analysis is divided into three parts. First, CCV is evaluated with respect to the aggregation time period, T, where the aggregation time period is defined as the time interval over which the traffic state is estimated. The performance of the traffic state estimation is examined for four different time periods. This will indicate for which aggregation time periods the method gives traffic state estimates that are representative for the traffic conditions on the road. Both density and speed estimates are considered. Second, the density estimates of the CCV are compared to similar methods: one method using only stationary detector data, one combined method, and one method using only connected vehicle data. The comparison is used to evaluate if the density estimates of the proposed method are comparable to those of detector-based, combined, and connected-vehicle-based methods found in the literature. Finally, it is investigated how well CCV manages to detect changes in the traffic states at different distances between detectors. Two cases are included with detectors placed 500 and 2500 meters away from each other. The results are compared to another combined method. This section describes the methods used for comparison, the simulation setup in a microscopic traffic simulation environment, including the choice of parameters, and the performance indicators used for evaluation.

4.1. Comparison of the Density Estimates. The density estimates are somewhat more complex to calculate compared to the speed estimates. Hence, methods with the same level of complexity are chosen for comparison. However, the methods require different types of data to estimate the density and include one stationary-detector-based method, SD, one combined method, CC, [19, 20], and one connected-vehicle-based method, GAP [23]. The required measurements for the different density estimation methods are summarized in Table 1. The measurements are divided into stationary-detector-based measurements and connected-vehicle-based measurements. The methods used for comparison with CCV are described below.

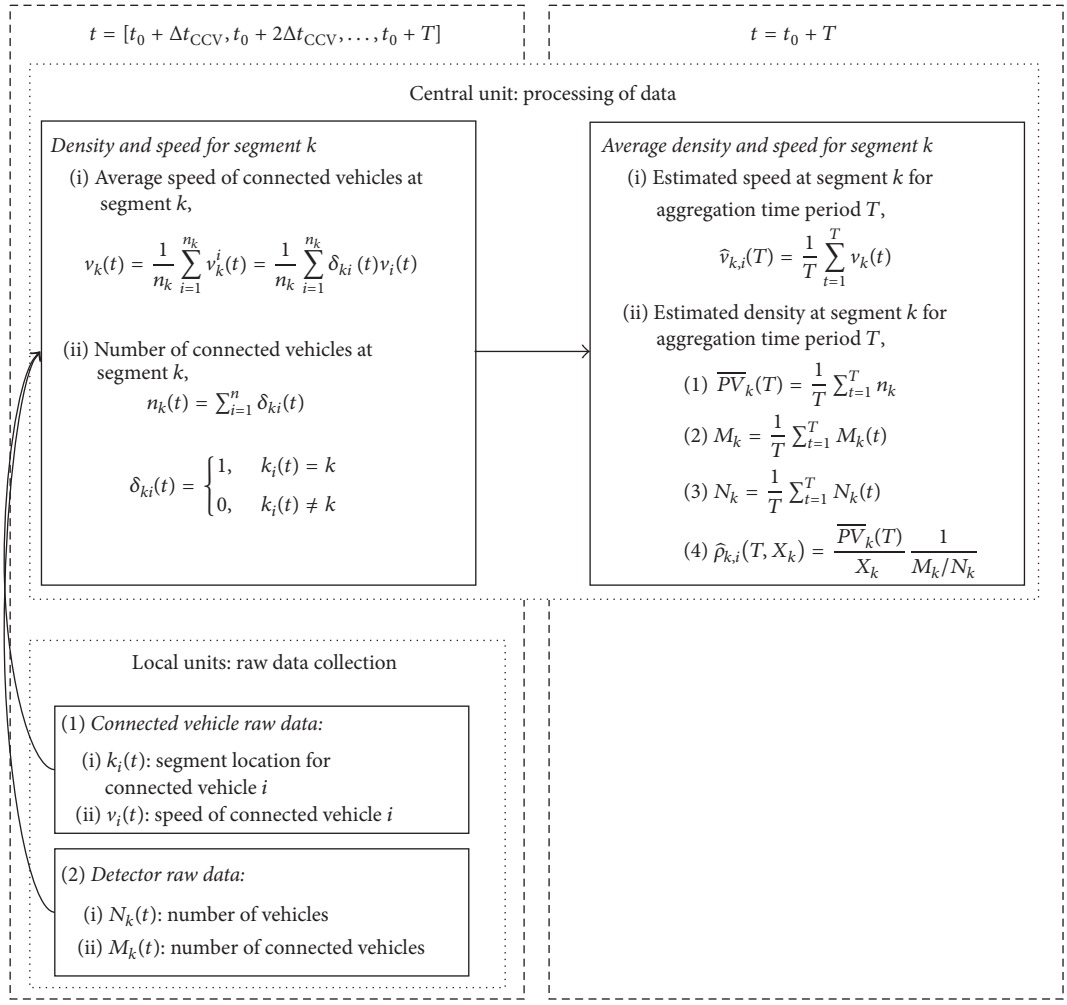


Figure 1: The communication flow between the local units (connected vehicles and detectors) and the central unit for the aggregation time period T.

A Detector-Based Method: the Stationary Detector (SD) Method. The detector-based method is using speed and flow measurements to estimate the density by the fundamental relationship

$$\overline{\rho}_k(T) = \frac{\overline{F}_k}{\overline{v}_k}, \tag{3}$$

where $\overline{F}_k$ and $\overline{v}_k$ are the mean flow and harmonic mean speed, respectively, detected at a detector station located just upstream of segment k and for the aggregation time period T. The method is hereafter referred to as stationary detector (SD) method. An alternative to using flow and speed measurements is to use the detector occupancy, that is, the amount of time a detector is occupied by vehicles. The occupancy can then be translated to density based on an estimate of the vehicle length. However, also by using occupancy, the resulting density will be represented only at a specific point. Further, the length of the vehicles has to be estimated or assumed.

A Combined Method: the Cumulative Count (CC) Method. The combined method is presented by Qiu et al. [19] and later extended by Ma et al. [20] and here is referred to as the Cumulative Count (CC) method. The method detects the number of vehicles located within segment k. Let $C(t_k^m)$ be the number of vehicles that have been detected up until time t_k^m, where t_k^m is the time connected vehicle m enters segment k. The density estimate for connected vehicle m is given by counting the number of vehicles, $C(t_k^m) - C(t_{k+1}^m)$, which have passed the detector upstream of the segment at the time when a connected vehicle m enters, t_k^m, and exits, t_{k+1}^m, the segment. The resulting density estimate becomes

$$\widetilde{\rho}_k(T, X_k) = \frac{1}{P_k}\sum_{m=1}^{P_k} \frac{C(t_k^m) - C(t_{k+1}^m)}{X_k}. \tag{4}$$

Here, the total number of connected vehicles, P_k, which exit segment k within the aggregation time period T is used to get an average density estimate. This means that the travel time for each connected vehicle within the segment can be

longer than the aggregation time period. Further, no vehicles are assumed to overtake the connected vehicle. This means that, on a segment with more than one lane, overtaking will result in a deviation from the actual density.

A Connected-Vehicle-Based Method: the GAP Method. The connected-vehicle-based method is presented by Seo et al. [23] and is referred to as the GAP method. In this method, connected vehicles are assumed to measure and communicate the gap to its leader, the position on the road, and time of the measurement. A set of connected vehicles, $\mathbf{P}_k(T, X_k)$, are located within the time-space region $[T, X_k]$. For each connected vehicle m, the total time, $tt_k^m(T, X_k)$, spent on segment k in aggregation time period T and the time-space gap, $|g_k^m(T, X_k)|$, between vehicle m and its leader is communicated. The density estimate on segment k becomes

$$\breve{\rho}_k(T, X_k) = \frac{\sum_{m \in \mathbf{P}_k(T,X_k)} tt_k^m(T, X_k)}{\sum_{m \in \mathbf{P}_k(T,X_k)} |g_k^m(T, X_k)|}. \quad (5)$$

This means that the trajectory of a connected vehicle, and its leader, within the segment has to be recorded in order to estimate the density. It should be noted that the measurement range of the gap between a connected vehicle and its leader has to be considered. If the gap is assumed to be measured through local on-board equipment, longer gaps might not be included due to a limited measurement range. This will lead larger gaps to be excluded from the density estimations, and as a result, the uncertainty in the estimations will increase. Further, the gap behind the connected vehicle is not included, which is of importance at platooning, since the gap between the connected vehicle and the vehicle behind can be long and maybe even not in the same segment. Since this gap is excluded, the density estimate becomes higher than the actual density on the road. Also, the frequency at which the connected vehicle data is transmitted is important. Between each time of communication for the connected vehicles, a linear relation in measured data is assumed. This can result in discretization errors that are larger at lower transmission frequencies.

4.2. Simulation Setup. The proposed method, as well as two of the comparison methods, requires identification and communication of information from connected vehicles. Microscopic traffic simulators describe individual vehicles in the traffic stream, allowing for gathering of information from single vehicles within the simulation. Thus, microscopic traffic simulation is suitable for analysis of the proposed method. In this study, we use the open-source microscopic traffic simulation tool SUMO (version 0.27.1) [26, 27]. SUMO is multimodal, space continuous, and time discrete. The car-following model used to model vehicle interactions is a further development of the work by Krauß [28] and is based on the calculation of a safe speed, compared to the approach of Gipps [29]. The lane-changing model is rule-based. The core model in SUMO is further described by Krajzewicz [30]. The connected vehicle and stationary detector data are accessed during the simulation through SUMO's Traffic Control Interface (TraCI). Python scripts [31] are used to implement the traffic state estimation methods.

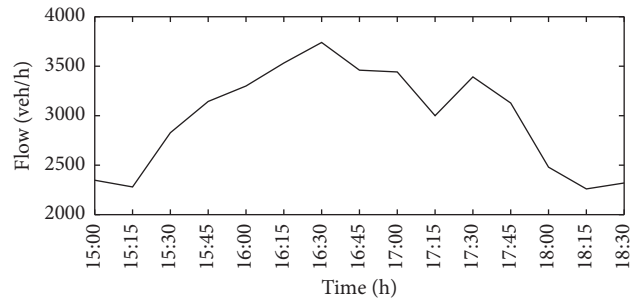


Figure 2: Inflow profile for the simulated scenario. The inflow profile is corresponding to the peak hours on a two-lane urban motorway in Stockholm.

The simulated scenario consists of a one-directional two-lane motorway, divided into ten 500-meter segments. Further, a segment for loading of vehicles and an end segment are included to avoid boundary effects, resulting in a 6 km long simulated road. The maximum allowed speed on the road is assumed to be 100 km/h. Simulations are performed with a flow pattern taken from flow measurements on a two-lane motorway in Stockholm during afternoon peak hours on a normal weekday (see Figure 2). The simulation is performed for a period of 3.75 hours, excluding a warm-up period of 5 minutes to prevent from loading effects. Further, eight different connected vehicle penetration rates are investigated. For the investigation of different distances between detectors, abrupt changes in the traffic state are required. Changes in the traffic state are modeled as an incident by letting one vehicle halt on a road segment for ten minutes after one hour. This is resulting in a temporary drop in capacity, which is considerably changing the traffic conditions.

4.3. Vehicle Parameters. The data used for calibration is collected through stationary radar detectors from four different locations on a two-lane urban motorway in Stockholm. The measurement from each detector consists of speed and flow measurements averaged over 15 minutes from three days in April 2016, with a typical pattern and no larger incidents reported. The flow profile is presented in Figure 2. The speed measurements, averaged over the three days, at the different detector locations and for different time instants are given in Figure 3(a). However, the original road stretch includes some on- and off-ramps, which have been excluded in this simulation study. The reason for this is to isolate the effects of the method by using a simple simulation scenario before trying more complex scenarios. Additionally, vehicle data at free flow conditions for four different vehicle classes are available for calibration of the desired speed distribution. The composition of vehicles and the speed distribution for each vehicle class are given in Table 2.

The calibration of a SUMO model of the two-lane motorway in Stockholm has resulted in calibrated vehicle parameters that are applied for this study. Figure 3 gives an overview of the speed at the four detectors and at

Table 2: Free flow speed distribution for the different vehicle classes.

Class	Allowed speed limit (km/h)	Vehicle length (m)	Mean speed (km/h)	Std of speed (km/h)	Composition (%)
1	110	0–8	109	13.13	91
2	90	8–12.5	92	10.33	5
3	90	12.5–24	90	8.35	3
4	80	24–36.5	88	5.55	1

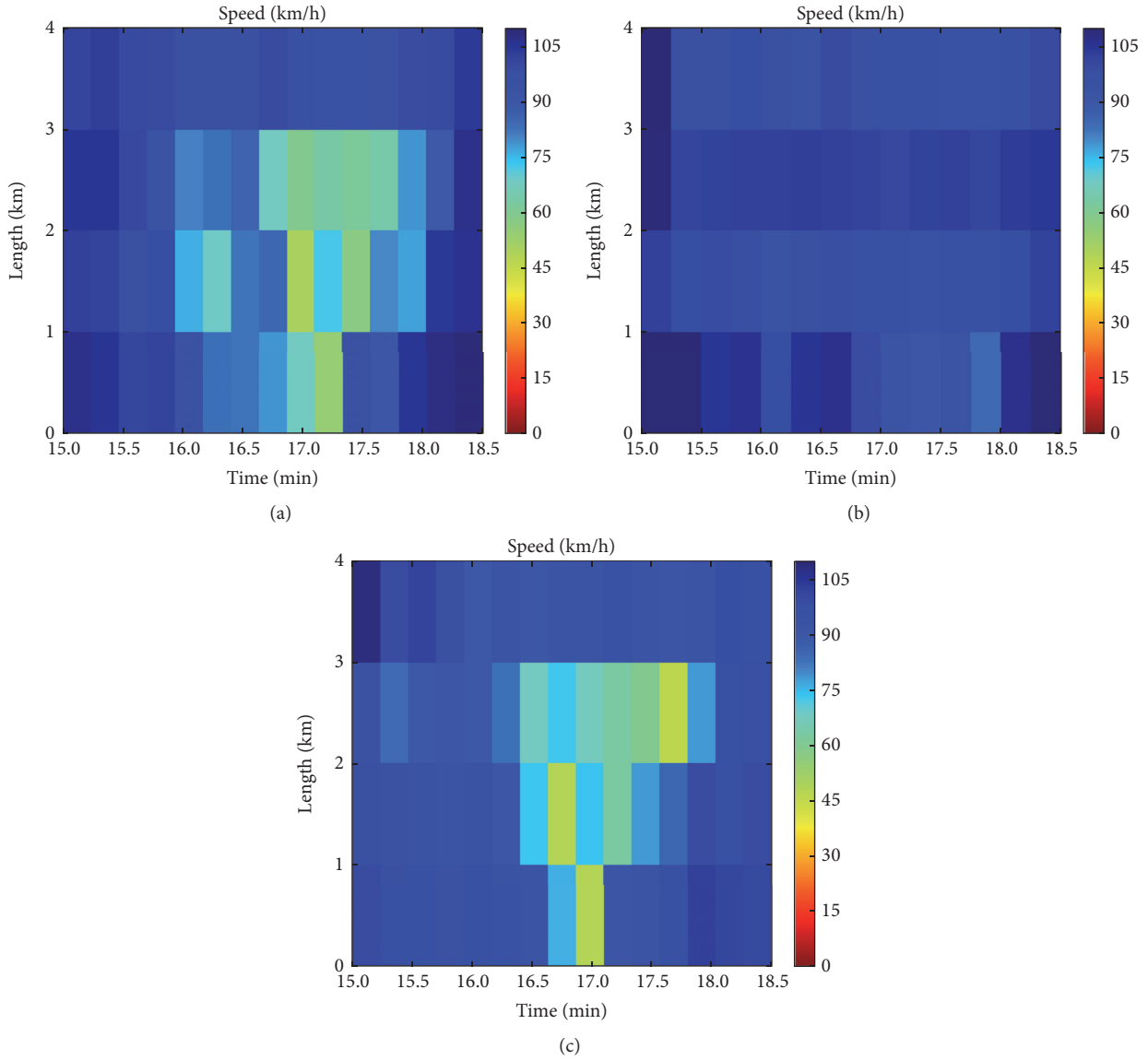


Figure 3: The mean speed for the given detector measurements (a), the uncalibrated scenario (b), and the calibrated scenario (c). The detectors are represented on y-axis and the time instants are represented on x-axis (15 min intervals). The mean speed is ranging from 0 km/h (red) to 110 km/h (blue).

15 min intervals for the measurements (a), the uncalibrated simulated measurements in SUMO (b), and the calibrated measurements in SUMO (c). The speed ranges from 0 km/h (red) to 110 km/h (blue). The default parameters and the calibrated parameters in SUMO are given in Table 3. The lane-changing parameters, lcCooperative and lcSpeedGain, have been changed substantially compared to the default values in SUMO. The reason for this is that by using the default values the throughput becomes much higher than observed on the motorway in Stockholm (see Figure 3(b)). Here, lcCooperative parameter controls the degree of cooperation with other vehicles when performing a lane-change.

Table 3: Vehicle parameters based on the calibrated scenario.

Parameter	Default	Class 1	Class 2	Class 3	Class 4
Accel. ability (m/s^2)	2.6	0.8	0.8	0.6	0.6
Decel. ability (m/s^2)	4.5	8.0	8.0	6.0	6.0
Imperfection (range: 0-1)	0.5	0.8	0.8	0.8	0.8
Reaction time (s)	1.0	0.85	0.85	0.85	0.85
lcCooperative (range: 0-1)	1.0	0.4	0.4	0.4	0.4
lcSpeedGain (range: 0–inf)	1.0	45.0	45.0	45.0	45.0
lcKeepRight (range: 0–inf)	1.0	1.0	1.0	10.0	10.0

A lower value results in decreased cooperation and thereby a decreased capacity on the road. The lcSpeedGain parameter is related to the willingness to increase speed in order to perform a lane-change. Hence, by increasing this parameter, the willingness to perform a lane-change is increased. This results in increased interaction between the vehicles due to the increased number of lane-changes, which in the end reduces the capacity on the road. Also, the willingness to keep right, controlled by the parameter lcKeepRight, has been increased for trucks and buses, corresponding to classes 3 and 4. The reason for this is that when the willingness to perform a lane-change is increased, the trucks and buses will also perform lane-changes more frequently, which is not the case in reality. The adjustments of the lane-changing parameters result in a throughput for the simulated scenario that is comparable to the available detector measurements. In the car-following model, the acceleration and deceleration abilities, as well as reaction time and the driver imperfection, have been adjusted to correspond to the actual capacity on the road. From Figure 3(c), it can be concluded that, by applying the final calibrated parameters, the simulated scenario is able to reproduce the measured scenario in terms of mean speed levels.

Vehicles are generated with exponentially distributed headways. The connected vehicles are also assumed to be generated with exponentially distributed headways and uniformly distributed in the total flow and between vehicle types. The connected vehicle data is collected and transmitted with a frequency of 1 Hz. The connected vehicles are able to detect a vehicle in front at a maximum distance of 1500 meters.

4.4. Performance Indicators. During the simulation, the number of vehicles within each segment and at each time step is counted and averaged over the aggregation time period to get the "true" simulated density and speed, hereafter referred to as the reference density. The reference density is compared to the density estimated using the proposed method, CCV, and the methods used for comparison (SD, CC, and GAP). The true simulated space mean speed is calculated by averaging of the speeds of all vehicles located within a segment for the considered aggregation time period. The comparison of both density and speed estimates is done with respect to the Root Mean Square Error (RMSE) for the total number of observations N:

$$\text{RMSE} = \sqrt{\frac{\sum_{n=1}^{N} \left(\text{est}_n - \text{obs}_n\right)^2}{N}}, \quad (6)$$

where est_n and obs_n are the estimate and the reference value for observation n, respectively. The estimate and the reference value are either speed or density. The difference between the estimated and the observed values is calculated for each segment and each aggregation time period and summarized to get the RMSE.

The RMSE is an aggregate performance measure of the estimations based on the total number of observations and it will not therefore capture how changes in the traffic conditions are reflected in the density and speed estimates. Therefore, an evaluation of how well the density estimates of CCV capture the changes in the traffic conditions is done by examining the estimated density and the reference density in a time-space diagram for an incident scenario. Further, a comparison of the time-space diagram of the density estimates for CCV and the other combined method, CC, is done. The means and standard errors of the means are calculated based on 10 replications of the simulation for the different scenarios.

5. Results

In this section, results from the simulation experiments are given. First, the aggregation time period is examined. This is followed by an investigation of the performance of CCV compared to SD, CC, and GAP. Finally, the ability for CCV to capture changes in the traffic conditions is presented.

5.1. Aggregation Time Period for CCV. The aggregation time period has to be chosen carefully. For a short aggregation time period, changes in the density can be discovered fast. On the other hand, the estimations become more uncertain due to a limited amount of data to base the estimation upon, especially for lower penetration rates. For a large aggregation time period, the traffic state estimation becomes more stable, but there is also a risk to smooth out changes and thereby miss useful and relevant information. The performance of CCV for the aggregation time periods of 15, 30, 60, and 120 seconds is examined by comparing the estimated density and speed to the reference density and speed, as explained in Section 4.4.

At low penetration rates and for short aggregation time periods, it is often not possible to estimate the traffic state due to no connected vehicle measurements to base the estimation on. In this case, the missing estimates are excluded from the RMSE and the resulting RMSE becomes uncertain. To give an indication of the amount of missing estimates at different penetration rates, the mean percentage of missing

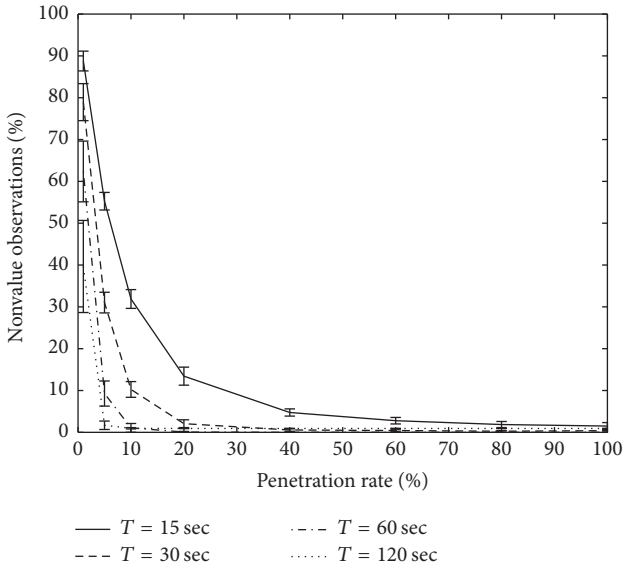


Figure 4: Mean percentage of missing estimates per segment for CCV at different penetration rates. The error bars are 95% confidence intervals, assuming normally distributed missing number of estimations over the simulation runs.

estimates per segment is given in Figure 4. The mean RMSE for speed and density for different penetration rates is shown in Figure 5.

From Figure 5(a), it is observed that the aggregation time period has limited effect on the RMSE of speed. Further, the RMSE of speed is highest for low penetration rates and decreases for higher penetration rates. However, this is the case only when it can be assumed that the connected vehicles have the same speed distribution as nonconnected vehicles. If the distribution of speed for the connected vehicles deviates from the distribution of speed of nonconnected vehicles, the proposed method for estimating the speed will be biased towards the speed of the connected vehicles. This will result in a higher or lower speed estimate depending on whether the connected vehicles are assumed to drive faster or slower than nonconnected vehicles.

Further, it is observed that the aggregation time period will have a great effect on the density estimates. As can be seen in Figure 4, a low penetration rate of connected vehicles will result in a large percentage of missing measurements. As a result, no, or only uncertain, traffic state estimates are available. This does explain the increase in RMSE with increasing penetration rates observed in Figure 5(b) for low penetration rates and for the aggregation time periods of 15–60 seconds. Since nonvalue estimates are excluded from the resulting RMSE at low penetration rates, an increase in penetration rate leads more observations to become available and more, but uncertain, estimates contribute to a larger RMSE. However, at penetration rates above 5–20%, depending on the aggregation time period, enough measurements are included in the estimation to give a reliable result and the RMSE is starting to decrease with increased penetration rate of connected vehicles. When the penetration rate is 100%, the RMSE of both speed and density are, as expected, zero due to the exclusion of measurement errors.

As a conclusion, the RMSE of speed and density are smallest for the aggregation time period of 120 seconds. However, by using such a large aggregation time period, the estimates become smoothed over a longer period. This might result in the fact that important phenomena in the traffic conditions are detected late or are even not detected at all. In order to reduce the uncertainty in the estimations and at the same time be able to capture the changes in the traffic flow, which is important for identifying changes in the traffic conditions, an aggregation time period of 60 seconds is therefore chosen for further investigations.

5.2. Performance of CCV. The density estimates of the CCV are compared to the density estimates of GAP, CC, and SD in order to investigate how the proposed method performs compared to existing methods. The mean RMSE of density for the four methods and using different penetration rates are presented in Figure 6. The aggregation time period is set to 60 seconds based on the results presented in Section 5.1.

As concluded in Section 5.1, the CCV method does not show a clear relationship between penetration rate and performance for penetration rates below 10% as a result of the missing or limited measurements to base the estimations on. Hence, for a penetration rate of connected vehicles of 1 and 5%, the RMSE of density for the CCV is not trustworthy. However, since the CCV performs poorly at low penetration rates as shown in Figure 6, it can be concluded that the penetration rate must be higher than 10% to give reliable estimates.

The density estimates based on GAP, CC, and CCV are, as expected, improved with an increased penetration rate. The CCV gives density estimates comparable to GAP at a penetration rate of 20%. It is first at a penetration rate of 40–50%; the accuracy in the estimates of CCV is comparable to estimates of CC and SD, although after a penetration rate of 40%, the improvements in the density estimates are rapid with increased penetration rate. When approaching a penetration rate of 100%, the estimated density is close to the reference density, unlike the other methods where a larger difference still exists. The reason for this is that, for CCV, a penetration rate of 100% means identifying all vehicles located within the segment and this is expected to be the same as the reference density over the same aggregation time period.

The other combined method, CC, has a larger RMSE of density at higher penetration rates. One reason for this might be the fact that it is assumed that no vehicles are overtaking the connected vehicle. Therefore, when the connected vehicles arrive at the downstream detector, all vehicles arriving at the upstream detector after the connected vehicle are assumed to still be within the segment. But since the segment consists of two lanes and the vehicles within the simulation have different desired speed, overtaking will occur, especially for connected vehicles with a low desired speed.

By using only connected vehicles, as is done in GAP, the local nature of the method does not seem to capture the traffic conditions in a larger area during inhomogeneous

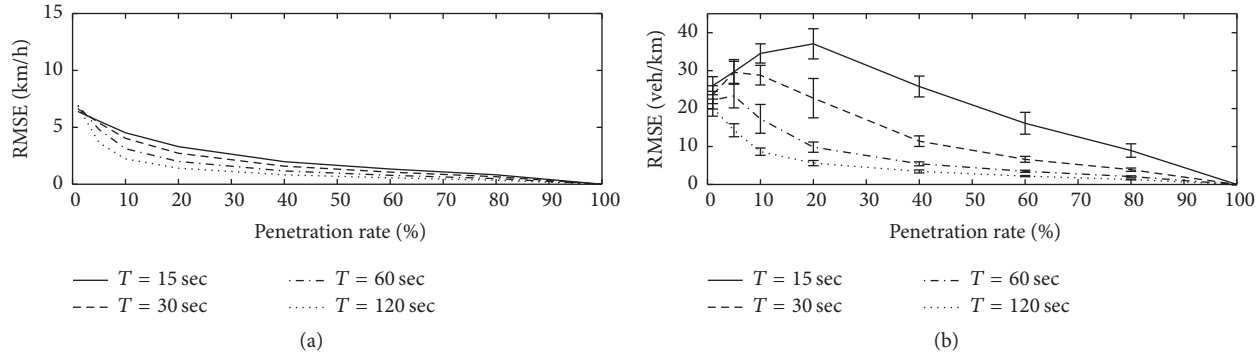


Figure 5: Mean RMSE for the estimated speed (a) and density (b) compared to the reference density and speed for CCV at different penetration rates and for different aggregation time periods. The error bars in (b) are based on 95% confidence intervals, assuming normally distributed RMSE over the simulation runs. The standard error of mean in (a) is at most 0.44 km/h, and hence the confidence intervals are excluded from the figure for greater legibility.

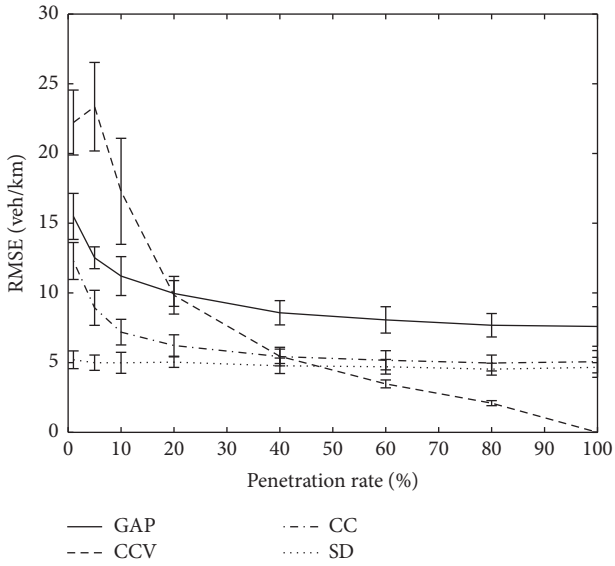


Figure 6: Mean RMSE of the estimated density compared to the reference density for the different methods and using different penetration rates. The aggregation time period is 60 seconds. The error bars are 95% confidence intervals, assuming normally distributed missing number of estimations over the simulation runs.

traffic conditions where the speed distribution of the vehicles becomes wider. By only considering the gap to the vehicle in front, the formation of platoons, with small gaps for the vehicles within the platoons and larger gaps in between platoons, will decrease the performance and there will be a bias towards gaps within the platoons. This is especially observed at medium flow levels where the left lane is mostly used for overtaking. In this case, a leader is sometimes missing as a result of the limited measurement range or the time-space distance behind the platoon is unknown in the overtaking lane where the traffic flow is more inhomogeneous. This is why the RMSE for the GAP method does not decrease as much as the other methods at higher penetration rates.

Finally, it is concluded that the density estimates using SD are, as expected, independent of the penetration rate. The difference between the true density and the density estimates for SD is due to the fact that the density estimates for SD are based on measurements at specific locations and not measurements for the whole segment.

As a conclusion, the SD is stable and gives reliable estimates of the density independent of the connected vehicle penetration rate and flow level. For the combined methods, the CC is preferable at lower penetration rates, while the CCV is more accurate at higher penetration rates. Results from GAP show that using only local measurements from connected vehicles gives a lower accuracy in the density estimates under inhomogeneous traffic conditions as a result of the inability to measure the gap behind vehicles and the limited measurement range. Since the accuracy of the density estimates is higher for the combined methods and the detector-based method, it is preferable to include stationary detectors when estimating the density.

5.3. Evaluation of Effects of Different Distances between Detectors. One of the advantages of CCV is that it can capture changes in the traffic conditions in between detectors, meaning that even though the detectors are sparsely placed, smaller segments in between the detectors can be considered for estimating the density. However, this is based on the assumption that the connected vehicle penetration rate is approximately constant in between detectors. For this reason, CCV is compared to the other combined method, CC, to investigate how the ability to detect changes in the traffic conditions, in this case modeled as an incident, is affected by sparsely located detectors for the two methods. Since CC has to include the whole stretch between two detectors as one segment, it is expected that the resulting performance is improved for the CCV.

Two time-space diagrams of density are presented in Figure 7. The distances between detectors are 500 and 2500 meters, respectively. An aggregation time period of 60 seconds is used based on the investigation in Section 5.1. A penetration rate of 40% is used, since CCV and CC are

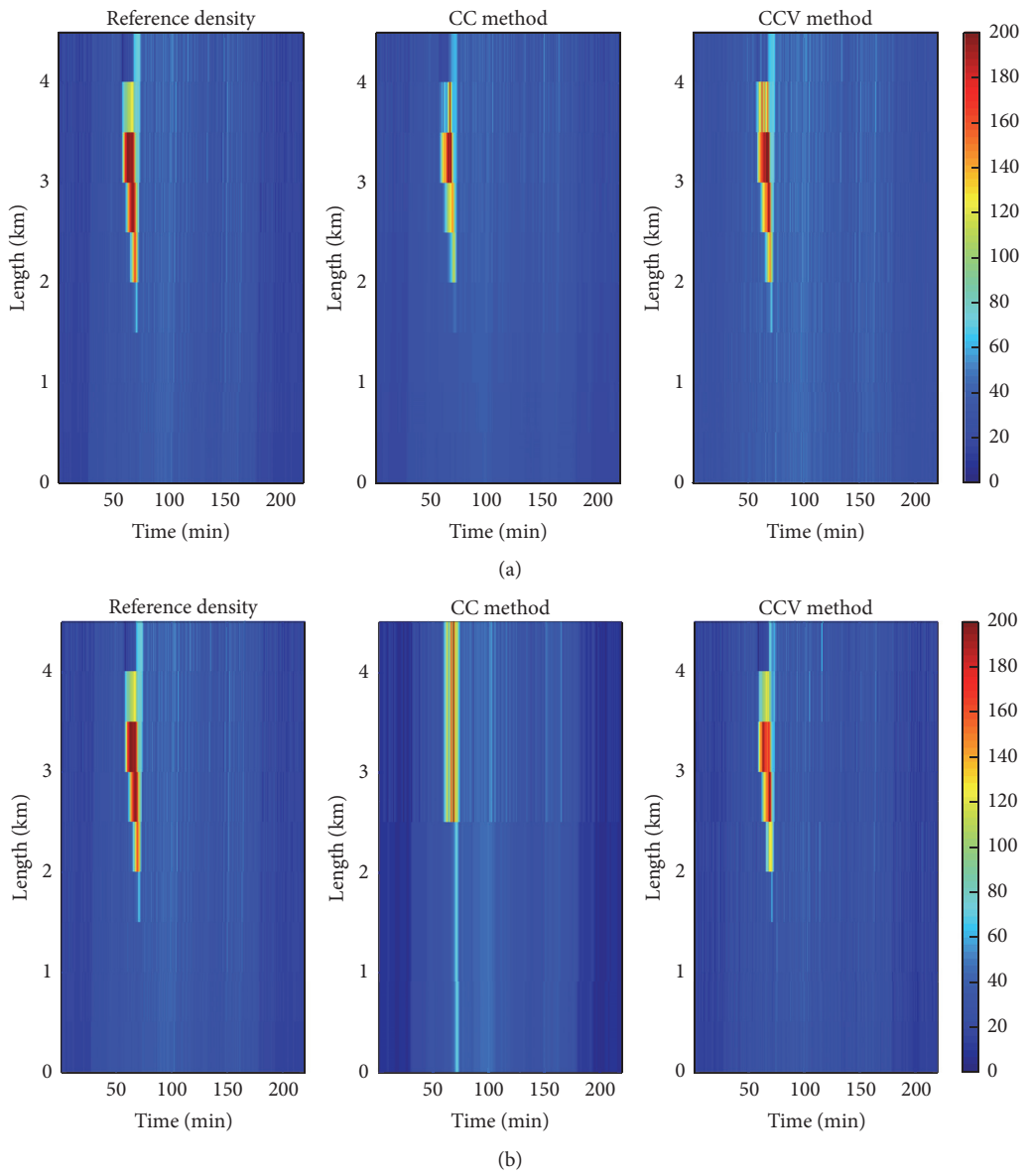


Figure 7: Reference density and density estimates using CC and CCV in time (min) and space (km) for a simulated incident scenario. The aggregation time period is 60 sec and the detector interval is 500 meters and 2500 meters in (a) and (b), respectively. The color map shows density (veh/km).

comparable in accuracy at this level. The presented results are means over 10 simulation runs. As can be seen in Figure 7(a), both CCV and CC manage to capture the tail of the incident with a distance between the detectors of 500 meters. However, both methods seem to overestimate the density at the incident and underestimate the density in the tail of the incident. An explanation to the overestimation for the CC is that vehicles might overtake the connected vehicle at an incident and pass to the next segment before the connected vehicle has exited the considered lane. This will result in a too large density estimate. Further, the connected vehicle data is sometimes based on retrospective information and information smoothed out over a longer time period than the aggregation time period, which will reduce the performance of CC especially during congested conditions, such as an incident. One reason for the deviation in the estimation for CCV might be due to an uncertain estimate of the penetration rates caused by the low speed levels at the detectors. This is further investigated below.

With more sparsely placed detectors, as in Figure 7(b), the same behaviour is seen for CCV, since the segment length is still 500 meters and the penetration rate in this example is approximately the same over the whole stretch. However, for CC, the distance between detectors and thereby the distance over which the density is estimated is considerably increased, which will result in decreased accuracy of the density estimates. Further, a consequence of the more sparsely located detectors is an even longer time delay in the estimates. As a

result, the incident is detected late when using the CC method and as a smoothed density estimate over the five segments without detectors.

It is concluded that CCV seems to capture changes in the traffic conditions, such as an incident. A benefit of CCV is that the estimates are based on the current situation on the road. This is becoming important during incidents and where the longer travel times seem to cause delays in the density estimation for CC. A second benefit is that, under the condition that the connected vehicle penetration rate can be assumed to be constant over a longer road stretch, as is the case in this study, CCV manages to capture changes in the traffic conditions also for longer distances between detectors. This is seen when the distance is increased to 2500 meters instead of 500 meters, whereas the results from CC are affected by even increased delay, as well as smoothing, resulting in the fact that the incident is detected late and as a smoothed density over five segments. However, it should be noted that CC does greatly limit the dependence of communication equipment, since reporting is only necessary at specific locations. Hence, when communication of connected vehicle data can only be done at specific points in time, or at specific locations, combined density estimation methods not dependent on continuously transmission of connected vehicle data are preferable over CCV.

The Effect of Penetration Rates at Incidents for the CCV. From Section 5.3, it is concluded that CCV overestimates the density at incidents and especially in the case of densely placed detectors. One reason for this could be an incorrect estimate of the penetration rate close to the incident. Therefore, a reference penetration rate, gathered by counting the number of vehicles and the number of connected vehicles on each segment in the simulation, is compared to the estimates from the CCV for the simulated incident scenario. Figure 8 gives an overview of the estimated and the reference penetration rate over time at the segments close to the incident.

From the figure, it becomes clear that the reference penetration rate and the estimated penetration rate deviate more during the incident and for the affected segments. This confirms that the estimate of the penetration rate is the reason for the uncertain estimates. It seems like, in order for the model to perform well under low-to-medium penetration levels, the traffic condition within a segment has to be homogeneous. Hence, if the segment can be subdivided to one congested and one uncongested traffic state, the penetration rate between those states might differ, resulting in an inaccurate estimate of the penetration rate and as a result an inaccurate estimate of the density. Further, the estimates of the penetration rate at detector intervals of 500 meters are more unstable close to the incident. This is probably due to the fact that the congestion is moving upstream, resulting in slow moving vehicles and a more uncertain number of connected vehicles passing the detectors, which creates local variations in the connected vehicle distribution in space. This corresponds well with the larger error in the density estimates at the incident for more densely placed detectors, which can be seen in Figure 7(a). For that reason, a more sophisticated method for estimating the penetration rate would probably increase the performance of CCV even at low penetration rates. Also the division of the road into smaller segments can reduce the possibility of having different traffic conditions within one segment. In this case, the penetration rate still needs to be estimated individually at each segment to increase the performance, since the inaccurately estimated penetration rate is the main problem.

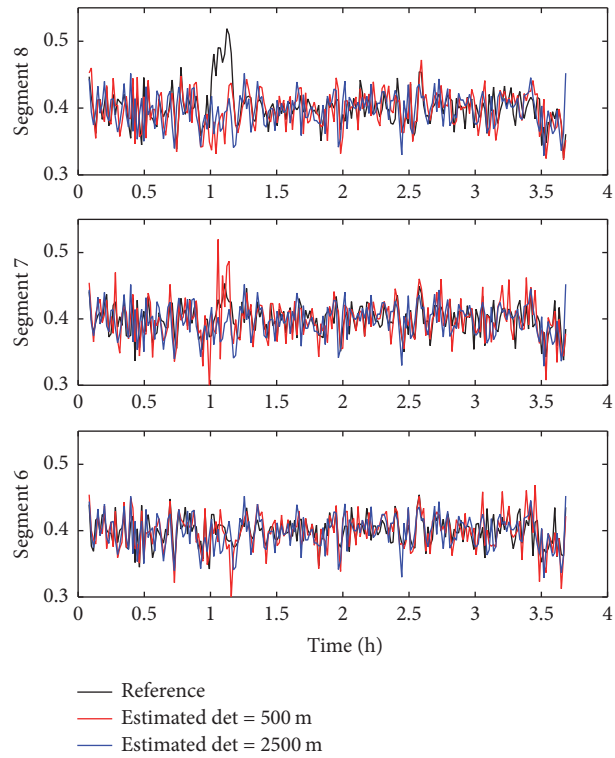


Figure 8: The reference (black) and the estimated penetration rate for segments 6–8 with a detector distance of 500 meters (red) and 2500 meters (blue).

6. Conclusions

We propose the Count Connected Vehicle (CCV) method for estimating the traffic state using connected vehicles in combination with stationary detectors. The method provides a straightforward approach to estimate the speed and density based on vehicle-to-infrastructure communication. The purpose of the method is to get fast and accurate traffic state estimations that can be used to detect changes in the traffic conditions and at the same time limit the dependence on detailed measurements communicated from the connected vehicles using as few stationary detectors as possible. The only measurements required from the stationary detectors are the number of passing vehicles and the number of passing connected vehicles. Hence, the complexity of the stationary detectors is limited. Moreover, if the penetration rate is assumed to be approximately the same over a longer road stretch, the stationary detectors can be sparsely placed. The limited dependence on measurements from the connected vehicles is due to the fact that the requirement in precision

in position data can be low. The only requirement is that the current segment be correctly reported. Hence, the measurement errors for estimating the number of connected vehicles located on a segment are limited to the boundaries of each segment.

The method is evaluated by the means of microscopic traffic simulation. The speed and density estimates of the proposed method are compared to the true simulated values. Further, the density estimates are compared to the density estimates using one detector-based method, one combined method, and one connected-vehicle-based method. The results of the study show that the proposed method is a promising alternative for accurately estimating density and speed on the road, especially at medium-to-high penetration rates of connected vehicles. Since the method is based on real-time positioning data from connected vehicles, it can capture abrupt changes in the traffic conditions, such as incidents. The traffic state can be estimated instantaneously given that there are at least a few connected vehicles at the segment. Hence, the aggregation time period can be short. This makes the method useful for traffic management purposes. Note that the penetration rate can be based on previous knowledge, that is, a moving average, during short aggregation time periods, to increase the accuracy in the estimation of the connected vehicle penetration rate.

Many interesting topics are identified for future research. First, a low connected vehicle penetration rate and a short aggregation time period are concluded to result in few or sometimes missing density estimates. Hence, by including data assimilation techniques, such as the ones described by Evensen [32], Wang and Papageorgiou [6], Antoniou et al. [33], and Seo et al. [13], the performance of the proposed method can be improved. Second, the penetration rate becomes inaccurate during inhomogeneous traffic conditions within a segment. By using a more advanced method to estimate the penetration rate, the accuracy of the method can be improved. For example, the dynamics of the penetration rate can be modeled using a macroscopic model as in Astarita et al. [14] and Bekiaris-Liberis et al. [15]. Third, the effect on the traffic state estimation for more complex designs of the road network and inclusion of measurements errors has to be further investigated. Finally, the use of the proposed method in traffic management systems is also a topic for future research.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to acknowledge the Swedish Transportation Administration (Trafikverket) for the financial support.

References

[1] R. Coppola and M. Morisio, "Connected car: technologies, issues, future trends," *ACM Computing Surveys*, vol. 49, no. 3, article 46, 2016.

[2] A. Kurkjian, S. B. Gershwin, P. K. Houpt, A. S. Willsky, E. Y. Chow, and C. S. Greene, "Estimation of roadway traffic density on freeways using presence detector data," *Transportation Science*, vol. 14, no. 3, pp. 232–261, 1980.

[3] B. Coifman, "Estimating density and lane inflow on a freeway segment," *Transportation Research Part A: Policy and Practice*, vol. 37, no. 8, pp. 689–701, 2003.

[4] K. Singh and B. Li, "Estimation of traffic densities for multilane roadways using a markov model approach," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4369–4376, 2012.

[5] L. Muñoz, X. Sun, R. Horowitz, and L. Alvarez, "Traffic density estimation with the cell transmission model," in *Proceedings of the American Control Conference*, vol. 5, pp. 3750–3755, IEEE, June 2003.

[6] Y. Wang and M. Papageorgiou, "Real-time freeway traffic state estimation based on extended Kalman filter: a general approach," *Transportation Research Part B: Methodological*, vol. 39, no. 2, pp. 141–167, 2005.

[7] L. Mihaylova, R. Boel, and A. Hegyi, "Freeway traffic estimation within particle filtering framework," *Automatica*, vol. 43, no. 2, pp. 290–300, 2007.

[8] A. Duret, L. Leclercq, and N.-E. El Faouzi, "Data assimilation using a mesoscopic Lighthill-Whitham-Richards model and loop detector data: methodology and large-scale network application," *Transportation Research Record*, vol. 2560, pp. 26–35, 2016.

[9] T. Darwish and K. A. Bakar, "Traffic density estimation in vehicular ad hoc networks: a review," *Ad Hoc Networks*, vol. 24, pp. 337–351, 2015.

[10] J. C. Herrera and A. M. Bayen, *Traffic Flow Reconstruction Using Mobile Sensors And Loop Detector Data*, University of California Transportation Center, 2007.

[11] D. B. Work, S. Blandin, O.-P. Tossavainen, B. Piccoli, and A. M. Bayen, "A traffic model for velocity data assimilation," *Applied Mathematics Research eXpress*, vol. 2010, no. 1, pp. 1–35, 2010.

[12] Y. Yuan, H. Van Lint, F. Van Wageningen-Kessels, and S. Hoogendoorn, "Network-wide traffic state estimation using loop detector and floating car data," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, vol. 18, no. 1, pp. 41–50, 2014.

[13] T. Seo, T. Kusakabe, and Y. Asakura, "Traffic state estimation with the advanced probe vehicles using data assimilation," in *Proceedings of the 18th IEEE International Conference on Intelligent Transportation Systems, ITSC 2015*, pp. 824–830, Spain, September 2015.

[14] V. Astarita, R. L. Bertini, S. d'Elia, and G. Guido, "Motorway traffic parameter estimation from mobile phone counts," *European Journal of Operational Research*, vol. 175, no. 3, pp. 1435–1446, 2006.

[15] N. Bekiaris-Liberis, C. Roncoli, and M. Papageorgiou, "Highway traffic state estimation with mixed connected and conventional vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3484–3497, 2016.

[16] R. Herring, A. Hofleitner, P. Abbeel, and A. Bayen, "Estimating arterial traffic conditions using sparse probe data," in *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems, ITSC 2010*, pp. 929–936, Portugal, September 2010.

[17] J. C. Herrera, D. B. Work, R. Herring, X. Ban, Q. Jacobson, and A. M. Bayen, "Evaluation of traffic data obtained via GPS-enabled mobile phones: the mobile century field experiment,"

Transportation Research, Part C: Emerging Technologies, vol. 18, no. 4, pp. 568–583, 2010.

[18] J. W. C. Van Lint and S. P. Hoogendoorn, "A robust and efficient method for fusing heterogeneous data from traffic sensors on freeways," *Computer-Aided Civil and Infrastructure Engineering*, vol. 25, no. 8, pp. 596–612, 2010.

[19] T. Z. Qiu, X.-Y. Lu, A. H. F. Chow, and S. E. Shladover, "Estimation of freeway traffic density with loop detector and probe vehicle data," *Transportation Research Record*, no. 2178, pp. 21–29, 2010.

[20] Y. Ma, J. Van Dalen, C. De Blois, and L. Kroon, "Estimation of dynamic traffic densities for official statistics," *Transportation Research Record*, vol. 2256, pp. 104–111, 2011.

[21] A. Bhaskar, T. Tsubota, L. M. Kieu, and E. Chung, "Urban traffic state estimation: fusing point and zone based data," *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 120–142, 2014.

[22] J. Zhang, S. He, W. Wang, and F. Zhan, "Accuracy analysis of freeway traffic speed estimation based on the integration of cellular probe system and loop detectors," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, vol. 19, no. 4, pp. 411–426, 2015.

[23] T. Seo, T. Kusakabe, and Y. Asakura, "Estimation of flow and density using probe vehicles with spacing measurement equipment," *Transportation Research Part C: Emerging Technologies*, vol. 53, pp. 134–150, 2015.

[24] L. Montero, M. Pacheco, J. Barceló, S. Homoceanu, and J. Casanovas, "Case study on cooperative car data for estimating traffic states in an urban network," *Transportation Research Record*, vol. 2594, pp. 127–137, 2016.

[25] T. Seo and T. Kusakabe, "Probe vehicle-based traffic state estimation method with spacing information and conservation law," *Transportation Research Part C: Emerging Technologies*, vol. 59, pp. 391–403, 2015.

[26] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO—Simulation of Urban MObility," *International Journal on Advances in Systems and Measurements*, vol. 5, no. 3 & 4, pp. 128–138, 2012.

[27] DLR and contributors, "Sumo homepage," http://sumo.dlr.de/wiki/Main_Page 2016.

[28] S. Krauß, *Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics [Ph.D. thesis]*, Hauptabteilung Mobilitt und Systemtechnik des DLR Köln, 1998.

[29] P. G. Gipps, "A behavioural car-following model for computer simulation," *Transportation Research Part B: Methodological*, vol. 15, no. 2, pp. 105–111, 1981.

[30] D. Krajzewicz, "Traffic simulation with sumo–simulation of urban mobility," in *Fundamentals of Traffic Simulation*, vol. 145, pp. 269–293, Springer, New York, NY, USA, 2010.

[31] G. van Rossum, "Python tutorial, technical report cs-r9526," Tech. Rep., Centrum voor Wiskunde en Informatica (CWI), Amsterdam, Netherlands, May 1995.

[32] G. Evensen, "The ensemble Kalman filter: theoretical formulation and practical implementation," *Ocean Dynamics*, vol. 53, no. 4, pp. 343–367, 2003.

[33] C. Antoniou, M. Ben-Akiva, and H. N. Koutsopoulos, "Nonlinear Kalman filtering algorithms for on-line calibration of dynamic traffic assignment models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 4, pp. 661–670, 2007.