Team Halmstad Approach to Cooperative Driving in the Grand Cooperative Driving Challenge 2016

Maytheewat Aramrattana, Jérôme Detournay, Cristofer Englund, Viktor Frimodig, Oscar Uddman Jansson, Tony Larsson, Wojciech Mostowski, Víctor Díez Rodríguez, Thomas Rosenstatter, and Golam Shahanoor

***Abstract*— This paper is an experience report of team Halmstad from the participation in a competition organised by the i-GAME project, the Grand Cooperative Driving Challenge 2016. The competition was held in Helmond, The Netherlands, during the last weekend of May 2016. We give an overview of our car's control and communication system that was developed for the competition following the requirements and specifications of the i-GAME project. In particular, we describe our implementation of cooperative adaptive cruise control, our solution to the communication and logging requirements, as well as the high level decision making support. For the actual competition we did not manage to completely reach all of the goals set out by the organizers as well as ourselves. However, this did not prevent us from outperforming the competition. Moreover, the competition allowed us to collect data for further evaluation of our solutions to cooperative driving. Thus, we discuss what we believe were the strong points of our system, and discuss post-competition evaluation of the developments that were not fully integrated into our system during competition time.**

***Index Terms*— GCDC 2016, platooning, autonomous driving, cooperative driving, cooperative adaptive cruise control, IEEE 802.11p.**

I. Introduction

In THE European Union (EU), road transportation stood for 75% of the inland goods transportation in 2014. In 2013, passenger cars accounted for 83% of the inland passenger transport. Combustion of fuel used for transport produced 23% of the CO_2 gas emissions in EU during 2014, and road transport accounted for 25.8% of the European energy consumption in 2013 [1]. To reduce the environmental impact and reach the 2°C ceiling target, EU and several governments have clear goals on how to handle this societal challenge and specifically to reduce these emissions, e.g., Sweden has an aim to have a fossil fuel independent transportation sector with the target to reduce fossil fuel consumption by 80% until 2030 [2].

Traffic safety is another large societal challenge. In EU, 28, 000 fatalities were reported in 2012 [1] and traffic accidents are among the ten most common causes of deaths according to the World Health Organization.[1] Worldwide, traffic accidents is the most common cause of death for young people aged 10–24 [3].

Urbanisation and demographical changes are other challenges for the future transportation system. Denser city population and more elderly drivers will further strain the transportation system. Moreover, limited space and high building cost make it difficult to expand roads. Automated and cooperative vehicles are one possible strategy to overcome all these challenges. The roads can be utilised more efficiently, and the environmental impact can be reduced by being able to drive with shorter inter-vehicular distance, which reduces air resistance and consequently lower the energy consumption. Furthermore, by unburdening the drivers with more automated functions the traffic safety will improve.

A. Related Work Within Cooperative and Automated Driving

Early projects within this field of research are, e.g., PROMETHEUS (Program for European Traffic with Highest Efficiency and Unprecedented Safety) [4] that was running between 1988–1995, with the vision to create intelligent vehicles as a part of an overall intelligent road traffic system. Other European projects are, e.g., the CVIS[2] and the SAFESPOT[3] projects. The California PATH (Partners for Advanced Transportation Technology) was initiated in 1986 and is still on-going. PATH pioneered platooning and demonstrated the first Automated Highway System (AHS) in 1994 with a four-car platoon featuring automated longitudinal control [5].

The Safe Road Trains for the Environment (SARTRE) project, running in 2009–2012, was co-funded by the European Commission within the 7[th] Framework Program. SARTRE aimed at developing strategies and technologies to allow platooning within regular public highways to create environmental, safety, and comfort benefits. The SARTRE project

Manuscript received November 28, 2016; revised April 20, 2017, June 22, 2017, and September 1, 2017; accepted September 2, 2017. Date of publication February 8, 2018; date of current version March 28, 2018. The Associate Editor for this paper was E. Semsar-Kazerooni. *(Corresponding author: Wojciech Mostowski.)*

M. Aramrattana is with the School of Information Technology, Halmstad University, SE-301 18 Halmstad, Sweden, and also with The Swedish National Road and Transport Research Institute, SE-402 78 Gothenburg, Sweden.

J. Detournay, V. Frimodig, O. U. Jansson, V. Díez Rodríguez, and G. Shahanoor were with the School of Information Technology, Halmstad University, SE-301 18 Halmstad, Sweden.

C. Englund is with the RISE Viktoria, SE-417 56 Gothenburg, Sweden, and also with the School of Information Technology, Halmstad University, SE-301 18 Halmstad, Sweden.

T. Larsson and W. Mostowski are with the School of Information Technology, Halmstad University, SE-301 18 Halmstad, Sweden (e-mail: wojciech.mostowski@hh.se).

T. Rosenstatter was with the School of Information Technology, Halmstad University, SE-301 18 Halmstad, Sweden, and also with the Information Technology and Systems Management, Salzburg University of Applied Sciences, A-5412 Salzburg, Austria. He is now with Chalmers University, SE-412 96 Gothenburg, Sweden.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2017.2752359

[1] http://www.who.int.

[2] http://www.transport-research.info/project/cooperative-vehicle-infrastructure-systems.

[3] http://www.safespot-eu.org.

demonstrated the benefits of platooning with reported fuel savings of up to 20% for the members of the platoon [6]. Another related FP7 project is AutoNet 2030 [7] which investigated how the heterogeneous fleet of vehicles could cooperate to increase safety and fluidity within traffic. Consequently, the project studied both what information should be exchanged between the different road users and how the road users should be organised (centralised or distributed).

In the same way as levels of automation have been defined, for example by SAE,[4] three dimensions of cooperation in ITS and driving has been proposed [8]. These dimensions are (1) individual, local, or global scope, (2) operational, tactical, or strategical task and (3) two, three, or more actors. Platooning is an example that requires cooperative behaviours in the task (operational, tactical, and strategical), scope (individual, local, and global) and number of actors (2 or more). Cooperative adaptive cruise control and operation concepts [9] are important parts of platooning but it does not cover, e.g., the lane change manoeuvres that are needed.

To perform cooperative behaviour related to the formation, joining, or leaving a platoon on a highway, different types of cooperation, coordination, and agreement protocols have been proposed and evaluated in simulated scenarios [10]–[12]. There is also work related to the higher level issues to find out which vehicles can gain on forming a platoon, taking into account, e.g., that they have similar goals at the more strategic task level [13]. To handle advanced cooperative behaviour several messages need to be exchanged within limited time. This is a serious problem with Inter Vehicle Communication (IVC) [14] and especially in highly congested traffic. Segata *et al.* [15] address these problems by proposing and evaluating a slotted beaconing protocol as a time organised alternative to the ETSI ITS-G5 proposed protocols Decentralized Congestion Control [16] and Dynamic Beaconing [17].

B. Grand Cooperative Driving Challenge

In 2011, the first Grand Cooperative Driving Challenge (GCDC 2011) [18] was arranged. The goal of GCDC 2011 was to accelerate the development, integration, demonstration, and deployment of cooperative mobility. In GCDC 2011, two scenarios were demonstrated, one highway and one urban scenario. In the urban scenario a traffic light-controlled intersection was used to coordinate two platoons in the same lane that were instructed to join after each other. In the highway scenario it was demonstrated how shock waves, that are common on highways, can be attenuated by using cooperative adaptive cruise control supported by V2V communication, i.e. making use of position and speed information similar to the content of a Cooperative Awareness Message (CAM). Furthermore, a dedicated GCDC cooperative interaction protocol was designed to enable execution of the intersection scenario.

GCDC 2016 organised by the i-GAME (Interoperable GCDC AutoMation Experience) project defined new competition scenarios, which besides adding lateral manoeuvres, introduce the most important additional challenge compared to GCDC 2011. That is, to use cooperative lane change messages to handle joint operations among pairs of vehicles driving in two adjacent platoons. It also includes coordinating if and when to alter distances between vehicles and when to change lane in a cooperative manner. The iCLCM (i-GAME Cooperative Lane Change Message) and protocol needed for this were developed by i-GAME before the challenge. Consequently, the challenge was also an essential field test of this approach.

The main challenge in GCDC, both 2011 and 2016, is the multi-vendor approach, where vehicles of different size and brand, developed by different teams at different locations, are going to collaborate and perform cooperative manoeuvres on a real highway at a considerably high speed (80 km/h).

C. Contribution

This paper summarises the system developed by team Halmstad for the GCDC 2016 competition, which builds on the experiences gained in GCDC 2011 [19], and elaborates on distributed vehicle coordination. The competition consists of three scenarios: (1) merging of two platoons on a highway; (2) cooperative intersection crossing; and (3) a demonstration of an intelligent emergency vehicle warning application. All scenarios are enabled by distributed negotiation where vehicles communicate to coordinate with each other. A brief description of the GCDC scenarios is given in Sect. II, a more detailed description of the scenarios and GCDC can be found in [20]. In this paper, we describe our implementation of cooperative adaptive cruise control, our solution to the communication and logging requirements, as well as the high level decision making support. Due to space restrictions and the intended character of this paper, the solutions are described with a varying level of detail. Consequently, our communication and trust system solutions are discussed in greater detail in two accompanying publications [21], [22].

One of the approaches developing our system for the competition was to use, where possible, cost efficient hardware. The paper describes the associated challenges, most vividly in communication, and how they were addressed. In this context, the competition was a source of real data, which was used to further develop and evaluate our ideas and solutions.

D. Paper Organisation

The rest of the paper is organised as follows; Sect. II gives an overview of the GCDC 2016 scenarios. Section III describes the experimental setup of the vehicle and the architecture of the developed system. The vehicle control system is presented in Sect. IV. Section V describes the V2V communication module. Sections VI–VIII briefly describe the high-level control, decision making, and the perception and sensor-fusion module. In Sect. IX the preparatory simulations and work are described whereas in Sect. X results from post-competition analysis are presented. Finally, Sect. XI concludes the paper and suggests directions for future work.

II. SCENARIOS

Besides the overall goal of GCDC – to boost the introduction of cooperative and automated driving – the scenarios in

[4]http://standards.sae.org/j3016_201609/.

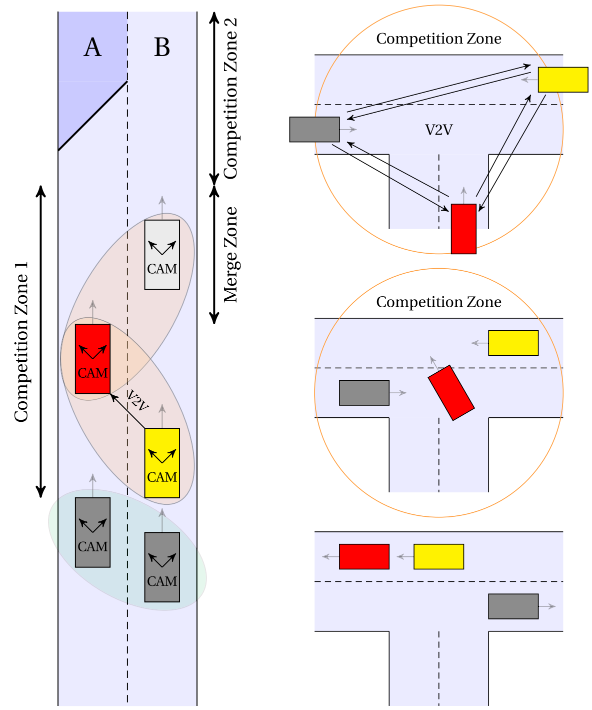


Fig. 1. The schematics of the two judged competition scenarios.

GCDC are designed to also demonstrate the current development within cooperative intelligent transport systems (C-ITS). The scenarios are influenced by suggestions from domain experts as well as proposals from the participating teams.

The first scenario, shown in Fig. 1 on the left, is the cooperative platoon merge. It involves two platoons driving on two adjacent lanes on a highway. The two platoons must merge into one due to an upcoming construction site where one lane is closed. A competition zone is defined as a zone where the vehicles' operations are judged.

The second scenario, the cooperative intersection shown in Fig. 1 on the right, considers a common urban traffic situation, an uncontrolled T-intersection. Whereas the first scenario requires that all vehicles are interacting and communicating, this scenario involves a mixture of non-cooperative (not communicating) and cooperative vehicles. The scenario involves three cooperative vehicles, one vehicle that is approaching a busy road with two other vehicles driving in both directions. The approaching vehicle transmits its intention, to turn left in the intersection, and the cooperative vehicles on the main road acknowledge its request and help to facilitate the manoeuvre in an efficient manner, i.e., without coming to a full stop. Consequently, the vehicles on the main road help to create proper gaps, allowing a smooth passage for the left-turning vehicle to safely and efficiently cross the intersection. The non-communicating vehicles take only an assumed part in the scenario, virtually following the communicating vehicles on the main road, i.e., only the three mentioned communicating vehicles take part in the challenge scenario.

The third scenario demonstrates an emergency vehicle requiring passage along a highway with congested traffic. This scenario is not part of the judging, yet it is used to demonstrate an everyday traffic situation that needs efficient solutions.

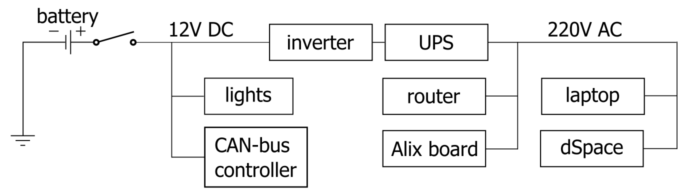


Fig. 2. Power supply setup for the system.

Emergency vehicle warning has been considered in the basic set of applications of the C-ITS standard, see e.g. [23] for further details. Since the current version of the emergency vehicle warning only provides a warning about an approaching emergency vehicle it is still confusing for the other road users about where to place their vehicle. With the proposed amendments in GCDC, the emergency vehicle will be able to inform other vehicles of its itinerary and how it wants other vehicles to behave, thereby providing a safe passage.

III. System Architecture

Both on the software and hardware side it was decided to go for a relatively simple solution, for two reasons. First, the general i-GAME concept is to provide robust future solutions for the automotive industry and such a solution fits better with these conceptual demands of the GCDC context. Second, the former Halmstad Team from GCDC 2011 [19] achieved a very good result with a similar setup, thus it was decided to work based on their good experiences.

The base vehicle was a production Volvo S60. The additional control, sensor, communication, and supporting devices mounted in the vehicle (mostly in the trunk) were the dSpace MicroAutoBox (MAB) real-time controller,[5] Trimble differential GPS, a radio computer (see Sect. V), a Nexus 9 tablet, and a router. The only gateway to the S60 systems was through the dSpace MAB that intercepts car's CAN bus messages and is able to inject additional ones. No other devices were used, in particular the only sensors used in our system were the the radar system of the vehicle monitored through MAB and the Trimble GPS. Finally, the high-level control and coordination of the system was done with a regular laptop, see below.

A. Power and CAN Bus Arrangement

The power supply arrangement is shown in Fig. 2. The main power source was the car battery of the S60. To operate the 220V equipment a DC to AC inverter was used and to avoid power failures an UPS (Uninterrupted Power Supply) was installed. The 12V equipment such as the CAN bus interface and mode signalling roof lights were powered from the battery. To disconnect the power supply in an unexpected situation the main power bus was controlled by a driver-side switch. The Trimble GPS has its own battery, thus no external power was needed other than periodically charging the battery.

A custom made CAN bus interface depicted in Fig. 3 was used to switch between automatic and manual driving mode. The emergency button had the highest priority on the bus.

[5] https://www.dspace.com/en/pub/home/products/hw/micautob.cfm.

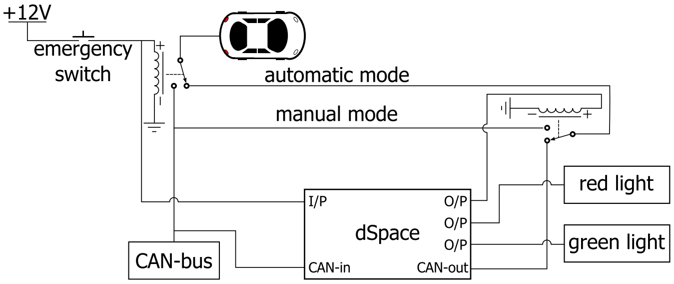


Fig. 3. CAN bus interface diagram.

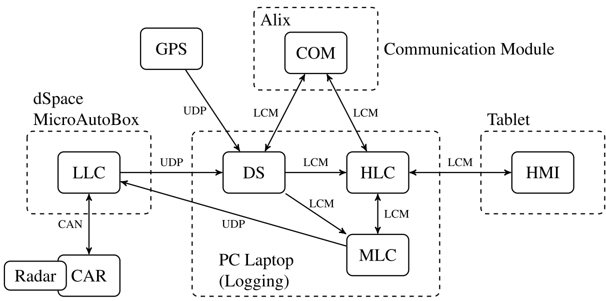


Fig. 4. Software system architecture.

Automatic to manual mode transition was also controllable from the MAB, but with lower priority than the button.

Otherwise, all the devices comprising the system were interconnected through an Ethernet router with cables, with the exception of the Human-Machine Interface tablet that was connected to the car network through wireless communication.

B. Software Modules

The software architecture of the system is characterised by its modularity. The system is split into modules as shown in Fig. 4. Each module can be executed independently and on different hardware units. They communicate using Lightweight Communications and Marshalling (LCM) [24], which provides a programming language agnostic solution to communication that abstracts from the actual media, in our case UDP packets in the local network.

The communication (COM), data source (DS), high-level control (HLC), and mid-level control (MLC) modules are implemented as Java applications running on a regular, non-real time Java Virtual Machine:

- COM implements the ITS-G5 communication stack and services used to send and receive V2V messages;
- DS performs sensor fusion with the information received from the COM module and LLC module (see below) directly from the car and the GPS, and provides this information to the other modules;
- HLC makes high-level decisions regarding manoeuvres following the competition interaction protocols;
- MLC calculates parameters for the LLC speed controller.
- LLC is the speed controller, described in the next section.

The low-level control (LLC) is a Simulink model executed on the dSPACE MicroAutoBox connected to the CAN bus of the car. This controller fully replaces (bypasses) the factory adaptive cruise control system. Finally, the human-machine interface (HMI) Java application runs on an Android tablet. It collects the driver's input regarding configuration for the scenario and asks for confirmation before performing manoeuvres autonomously. It also provides information regarding the status of the scenario and the neighbouring vehicles.

IV. Cooperative Adaptive Cruise Control

A. Controller Design

The proposed control system strives to utilise the full potential of cooperative driving. It is inspired by the previous Halmstad team solution [19] and [25]. To achieve robustness and modularity, the controller is divided into two layers, mid-level control (MLC) and low-level control (LLC). The MLC communicates with all other processes, determines maximum speed, desired time headway and gathers preceding vehicle information from the sensor fusion module (DS). The LLC keeps the desired distance from the preceding vehicle, and maintains all the constraints provided by the MLC. The LLC strategy is explained below, and its overview is shown in Fig 5.

The constant time gap (CTG) policy [9] is chosen as the gap regulation policy. According to [9], the time gap, referred to as a *time headway* in this paper, is defined as *"the time between when the rear bumper of the leading vehicle and the front bumper of the following vehicle pass a fixed location on the roadway (measured in seconds)"*. Therefore, the desired inter-vehicular distance for the i^{th} vehicle in a platoon is proportional to its speed, plus a fixed offset (standstill) distance. The desired distance is calculated by eq. (1):

$$d_{i,\text{des}}(t) = d_{\min} + h_i \cdot v_i(t) \tag{1}$$

where $d_{i,\text{des}}(t)$ is the desired distance (m), $d_{\min}$ is the standstill distance (m), h_i is the time headway (s), and $v_i(t)$ is the vehicle speed (m/s). During GCDC h_i was specified to be 1 s and $d_{\min}$ to be 6 m.

The proposed system is composed of three main controllers:

- C_1, a proportional controller with a lead compensator that acts on speed error ε_i,
- C_2, a proportional-integral controller that acts on distance error δ_i,
- C_3 applies a gain on acceleration from the preceding car as reported through CAM messages.

Since C_1 considers output from C_2, let us first discuss C_2. The distance error δ_i is defined as:

$$\delta_i(t) = d_{i,\text{act}}(t) - d_{i,\text{des}}(t) \tag{2}$$

$$\delta_i(t) = S_i(t) - S_{i-1}(t) - l_{i-1} - (d_{\min} + h_i \cdot v_i(t)) \tag{3}$$

where S_i represents position of the i^{th} vehicle, and l_{i-1} is the length of the preceding vehicle. The control law for C_2 is:

$$v_{i,\text{des}}(t) = K_{P2} \cdot \delta_i(t) + K_{I2} \int_0^t \delta_i(t)\, dt \tag{4}$$

where $v_{i,\text{des}}$ is the desired speed, $K_{P2} = 2.9497$, and $K_{I2} = 4.3615$ (the gain parameters where chosen experimentally, see below).

The controller C_1 then acts on the speed error, which is:

$$\varepsilon_i(t) = v_{i,\text{act}}(t) - v_{i,\text{des}}(t) = v_{i-1}(t) - v_i(t) - v_{i,\text{des}}(t) \tag{5}$$

where $v_{i,\text{des}}$ is calculated in eq. (4). C_1 is a proportional controller with a lead compensator given by:

$$a_{i,\text{des}} = K_{P1}\big(\varepsilon_i(t) - 7.5e^{-10t}\big) \quad \text{where } K_{P1} = 0.872 \tag{6}$$

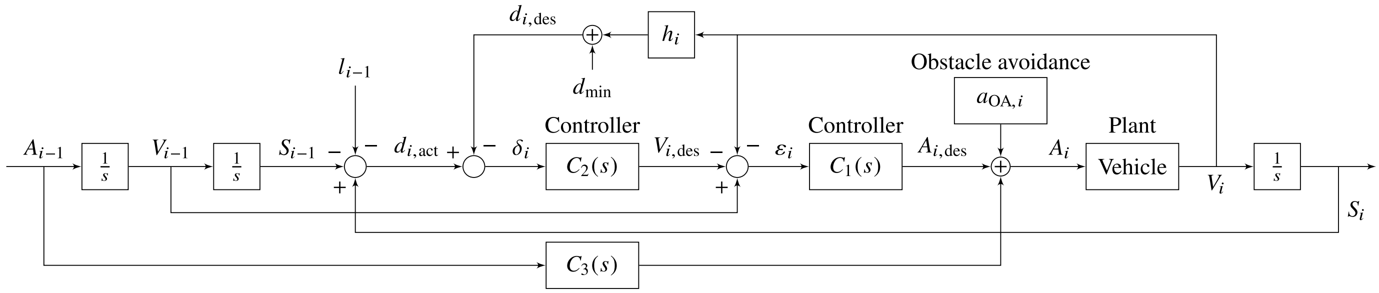


Fig. 5. Overview of the low-level control (LLC) system. $C_1(s)$ is a speed controller, it keeps the speed of the previous vehicle and indirectly regulates the distance. $C_2(s)$ regulates the distance to the vehicle in-front. $C_3(s)$ manipulates the feed-forwarded acceleration A_{i-1}, from the preceding vehicle.

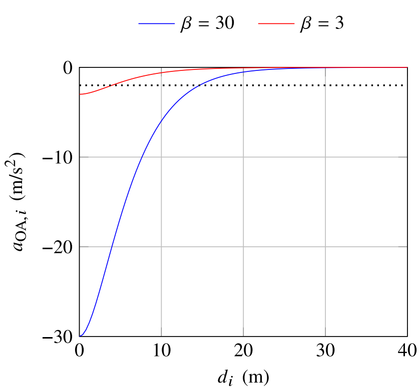


Fig. 6. Plot for OA with different β with black dotted line showing maximum (cut-off) deceleration of $-2\,\text{m/s}^2$.

As part of a safety feature, the Obstacle Avoidance (OA) controller with a potential function according to eq. (7) is used to increase deceleration, in case of the preceding vehicle instantaneously applies high deceleration:

$$a_{\text{OA},i} = \begin{cases} -\beta(\alpha d_i + 1)e^{-\alpha d_i} & a_{i-1} < 0 \text{ and } d_i < d_{i,\text{des}} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where β is a gain factor which indicates the maximum effort of the controller when the distance goes to zero, α is a fall-off rate when the preceding vehicle is getting away, and d_i is the distance to the preceding vehicle. This equation is similar to the OA controller stipulated by the competition organisers [26]. However, apart from activating when the preceding vehicle is decelerating (as in [26]), an added condition is when the actual inter vehicular distance is shorter than the desired distance. Therefore, the OA is applied to facilitate the braking only when these two conditions are true. Because of this, we also decided to amplify the effort of the OA once it engages, the particular parameters we used were $\alpha = 0.3$ and $\beta = 30$, while the organisers proposed $\beta = 3$. Figure 6 shows the comparison of characteristics of the OA function with our β control parameter and the suggested one. Therefore, the complete acceleration input to the plant is formulated as:

$$a_i = a_{i,\text{des}} + a_{\text{OA}}, i + K_{P3}a_{i-1} \quad -2 \le a_i \le 2 \tag{8}$$

where $K_{P3} = 0.4981$, and the final value a_i is bounded by the maximum acceleration and deceleration, which is -2 to $2\,\text{m/s}^2$ according to the GCDC rules [27].

B. Controller Evaluation

During the development phase of the CACC controller the performance was evaluated using Matlab simulations according to the following criteria:

- Performance: to what extent the system is capable of keeping the desired distance to the preceding vehicle and does the system maintain the string stability condition;
- Safety: the system is considered as safe if the actual distance is larger or equal to the desired distance:

$$\begin{aligned} d_{i,\text{act}} &\ge d_{i,\text{des}} && \text{safe,} \\ d_{i,\text{act}} &< d_{i,\text{des}} && \text{unsafe,} \\ d_{i,\text{act}} &< d_{\min} && \text{risk of collision.} \end{aligned}$$

- Comfort: the smoothness of the controller is measured by the jerk effect, which ideally should be zero: $\ddot{a}(t) = 0$.

It was only after the competition that we were able to evaluate the controller in a realistic setting with the actual competition heats data. This is further discussed in Sect. X.

In theory, according to the final value theorem, using just the proportional controllers would be sufficient, i.e., the errors are eventually brought down to zero in steady state conditions. Actual experiments exhibited distance lagging and prompted the introduction of the integral component in C_2 to decrease the reaction time on δ_i. A derivative component would dampen the behaviour and provide smoothness, however, in the rather steady state conditions of GCDC it was not necessary and the tedious tuning of the derivative gain was avoided.

The controller gain parameters $K_{\{P1,P2,I2,P3\}}$ were tuned first approximately with the SISOTOOL from Matlab, and then by experimentation with the organiser team during the competition preparation week. The distinguished feature of the proposed control strategy is the feed forwarding of the acceleration of the preceding vehicle A_{i-1} obtained from the MLC module via CAM messages. This is the point where the cooperative character of the controller is exhibited. The acceleration is manipulated with the gain in C_3, and feed-forwarded to the vehicle. The controller also has the option to use *intended* acceleration of the preceding vehicle (if available) rather than the actual. During brief GCDC off-line experiments the use of the intended acceleration was successfully added to the controller. However, during competition heats the intended acceleration of participants was often either unavailable or faulty, hence the actual acceleration was used during the competition to achieve robustness. Our experimentation result from using the intended acceleration as well as a brief evaluation of the controller are further discussed in Sect. X.

V. Communication and Logging Modules

A. Communication

The communication system for V2X is composed of: (1) a radio module to handle the physical and data link layer of the communication stack; and (2) a computer to handle the rest of the layers, from network to application layer.

The hardware used for the radio module is a Wistron DCMA82 with an Atheros AR922X chipset attched to an ALIX 2D13 system board, containing 256 MB of RAM and a AMD Geode LX800 processor at 500 MHz. The decision to use this particular board was purely pragmatic – during a communication workshop organised in Sweden two other teams reported it to be capable of meeting the competition requirements. These two teams (from Chalmers) were also geographically nearest which enabled mutual pre-competition testing and support. The computer module used for the upper layers features 8 Gb of RAM and an Intel Core i5-5300U at 2.3 GHz. Besides running the communication module, this computer also executes most of the other components in the system, as illustrated in Fig. 4, and is connected to the rest of the system via Ethernet, using the User Datagram Protocol (UDP) to Ethernet conversion daemon (*udp2eth*).[6]

According to the Open Systems Interconnection (OSI) model, the physical and data link layers are implemented as a modification of the *ath9k* Linux kernel driver, which can be found on GitHub.[7] The modified drivers were loaded in Voyage Linux,[8] a lightweight Debian-based Linux distribution. The remaining layers (network, transport, session, presentation, and application) are encapsulated in a Java application. The system uses the GeoNetworking and ASN.1 UPER encoder/decoder implementation by Voronov *et al.* [28].

The communication module works independently and transparently to the rest of the system. It decodes and relays information to and from the other modules, and transmits information as V2V messages of the following three types:

- Cooperative Awareness Message (CAM), containing vehicle status information such as position, movement, and other sensor data [29];
- i-GAME Cooperative Lane Change Message (iCLCM), containing information required to perform manoeuvres during the competition [30]. Similarly to CAM, iCLCM-s are sent periodically and contain scenario control flags for the execution of the competition (e.g., start of scenario), platooning information (platoon identifier, desired acceleration), merge scenario information (merge requests and confirmations, pairing arrangements), and intersection scenario information (vehicle identifier and intention);
- Decentralised Environmental Notification Message (DENM), used to notify other users of events such as dangerous road conditions and emergency situations [31].

[6]https://github.com/jandejongh/udp2eth.
[7]https://github.com/CTU-IIG/802.11p-linux.
[8]http://linux.voyage.hk.

The communication module interacts with the system via LCM messages. Two LCM channels (input and output) are used for each type of V2V message. Whenever a module wants to transmit a message, it sends the information on the output LCM channel for that specific message type, CAM, DENM, or iCLCM. Similarly, if a module wants to obtain the information from a given type of message, it can listen to the corresponding LCM input channel. The data types used in these LCM channels contain the same fields and use the same units, as defined in the European Telecommunications Standards Institute (ETSI) standards for CAM and DENM, and as defined by the i-GAME project for iCLCM.

A container class with the latest information required to construct CAM and iCLCM messages is stored in the communication module and updated when new information is received via LCM. For GCDC 2016, the organisers defined the update frequency of CAM and iCLCM to be 25 Hz. Therefore, every 40 ms this information is used to construct the messages, encode them and send them via a Basic Transport Protocol (BTP) socket. Although CAMs and iCLCMs are sent with the same frequency, they are generated with an offset of half a period, i.e. 20 ms, in order to spread the computation load over time and avoid peaks when the messages are generated.

A separate thread receives the messages. Packets are extracted from the BTP socket, decoded, and equivalent LCM messages are created that are broadcast inside the system by placing them in the corresponding LCM channel, through which listening modules receive their own message copies.

No particular signal quality was required by the organisers (e.g., in terms of signal to noise ratio), apart from the frequencies mentioned above and the bi-directional communication distance of 200 m when no obstacles are present [32]. During preparatory tests with the other teams the communication range was verified to be at just this distance in plain sight using small antennas. At the competition site, even using large antennas, the communication was occasionally disrupted at one particular spot during the merge scenario due to a bridge and by the presence of tall vehicles in the platoon. However, at that point no ad-hoc solution could be provided by any of the teams to improve the communication quality, apart from vehicle control fall-back procedures. As for the throughput, during the competition it was necessary to process communication from only 10 participating vehicles, hence a simple message dispatching system based on FIFO queues was sufficient to manage the communication with the required frequency. In a more realistic scenario FIFO processing would not be sufficient and possibly cause message buffering congestion, thus after the competition we developed a message prioritising and filtering system [21], outlined in Sect. X.

In general, the ITS-G5 standard itself does not require packet delivery guarantees, it is the responsibility of the higher level application (in this case the GCDC iCLCM interaction protocol [30], [33]) to provide safe communication. In particular, low-level packets are not acknowledged, thus it is not possible to state how many of our packets were received by the participants at all times. However, other competition safety requirements were related to the accuracy and delay of the transmitted positioning and velocity data, see Sect. VII.

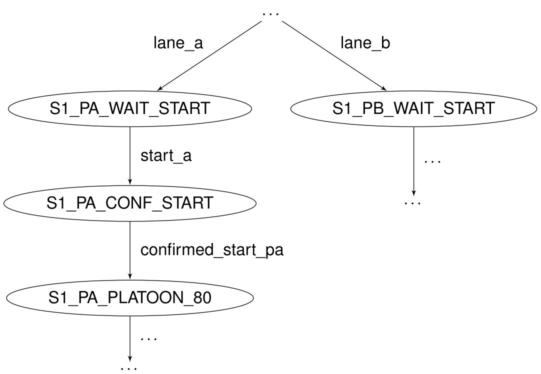


Fig. 7. Sample state transitions from [33].

B. Logging

The logging facility of the system works besides the communication module and is divided into three parts. The first part, implemented as a Java application, keeps track of selected fields from the CAM, DENM, and iCLCM messages. When a message is received, the fields of interest (ones required by the competition requirements, e.g., current position, speed, etc.) are written into a Comma Separated Values (CSV) file. Each line corresponds to one message and it is timestamped with the ETSI Timestamp [34]. The received and transmitted data were stored in separate files and conveyed to the GCDC organisers for judging purposes.

The second part of the logging is done by Wireshark, a network protocol analyser [35] capturing all network traffic and saving it in a packet capture (pcap) file. These files were used after the competition to analyse the performance of the communication protocol, see Sect. X.

The last part of the logging system records the LCM traffic, all input and output LCM messages between the different modules of the system are logged. These logs can be replayed to analyse the behaviour of the system off-line.

VI. High-Level System Control

The HLC's decision making is implemented as a finite state machine (FSM) that directly follows the GCDC interaction protocol specification predefined by the organisers in [12] and [33]. The events for triggering the state transitions of the FSM are depending on the interaction with the other vehicles, the confirmations via the HMI, and the internal state of the vehicle control system. Confirmations of the driver have been included for safety reasons and to increase the driver's situation awareness in an automated vehicle. An example of a state transition shown in Fig. 7 is when the system is in the S1_PA_WAIT_START state – the ready vehicle is waiting at standstill – and the message with a startScenario is received from the RSU triggering the transition start_a to launch the vehicle. For safety reasons (the vehicle is about to move by itself), the system goes into an intermediate confirmation state S1_PA_CONF_START that sends a request to the HMI and waits for the driver's response before going into the actual vehicle platooning state S1_PA_PLATOON_80. The confirmation steps are skipped for less critical transitions, or when the driver causes an unacceptable delay, most notably during the execution of the intersection scenario.

The state transition rules are part of the *World* – a processing context where the information about the surrounding environment and its state is kept. Information about every vehicle sending CAM or iCLCM message is put into the *World*. This data is fused into one object describing the vehicle within the perception module (within Data Source – DS, see Fig. 4) that was developed by the team for the competition to support the HLC. Furthermore, the concept of a Trust System (TS) was developed to support decision making in an *untrusted* environment. We describe both in the following sections.

VII. Perception and Sensor Fusion

The high-level controller needs the knowledge about the surrounding environment. The vehicle perceives this environment with the built-in front radar detecting a single target in front of the vehicle, the RTK-GPS device, and the information provided via V2V communication. To build the knowledge about the environment two models are used – vehicle distance model (VDM) and vehicle position model (VPM).

A. Vehicle Distance Model

The VDM was designed to describe the relation between the measured radar distance to the car in front and the calculated distance by using the geographical position of the ego (from GPS) and the preceding vehicle (from V2V). The model applies a Kalman Filter, which is proposed in [36], to the distance to the preceding vehicle in case that both distances match each other given a certain boundary (1 m). When the difference between these two ranges exceeds the predefined threshold, the distance measured by the radar takes precedence and it is broadcast to the other modules of the control system. This approach ensures, for safety reasons, that the vehicle's controller uses the radar information about the closest physical obstacle in front of the ego vehicle when wrong information is received via V2V message exchange.

The VDM has been evaluated against the competition requirements by relative comparison with the organisers' equipment in their reference vehicles. That is, by following each other in a platoon, the reported and measured data was verified by the organisers to stay within pre-defined tolerances [27], [32]: 1 m for the position, 0.5 m/s for velocity, 0.2 m/s^2 for acceleration and deceleration, and 200 ms for communication latency (time from data readout to reception by another vehicle). The accuracy of our own sensors were 1 cm for the RTK-GPS when stationary, and 0.25 m for the radar (for the GCDC applicable short range distance of up to 60 m). Our VDM calculations have satisfied these bounds, however, no precise error measurements were reported to us to evaluate how well we stayed within these bounds. Lacking a ground canonical reference for vehicle positions, this was the only applicable verification method.

B. Vehicle Position Model

The VPM that applies an Extended Kalman Filter (EKF) to the vehicle's position and inertial sensor information to

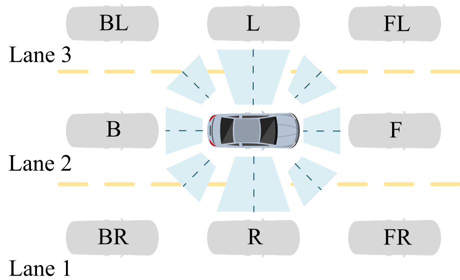


Fig. 8. Illustration of the relative positions in the map.

improve its geographical position was designed according to [37]. This position model is also applied to the other vehicles based on their transmitted information. An EKF is split into two updates, the time update (prediction) and the measurement update (correction). At first a state vector, which consists of the east and north coordinates, heading to the north, velocity, yaw rate, and acceleration of the vehicle, has been defined. A motion model based on a simple bicycle model [38] is used to predict the new state during the time update. The computation of the Kalman gain and the update of the state using new measurements is performed during the correction phase. The measurement noise covariance matrix describing the error of the measurements is set depending on the source of the information, e.g., inertial sensors and GPS position or V2V information. Due to computational reasons, VPM is applied only to vehicles identified as important, such as the preceding vehicle. Moreover, the Kalman gain of the VPM has been used as an indicator for sensor accuracy [37], [38].

The surrounding vehicles are identified according to their relative position, e.g., *front-left*, *in front of*, or *behind* the ego vehicle, and put into a map categorised by these relative positions. An illustration of the map is depicted in Fig. 8. This map is broadcast to all other control modules. As the perception of the competition car is limited, in particular in tighter curves, the map is generated by applying two different neighbour identification methods that evaluate the received CAM and iCLCM messages. The results of both methods are combined in order to provide a robust identification of the surrounding vehicles.

The first method sorts the vehicles according to their driving direction and platoon identifier from iCLCM. The relative position category is classified based on predefined angles and the distance to the ego vehicle. The second method discards the platoon identifier and relies only on the relative angles and the distance. The range of the angles for each category is calculated dynamically for each vehicle taking the distance to the vehicle and its dimension into account.

Each method puts the vehicle with the shortest distance in each category in a separate map. When combining the two maps, the first one where the platoon identifier was used is prioritised, since it is more robust in curves. When there is no vehicle match for the given category, e.g. *front-right*, in the first map, the vehicle possibly identified using the second method is assigned to this category in the combined map. Combining the results this way provides a robust identification of vehicles even if the platoon identifier is missing. Admittedly, it is possible for this procedure to identify one vehicle in two different categories, each resulting from the corresponding map. For example, the first map identifies the vehicle as the *in-front* vehicle and the second map identifies it as the *in the front-left* vehicle. This possibility only occurs when the first map has no vehicle assigned to the given category, *in the front-left* in this case. However, this has no negative impact on the system and it is safe – it is a clear case of a false positive when an existing vehicle is reported in an additional position, while a false negative of not reporting an existing vehicle in any of the positions would be far more dangerous.

The necessity to develop this two-stage identification system became apparent during testing at the Film and Test Location (FTL GmbH)[9] in Aachen, Germany, and at the Dutch Road Transport Authority (RDW) test track in Lelystad. Both locations have considerably tighter curves than the competition zone and hence exhibiting frequent classification errors when using just one method. After introducing the two-stage method and after these tests we had no more opportunities to verify it in curvy road conditions, only at the competition zone in Helmond, which was comparatively straight providing ideal conditions and at which, by visual inspection, the method provided practical 100% robustness. However, more experimentation would be required to further evaluate the method and the choice of control parameters in other conditions.

VIII. Trust System

The concept of the Trust System (TS) is to evaluate the current situation based on the ego vehicle's and the other vehicle's trust as well as the trust in the environment. This information is represented as one scalar value, the Trust Index (TI), and can be used by the decision making algorithm to make more robust decisions. For instance, the TI can be used by the decision making algorithm to decide on the distance to the vehicle in front when driving in a platoon.

The TS generates and combines partial TI-s for the sensor quality of the ego vehicle, the sensor quality and behaviour of the other vehicles, especially the preceding vehicle and forward partner, and the environment. The quality of a sensor reflects the sensor's precision and reliability. The TS applies the position model (see Sect. VII) to important vehicles and additionally applies the distance model to the preceding vehicle, for the reason that the location of this vehicle can be verified with one of the ego vehicles own sensors (the radar).

Further details about the perception module and the TS, including the necessary formulas and parameters, can be found in [22] and [39]. The concept of the TS as well as a prototype have been used during the GCDC. The final TS has been evaluated with the communication data gathered during the GCDC highway scenario heats. The sensor fusion as well as the TS are executed independently within the DS module (see Fig. 4). For reasons to be stated in Sect. X-C the TI-s have not been considered by the HLC during the competition.

IX. Preparations and Competition Performance

Despite starting the project early enough (the team was fully formed by the end of September 2015) and acquiring

[9]http://ftl-germany.com.

the competition vehicle (December 2015), large efforts were put in during several months to enable and stabilise the CAN interface between the dSPACE MicroAutoBox and the car. In effect, the autonomous control of the car was only available six weeks before the competition, yet with continuing stability issues. Furthermore, only two weeks before the competition the interface to the radar module in the car was functioning. Because of these factors, no attempt has been made to support automatic lateral control of the vehicle, despite the fact that the car does offer limited control of the steering from the factory lane-assist system. Yet, the modularisation of the system allowed us to work on the other parts of the system in parallel to the car interfacing work. In particular, extensive use of PreScan simulation software[10] [40] enabled testing the system without the car or its CAN bus interface.

A. PreScan Simulations

During development, the complete system was tested by running simulations. PreScan was used to generate simulations that could provide realistic car information to the system, such as GPS position and radar information. UDP sockets were used to communicate between the simulation environment and the running system. Simulations were useful to test individual modules that use raw information originating outside of the system, e.g., the communication packets. Similar simulations were also used to test the interaction among several modules. For example, the use of the CACC with the processed information provided by the communication module.

Vehicle-in-the-loop tests were executed by using data provided by simulated vehicles. This method was particularly useful to test and evaluate the CACC controller. Using a simulated vehicle as the preceding vehicle for the CACC evaluation, eliminated the risk of accident if a real car had been used and a serious error in the controller had occurred.

Lastly, complex simulations involving several vehicles were executed to test the interaction protocols for the highway and intersection scenarios. Multiple instances of the system were executed simultaneously, each one controlling a different simulated vehicle.

B. Competition Time

The system was tested and modified at the Automotive Campus in Helmond during the competition preparatory week of May 20–27 2016. In particular, we developed the two-stage vehicle identification on spot, see Sect. VII. During the actual competition only minor changes were made. Apart from pre-testing with the Swedish teams at AstaZero[11] in late April 2016, the preparation week was the first time when the interaction and communication could be tested against complete systems from other teams. During that time we could fine tune the CACC controller and fix other encountered issues, however, not all of them, like the communication issues suffered by all teams that were caused by the local surroundings, see Sect. V.

[10]https://www.tassinternational.com/prescan.

[11]http://www.astazero.com.

X. Post-Competition Evaluation

The main ideas and solutions for the three building blocks of the proposed system have been developed, evaluated and tested using Matlab and PreScan simulations, and by single try-outs with other teams during the pre-competition meetings at the IDIADA testing ground[12] in Spain (4 days) or at the AstaZero test track in Sweden (3 days). During this mutual testing, none of the competing systems were fully developed. It was only at the actual competition week that enough data was collected from sufficiently working systems to evaluate the solutions in a realistic setting. Concretely, the following evaluation aspects were considered:

- practical performance of the proposed CACC implementation in terms of string stability, safety, and comfort,
- robustness of the radio communication implementation against the demands of communicating with 9 other teams without (major) disruptions,
- the Trust Index distribution of TS in a realistic setting.

Apart from collecting the heat logs during the actual competition, the preparatory week was also used to do trial with other teams for experimentation with new features. In particular, with the A-Team from Technical University Eindhoven, the CACC controller was tested based on the intended acceleration as input rather than the actual acceleration.

A. CACC Performance

The main concern in evaluating a speed controller is the string stability [41]. As noted in [42], due to the number of control parameters, a proper analysis of string stability for a CACC controller is practically difficult. The particular transfer function of our controller, under simplifying assumptions (no actuation or communication delays, $d_{\min} = 0$, and $l_{i-1} = 0$) is given by eq. (9):

$$\frac{V_i(s)}{V_{i-1}(s)} = \frac{C_3(s) \cdot s^2 + C_1(s) \cdot s + C_1(s)C_2(s)}{s^2 + (1 + C_2(s)h_i)C_1(s) \cdot s + C_1(s)C_2(s)} \quad (9)$$

Thus, similarly to [42], our CACC controller has been evaluated using Matlab simulations where an arbitrary number of vehicles with different test parameters could be tested. With these simulations, our controller already proved to perform slightly better than the Sliding Mode Algorithm (SMA) [43]. The simulations were performed against an assumed simple vehicle model which is parametrised by an engine time constant (vehicle's reaction time to acceleration and braking) and sensor (communication) delay, moreover, the obstacle avoidance function was not considered. As an example, Fig. 9 shows a comparison of distance error propagation between our controller (top) and an instance of the SMA controller (bottom) under fixed headway time of 1 s, fixed sensor delay of 0.1 s, and a varying engine time constant of eight vehicles between 0.2 and 0.6 s, i.e., in a simplistically heterogeneous platoon. Both algorithms are string stable in this case (the distance error does not amplify as it propagates backwards to the following vehicle), however, our controller shows a significantly smaller amplitude of the distance error (max. $\approx$ 0.2 m) compared

[12]http://www.applusidiada.com.

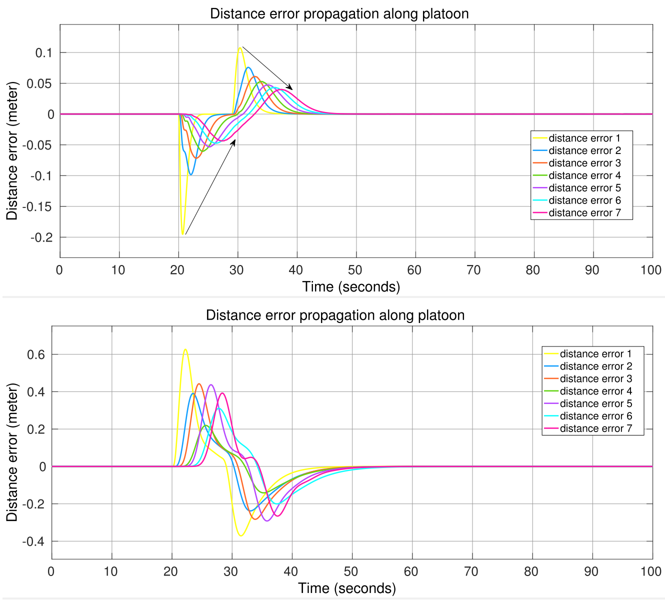


Fig. 9. Distance error propagation comparison to the SMA controller in a heterogeneous platoon simulation.

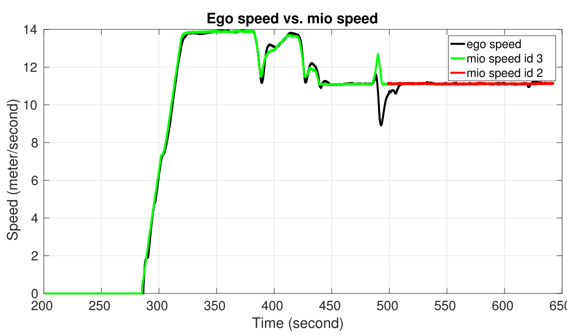


Fig. 10. CACC vehicle speeds during the merging heat.

to SMA (max. ≈ 0.6 m) as well as quicker convergence over time. Considering the communicated acceleration feed-forward capability of our controller that SMA naturally lacks, this advantage is rather not surprising, moreover, given a simplistic vehicle model and idealised simulation scenario, hardly conclusive for a realistic setting.

The competition data provided a more realistic evaluation target, however, it was only possible to evaluate the CACC controller against just one vehicle, the currently preceding vehicle (most important object – MIO) in the platoon. The fact that the preceding vehicle is changing during the highway merging scenario from the initially followed vehicle to the merging one essentially introduces additional dimension to the test, as effectively the speed to follow momentarily changes in a non-continuous fashion, while in simulations only an ideal platooning scenario was considered.

Figure 10 shows the relationship of speeds between three vehicles merging during one of the slow-speed heats, with the vehicle arrangement shown in Fig. 11. The Halmstad vehicle followed the OPC2 (Organiser Pace Vehicle) reference vehicle (id3) and made gap for OPC1, vehicle (id2), to merge in at around timestamp of 540 to 620 s. The drop in speed of the ego vehicle at ≈500 s is obviously intentional to make the

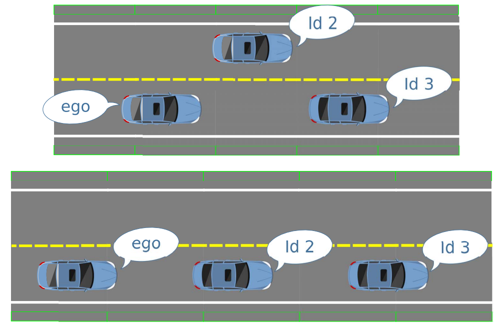


Fig. 11. Vehicle arrangement during the merging heat.

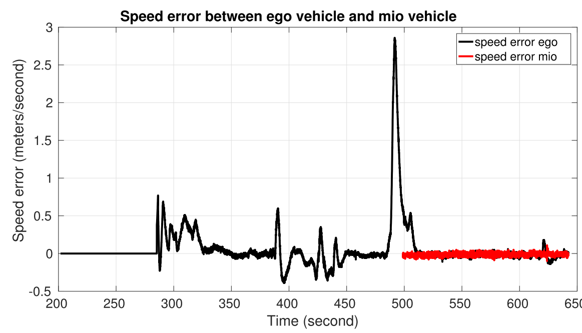


Fig. 12. Speed error for the speeds presented in Fig. 10.

gap for id2, after which the ego vehicle immediately follows the new vehicle id2. The corresponding speed error to the preceding vehicle is shown in Fig. 12. Apart from gap-making, the error never exceeds 1 m/s and stays within 0.5 m/s margin most of the time, while it is practically at 0 in stable speed conditions. Moreover, the recorded jerk of the ego vehicle during this heat was within the 0.3 m/s^3. Further details and evaluation of the CACC controller can be found in [44].

Apart from evaluating the base version of the CACC controller with the competition heats, a brief experimentation with the TU/e's A-team during the preparatory week was also performed to preliminarily evaluate the controller that uses the communicated intended acceleration from the MIO rather than the actual acceleration. This was done with the hope that the lag caused by vehicles' dead time could be further reduced. That is, in practice the vehicles should be able to synchronise better on mutual acceleration. Figure 13 shows a 1 minute snapshot of the resulting accelerations of the two vehicles and the recorded distance. This single experiment is not sufficient to evaluate this approach (in particular, w.r.t. string stability), nevertheless, the graphs further visualise the behaviour of our CACC controller. During the first phase while going with equal speeds, the ego vehicle matches the desired distance to MIO almost perfectly. When the MIO accelerates (timestamp 168 s), the reaction of the ego vehicle in terms of acceleration is essentially simultaneous and the actual distance drops below the desired one. This is a safe behaviour when accelerating. When the acceleration subsides again (timestamp 183–200 s), the distance error drops. During the braking that follows (timestamp 202 s onwards), the obstacle avoidance function contributes to the deceleration, resulting in the ego

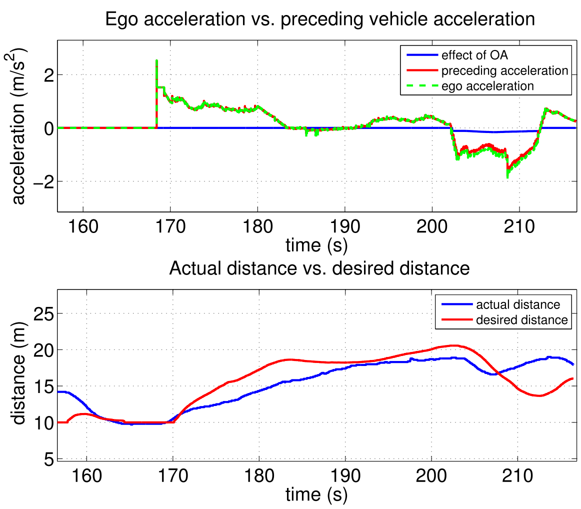


Fig. 13. CACC vehicle accelerations and distance with intended acceleration as input.

acceleration to stay slightly below the MIO acceleration. This allows to increase the gap between the cars and progressively reach the state when the actual distance is above the desired one (from timestamp 208 s on).

B. Communication Robustness

The communication module was the first thing developed during the project and it performed very well during the competition and also already during the IDIADA tests in April 2016. In fact, our immediate belief after receiving the final results from the organisers was that the communication robustness, together with complete and requirements compliant logs, were two major contributing factors to our success.

Despite the satisfactory performance of the system during the competition, the communication module has been further developed after GCDC within an MSc project [45]. The main idea is to provide an efficient solution for Vehicular Ad-hoc Networks (VANET) applicable beyond the GCDC setting. The competition scenario involved only 10 vehicles, while a realistic VANET application may involve many more and thus significantly higher communication load. Keeping up with the 25 Hz communication frequency in such a setting becomes a challenge [30].

To this end a packet prioritising and filtering system based on priority queues called Stream-wise Accumulating Priority Queue (SAPQ) has been developed. In short, the streams are identified by the message type (iCLCM, CAM, or DENM) and origin (vehicle identifier), and each message in the stream has a dynamic accumulating factor b_c based on the message type, physical distance of the message origin, taking into account predicted position due to the processing time overhead and the speed of the sending vehicle. Then, the instantaneous priority q_s of the message in the stream s at time t can be given with the following accumulating priority queue formula [46] $q_s(t) = (t - t_s) \cdot b_c$, where t_s is the arrival time of the first message received in the stream since the last time it was served. Then, in each stream fresher massages override the older messages, which are simply dropped. Out of all streams,

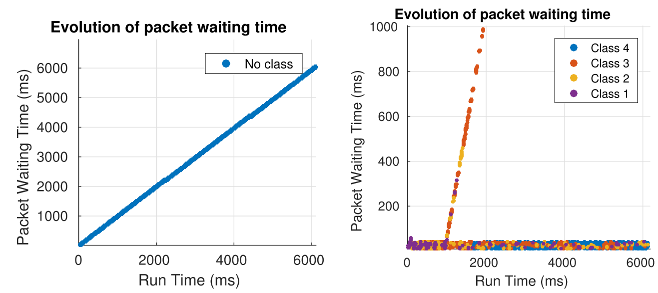


Fig. 14. FIFO (left) vs. SAPQ (right) packet waiting times (high load).

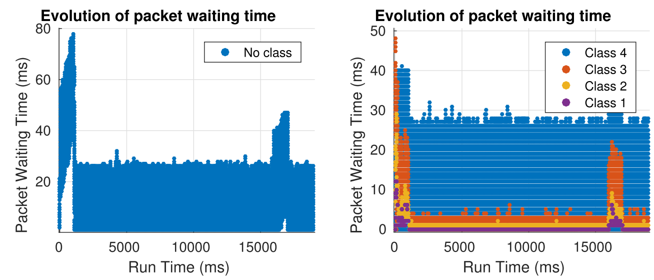


Fig. 15. FIFO (left) vs. SAPQ (right) packet waiting times (low load).

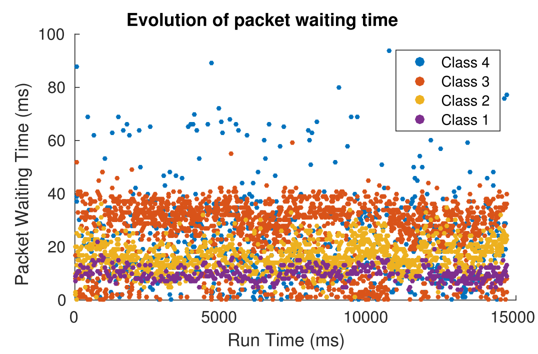


Fig. 16. SAPQ GCDC packet waiting times (high load).

the message with the highest priority is served first. This substantially improves message waiting times for the most important messages and prevents message congestion.

To give an idea of the improvement that SAPQ provides, Fig. 14 shows a comparison of the progression of packet waiting times under high system load (300 vehicles) during a simulation. While the simple FIFO implementation constantly increases the waiting time for all of the packets, the SAPQ implementation suffers only from a temporary increase of waiting times during a simulated DENM message outburst, yet with Class 1 packets not exceeding the waiting time of 380 ms. Under low system load (100 vehicles), the waiting times of high priority packets are considerably lower in the SAPQ implementation compared to an average time for unclassified FIFO, see Fig. 15. Finally, Fig. 16 shows the waiting times of the actual communication messages collected during the competition in an otherwise simulated high load setting. Further technical details and analysis of the SAPQ solution can be found in [45] and a companion paper [21].

C. Trust Index Visualisation

The TS and TI calculation to support decision making described in Sects. VI–VIII have been developed for GCDC,

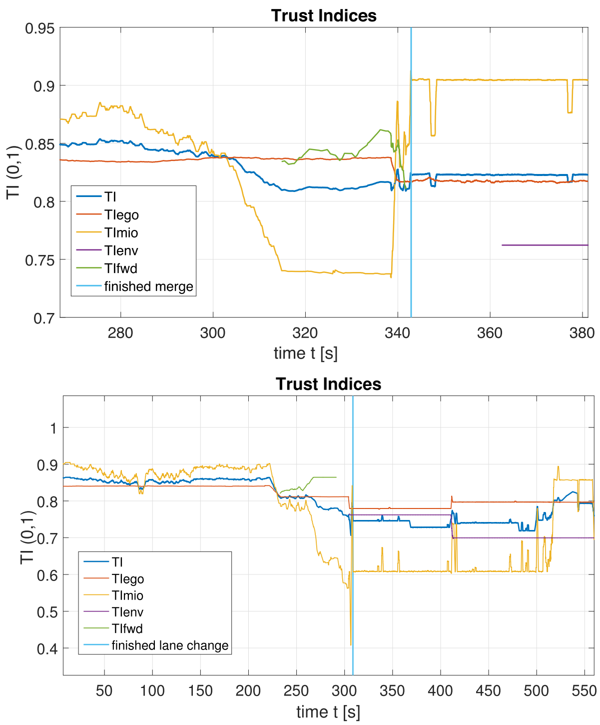


Fig. 17. Trust Index distribution during two merge scenario heats.

but the TIs have not been used to support decision making during the competition for two reasons: (1) due to limited testing possibilities, no prior experimentation data was available to properly weight the decision making based on TIs, the competition data from the fully running systems was the first data available; (2) according to the competition requirements, most decisions had to be driver-confirmed, effectively taking full automation, for which TS would be crucial, out of the scope. In other words, regardless of the calculations a very low trust in the surrounding environment was assumed due to experimental context. Nevertheless, the TS building blocks, in particular the vehicle distance and vehicle position model for sensor data fusion, have been implemented and were running live in the system which produced the necessary data for further evaluation of the TS results.

Figure 17 shows the TI distribution from two highway heats. In the first heat the vehicle was merging from the left platoon, in the second one the vehicle was on the right making a gap. In the first case TI_{MIO} is degrading due to losing the MIO out of sight. After the merge (342 s) the TI_{MIO} re-establishes itself at $\approx$0.9 level when the system identifies a new MIO. In the second case TI_{MIO} is initially similar to the first case, while it drops after the merge is completed. In this particular case the new MIO did not provide correct position through V2V, only occasionally which is represented by the spikes in the graph. In effect, the global TI is smaller by $\approx$0.1 compared to the first case indicating overall lower situation awareness. TS information could be, e.g., used to decrease the headway time to the MIO following the high trust in the reliability of its data. Further technical details about the TS can be found in [39] and a companion paper [22].

XI. CONCLUSIONS

We have presented the general design and functioning of the team Halmstad system for cooperative driving that was field-tested during GCDC. Apart from successfully competing and winning GCDC 2016, the competition was used as a site for collecting data, which assisted further development, and evaluation of our system during and after the competition. The major contributions and outcomes from the project as a whole are (1) result from the field functioning of the CACC controller and an experiment using intended acceleration for forward feedback in our controller; (2) a priority queue based message dispatching system in the communication module, which substantially improves the communication throughput (further described in [21]); and (3) the trust system, that was evaluated with realistic data (further described in [22]).

The goal of the i-GAME project that organised the GCDC competition is to address and advance research in intelligent transportation systems keeping the societal challenges in mind. For the Halmstad GCDC student team the main challenge and focus were in the robustness of the system and developing the system in a timely fashion. For these reasons optional functionality was not implemented, in particular the system had no support for lateral control of the vehicle. However, to follow the modern publicity and dissemination trends, two short films advertising the team's efforts were made and published on YouTube,[13] one at the preparatory stage of the project, and one after the competition.

An ideal follow-up of the team's effort would be to complete the prototype system with the optional functionalities and continue with the evaluation and development of the single modules to reach production grade quality and applicability in a real traffic environment.

ACKNOWLEDGMENT

We would like to thank the i-GAME project and associated institutions for preparing and organising GCDC 2016, the Swedish CoAct project and institutions for supporting us organisationally and financially, and all of our sponsors including Fengco, TASS International, and Volvo Cars. The work of Wojciech Mostowski is supported by the Swedish Knowledge Foundation grant for the AUTO-CAAS project. We would also like to thank the anonymous reviewers for their valuable comments that helped to improve this paper.

REFERENCES

[1] "Statistical office of the european communities, energy, transport and environment indicators," in *Eurostat Pocketbooks on Environment and Energy*. Luxembourg: Publications Office of the European Union, 2014.

[2] C. Wallmark, M. Goldmann, P. B. Odhner, S. Forward, and K. Hoyer, "Fossiloberoende fordonsflotta 2030—Hur realiserar vi målet?" Sweco, Stockholm, Sweden, Tech. Rep. Slutrapport 2016-09-01, 2016.

[3] G. C. Patton *et al.*, "Global patterns of mortality in young people: A systematic analysis of population health data," *Lancet*, vol. 374, no. 9693, pp. 881–892, Sep. 2016.

[4] W. J. Gillan, "PROMETHEUS and DRIVE: Their implications for traffic managers," in *Proc. Conf. Rec. Veh. Navigat. Inf. Syst. Conf.*, Sep. 1989, pp. 237–243.

[13]https://youtu.be/rKvb6Dmy7sY, https://youtu.be/2IoVsAl2yTA.

[5] R. Rajamani, H.-S. Tan, B. K. Law, and W.-B. Zhang, "Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons," *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 4, pp. 695–708, Jul. 2000.

[6] C. Bergenhem, Q. Huang, and A. Benmimoun, "Challenges of platooning on public motorways," in *Proc. 17th World Congr. Intell. Transp. Syst.*, 2010, pp. 1–12.

[7] A. De La Fortelle *et al.*, "Network of automated vehicles: The AutoNet 2030 vision," in *Proc. ITS World Congr.*, 2014. [Online]. Available: https://hal-mines-paristech.archives-ouvertes.fr/hal-01063484/document

[8] M. Aramrattana, T. Larsson, J. Jansson, and C. Englund, "Dimensions of cooperative driving, its and automation," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun./Jul. 2015, pp. 144–149.

[9] S. E. Shladover, C. Nowakowski, X.-Y. Lu, and R. Ferlis, "Cooperative adaptive cruise control (CACC) definitions and operating concepts," in *Proc. 94th Annu. Meet. Transp. Res. Board (TRB)*, 2015, pp. 1–16.

[10] H. H. Bengtsson, L. Chen, A. Voronov, and C. Englund, "Interaction protocol for highway platoon merge," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 1971–1976.

[11] M. Amoozadeh, H. Deng, C.-N. Chuah, H. M. Zhang, and D. Ghosal, "Platoon management with cooperative adaptive cruise control enabled by VANET," *Veh. Commun.*, vol. 2, no. 2, pp. 110–123, 2015.

[12] E. S. Kazerooni and J. Ploeg, "Interaction protocols for cooperative merging and lane reduction scenarios," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 1964–1970.

[13] S. Eilers *et al.*, "Companion—Towards co-operative platoon management of heavy-duty vehicles," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 1267–1273.

[14] A. Böhm, M. Jonsson, and E. Uhlemann, "Performance comparison of a platooning application using the IEEE 802.11p MAC on the control channel and a centralized MAC on a service channel," in *Proc. IEEE 9th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2013, pp. 545–552.

[15] M. Segata *et al.*, "Towards inter-vehicle communication strategies for platooning support," in *Proc. 7th Int. Workshop Commun. Technol. Veh. (Nets4Cars-Fall)*, Oct. 2014, pp. 1–6.

[16] *Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems Operating in the 5 GHz Range; Access Layer Part*, document ETSI, TS 102 687 V1.1.1, Jul. 2011.

[17] C. Sommer, S. Joerer, M. Segata, O. K. Tonguz, R. L. Cigno, and F. Dressler, "How shadowing hurts vehicular communications and how dynamic beaconing can help," *IEEE Trans. Mobile Comput.*, vol. 14, no. 7, pp. 1411–1421, Jul. 2015.

[18] E. van Nunen, R. J. A. E. Kwakkernaat, J. Ploeg, and B. D. Netten, "Cooperative competition for future mobility," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1018–1025, Sep. 2012.

[19] K. Lidström *et al.*, "A modular CACC system integration and design," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1050–1061, Sep. 2012.

[20] C. Englund *et al.*, "The grand cooperative driving challenge 2016: Boosting the introduction of cooperative automated vehicles," *IEEE Wireless Commun.*, vol. 23, no. 4, pp. 146–152, Aug. 2016.

[21] V. Rodríguez, J. Detournay, A. Vinel, and N. Lyamin, "An approach for receiver-side awareness control in vehicular ad-hoc networks," to be published, doi: 10.1109/TITS.2017.2749966.

[22] T. Rosenstatter and C. Englund, "Modelling the level of trust in a cooperative automated vehicle control system," to be published, doi: 10.1109/TITS.2017.2749962.

[23] C. Englund, K. Lidström, and J. Nilsson, "On the need for standardized representations of cooperative vehicle behavior," in *Proc. 2nd Int. Symp. Future Active Safety Technol. Toward Zerotraffic-Accident*, Sep. 2013, pp. 1–6.

[24] A. S. Huang, E. Olson, and D. C. Moore, "LCM: Lightweight communications and marshalling," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2010, pp. 4057–4062.

[25] A. Ebadighajari, "Vehicle control in cooperative driving," M.S. thesis, Dept. Signals Syst., Chalmers Univ. Technol., Gothenburg, Sweden, 2011.

[26] A. M. Medina and E. Semsar-Kazerooni, "DEL150318 i-GAME D2.2 Generic real-time control system FINAL," i-GAME Project, Tech. Rep. D2.2, Mar. 2015.

[27] C. Englund *et al.*, "DEL151031 i-GAME D1.4 final report on requirements specification," i-GAME Project, Tech. Rep. D1.4, 2015.

[28] A. Voronov, J. De Jongh, D. Heuven, and A. Severinson. (Jun. 2016). *Implementation of ETSI ITS G5 GeoNetworking Stack, in Java: CAM-DENM/ASN.1 PER/BTP/GeoNetworking*. [Online]. Available: https://doi.org/10.5281/zenodo.55650

[29] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*, document ETSI, TS 302 687-2 V1.3.1, Sep. 2014.

[30] O. Baijer, P. Balaguer, H. H. Bengtsson, L. Chen, L. Garcia-Sol, and J. van de Sluis, "D3.2 proposal for extended message set for supervised automated driving," i-GAME Project, Tech. Rep. D3.2, Sep. 2015.

[31] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specification of Decentralized Environmental Notification Basic Service*, document ETSI, TS 302 637-3 V1.2.2, Nov. 2014.

[32] J. van de Sluis, "D3.1 wireless communication basic specification document," i-GAME Project, Tech. Rep. D3.1, Sep. 2015.

[33] E. Semsar-Kazerooni, A. M. Medina, and H. H. Bengtsson, "DEL150330 i-GAME D2.1 interaction protocol," i-GAME Project, Tech. Rep. D2.1, 2015.

[34] *Data Elements and Interchange Formats—Information Interchange—Representation of Dates and Times*, document ISO 8601:2004, 2004.

[35] G. Combs *et al. Wireshark*. Accessed: Oct. 2, 2017. [Online]. Available: http://www.wireshark.org

[36] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *J. Basic Eng.*, vol. 83, no. 1, pp. 95–108, 1961.

[37] N. Magnusson and T. Odenman, "Improving absolute position estimates of an automotive vehicle using GPS in sensor fusion," M.S. thesis, Dept. Signals Syst., Chalmers Univ. Technol., Gothenburg, Sweden, 2012.

[38] L. Danielsson, "Tracking and radar sensor modelling for automotive safety systems," Ph.D. dissertation, Dept. Signals Syst., Chalmers Univ. Technol., Gothenburg, Sweden, 2010.

[39] T. Rosenstatter, "Modelling the level of trust in a cooperative automated vehicle control system," M.S. thesis, School Inf. Technol., Halmstad Univ., Halmstad, Sweden, Sep. 2016.

[40] R. Schubert, N. Mattern, and R. Bours, "Evaluating automated vehicle systems using probabilistic sensor simulations," in *Proc. 10th ITS Eur. Congr.*, Helsinki, Finland, Jun. 2014, pp. 1–5.

[41] D. Swaroop and J. K. Hedrick, "String stability of interconnected systems," *IEEE Trans. Autom. Control*, vol. 41, no. 3, pp. 349–357, Mar. 1996.

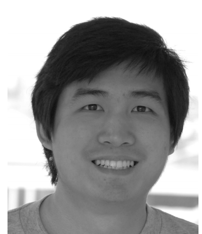
[42] C. Wang and H. Nijmeijer, "String stable heterogeneous vehicle platoon using cooperative adaptive cruise control," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 1977–1982.

[43] J. Zhou and H. Peng, "Range policy of adaptive cruise control vehicles for improved flow stability and string stability," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 2, pp. 229–237, Jun. 2005.

[44] O. U. Jansson and G. Shahanoor, "Evaluation of string stability during highway platoon merge," M.S. thesis, School Inf. Technol., Halmstad Univ., Halmstad, Sweden, Sep. 2016.

[45] V. D. Rodríguez and J. Detournay, "An approach for receiver-side awareness control in vehicular ad-hoc networks," M.S. thesis, School Inf. Technol., Halmstad Univ., Halmstad, Sweden, Sep. 2016.

[46] L. Kleinrock, "A delay dependent queue discipline," *Naval Res. Logistics Quart.*, vol. 11, nos. 3–4, pp. 329–341, 1964.



Maytheewat Aramrattana received the B.Eng. degree in electrical engineering from Kasetsart University, Thailand, in 2010, and the M.Sc. degree in computer science and engineering (embedded and intelligent systems programme) from Halmstad University, Sweden, where he is currently pursuing the Ph.D. degree. He was with the Swedish National Road and Transport Research Institute as part of his Ph.D. project, VICTIg. His research interests are in simulation, modeling, testing, and evaluation of cooperative intelligent transport systems.

Jérôme Detournay received the M.Sc. degree in embedded and intelligent systems from Halmstad University, Sweden, in 2016. He was part of the Team Halmstad for the Grand Cooperative Driving Challenge in 2016. His interests are in communication and networking.

Cristofer Englund received the Ph.D. degree in electrical engineering from the Chalmers University of Technology, Sweden, in 2007. He currently holds a research managerial position within cooperative systems at RISE Viktoria, Gothenburg, Sweden. He is also the Research Director of the Traffic Systems competence area with SAFER, Vehicle and Traffic Safety Centre, Chalmers University of Technology. In addition, he holds an adjunct senior lecturer position with Halmstad University within information technology. His research interests include automated driving, machine learning, and data mining.

Viktor Frimodig is currently pursuing the M.Sc. degree in computer science and engineering with Halmstad University, Sweden. He is with Volvo Group Trucks Technology. He was part of the Team Halmstad for GCDC 2016, where he was mainly focused on the design of control algorithms. His interests include embedded and real-time systems and control theory.

Oscar Uddman Jansson received the B.Sc. degree in mechatronics and the M.Sc. degree in embedded and intelligent systems from Halmstad University, Sweden, in 2014 and 2016, respectively. He was a Consultant with Sigma Technology, Sweden. He was a part of the Halmstad University's GCDC 2016 Team.

Tony Larsson received the master's degree in mechanical engineering and the Ph.D. degree in computer science from Linköping University, Sweden, in 1974 and 1989, respectively. He was with Ericsson AB for 28 years in the areas of design and testing of distributed real-time systems. During this time, he was appointed as a Technical Expert in system design methods. After this, he was shortly with FMV with system architecture for network-based defence and in 2003 he became a Professor of embedded systems with Halmstad University, Sweden. He is currently a Professor with Halmstad University. His current interests include software architecture supporting cooperative embedded systems aiding intelligent transport, vehicle and mobile sensor network applications relying on wireless communication.

Wojciech Mostowski received the Ph.D. degree from the Chalmers University of Technology, Sweden, in 2005. He held post-doctoral position with two universities in The Netherlands. Most recently, he is involved in the AUTO-CAAS project concerned with the model-based testing of AUTOSAR-based automotive software. He is currently an Assistant Professor with Halmstad University, Sweden. His research interests are in formal methods and software correctness, smart card technology and applications, and in embedded systems for automotive applications.

Víctor Díez Rodríguez received the B.Sc. degree in computer science from the University of Burgos, Spain, in 2014, and the M.Sc. degree in embedded and intelligent systems from Halmstad University, Sweden, in 2016. He was a Consultant Engineer with Segula Technologies, Sweden. He was the part of Team Halmstad during the Grand Cooperative Driving Challenge 2016. His main interests are vehicular communication and software engineering.

Thomas Rosenstatter received the master's degree in embedded and intelligent systems from Halmstad University, Sweden, in 2016, and the Dipl.-Ing. degree from the Salzburg University of Applied Sciences, Austria, in 2016. He is currently pursuing the Ph.D. degree in vehicular security with the Chalmers University of Technology, Sweden. His interests are in security and software architecture. He participated in GCDC 2016 as part of Team Halmstad.

Golam Shahanoor received the B.Sc. degree in electrical and electronic engineering from Eastern University, Bangladesh, and the M.Sc. degree in embedded and intelligent system from Halmstad University, Sweden. He is a Research Engineer in autonomous vehicles with Semcon AB, Sweden. His area of interests is in robotics and autonomous system. He was a member of the Team Halmstad during Grand Cooperative Driving Challenge 2016.