# System Architecture of a Driverless Electric Car in the Grand Cooperative Driving Challenge

Philippe Xu, Gérald Dherbomez, Elwan Héry, Abderrahmen Abidli and Philippe Bonnifait

*Abstract*—This paper presents the complete system architecture of a connected driverless electric car designed to participate in the Grand Cooperative Driving Challenge 2016. One of the main goals of this challenge was to demonstrate the feasibility of multiple autonomous vehicles cooperating via wireless communications on public roads. Several complex cooperative scenarios were considered, including the merging of two lanes and cooperation at an intersection. We describe in some detail an implementation using the open-source PACPUS framework that successfully completed the different tasks in the challenge. Our description covers localization, mapping, perception, control, communication and the human-machine interface. Some experimental results recorded in real-time during the challenge are reported.

*Index Terms*—Driverless car, autonomous vehicle, cooperative driving, Grand Cooperative Driving Challenge, electric car.

## I. INTRODUCTION

**A**UTONOMOUS vehicles have been the subject of a number of research works in fields including robotics, computer vision and machine learning [1]. In the last decades, tremendous progress has been made towards driverless cars [2]. The well-known DARPA Grand Challenge [3], [4] has triggered a great deal of research by confronting academics with real practical situations. Following in the path of this challenge, others have since been set up, such as the Intelligent Vehicle Future Challenge (IVFC) in China and the Grand Cooperative Driving Challenge (GCDC) in Europe [5].

Autonomous driving by vehicles with perception capabilities has been demonstrated in many contexts. In particular, an automatic lane keeping function has already been implemented in some models of cars sold to the general public. But there are still challenges to be overcome, and one of these challenges is executing complex driving maneuvers in interaction with surrounding vehicles. Wireless communication between vehicles and the infrastructure is one way of pushing autonomous driving a step further. More complex situations can be solved where vehicles are able to cooperate with one another. One of the aims of GCDC is to show the feasibility of cooperating systems.

The Heudiasyc Laboratory, a French National Center for Scientific Research (CNRS) and University of Technology of Compiègne (UTC) joint research unit, took part in GCDC in 2016. An automated electric car was used by the team (see Fig. 1). It was the only electric car that participated in the final stage of the challenge.

The authors are with Sorbonne Universités, Université de Techonologie de Compiègne, CNRS, Heudiasyc, Compiègne, France. E-mail: firstname.surname@hds.utc.fr.
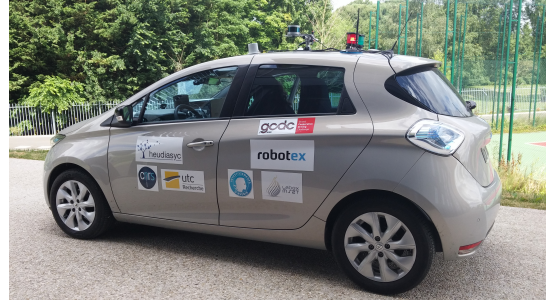


Fig. 1. Heudiasyc team APAChE electric car.

The purpose of this article is to present the complete system architecture of the driverless electric car that successfully completed the different scenarios in GCDC.

In Section II, GCDC is presented, including the different scenarios in the 2016 edition. Section III gives a detailed view of the system architecture, covering all the implemented modules: localization, perception, control, communication, HMI and supervision. The open-source software platform, PACPUS, developed by the Heudiasyc Laboratory for mobile robotics, is then presented in Section IV. Section V presents the experimental vehicle used by the Heudiasyc team. Some experimental results recorded during GCDC are reported and discussed in Section VI. Finally, Section VII concludes this paper and provides an overview of future work.

## II. THE GRAND COOPERATIVE DRIVING CHALLENGE

The first GCDC [5] was held in 2011 at Helmond in the Netherlands. The focus of the first GCDC was cooperative adaptive cruise control (CACC). The second edition was held in May 2016 and introduced more complex autonomous maneuvers in addition to cooperative platooning. Three cooperative scenarios were considered during this challenge.

*a) Merging:* The first and most challenging scenario was merging. The scenario took place on a two-lane highway. It started with normal platooning on the two parallel lanes at a speed of 80 km/h for the left lane and 60 km/h for the right lane. Some road works in the left lane were then signaled to all the vehicles by a roadside unit (RSU) using a specific message sent via wireless communication. After the vehicles in the two lanes had slowed down to 40 km/h, all the vehicles in the left lane had to merge into the right lane according to a specific iterative protocol.

*b) Crossing:* Three vehicles approached a T-intersection at a speed of 30 km/h and had to negotiate the insertion of one of the vehicles. The vehicles taking part in the challenge were
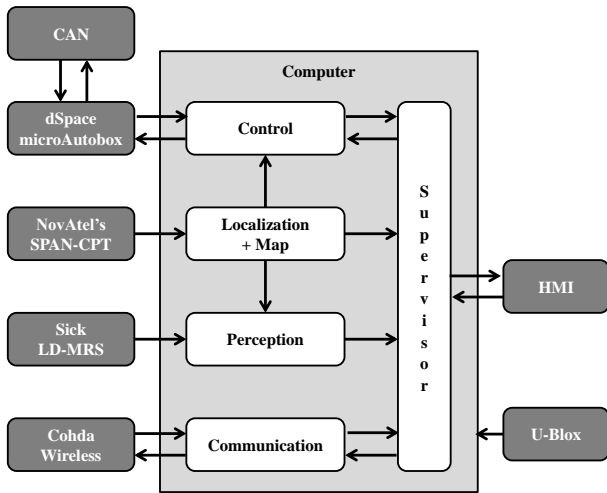
Fig. 2. Global architecture of the platform. The dark gray boxes correspond to physical devices. The white boxes are different software modules implemented within the PACPUS framework on a single computer.



Fig. 3. A270 highway in Helmond, Netherlands. The recorded trajectories, in blue, are overlaid on the map retrieved from OpenStreetMap.

on the main road, while a vehicle entering the intersection belonged to the organizers. The organizers' vehicle was deemed to have priority, meaning that the vehicles on the main road had to slow down and allow it to enter the intersection safely.

*c) Emergency:* The final scenario was for public demonstration only and was not included in the judging. While platooning on a two-lane highway at 80 km/h, the vehicles were required to create a lateral space between the two lanes in order to allow an emergency vehicle to pass between them.

Ten international teams took part, bringing a wide variety of automotive solutions, including two trucks. The challenge also included two vehicles from the organization committee. The merging and emergency scenarios involved all the participating vehicles. The cooperation between heterogeneous systems implementing different algorithms was one of the main challenges.

## III. SYSTEM ARCHITECTURE

In order to perform all the tasks required in GCDC, we needed to bring several sensors and processing modules into play. Fig. 2 shows the global architecture of the system implemented on the experimental platform. A single computer running a Linux OS was used to handle the different sub-modules. Four basic modules for control, localization, perception and communication were implemented using the PACPUS middleware. A supervision module was implemented to handle the protocol for interacting with other vehicles and for communicating with the driver through a human-machine interface (HMI).

### A. Localization and map

Localization is a critical feature in autonomous driving. By having a spatial positioning with respect to its environment, the vehicle can define trajectories to be followed and perform autonomous driving maneuvers. The environment itself is often represented as a navigation map. However, the maps

provided by services such as OpenStreetMap [6] are usually not accurate enough to be used in an automatic control context. High-definition maps [7], [8] with lane-level accuracy are a topic of particular interest to both map providers and car manufacturers. They are destined to cover more and more geographic areas in the near future. The map used in GCDC was built online prior to the challenge, and featured only the center paths of each of the two lanes of the A270 highway used during the challenge in Helmond. Fig. 3 shows the recorded trajectories overlaid on a map from OpenStreetMap.

A NovAtel SPAN-CPT was used to give localization data with centimeter accuracy. This combines a Global Navigation Satellite System (GNSS) and an Inertial Navigation System (INS). With Real-Time Kinematic (RTK) corrections, the solution combining GNSS and INS solution can provide both absolute accuracy and continuity for localization. By using the highest frequency of the inertial measurement unit (IMU), the SPAN can provide GNSS/INS localization data at a frequency of 50 Hz. Among other data, the solution provides information on localization (latitude, longitude, altitude), velocity (w.r.t. east, north, up directions), acceleration (lateral, longitudinal, vertical), rotation (roll, pitch, azimuth) and rotation rate (roll rate, pitch rate, yaw rate). It also provides the standard deviations for all these estimates.

Geodetic coordinates [9], i.e., latitude and longitude, are often unsuitable for robotic tasks such as navigation, and Cartesian coordinates are much more practical. We chose to use local East, North, Up (ENU) coordinates [10]. Compared to other Cartesian coordinate system such the Earth-Centered, Earth-Fixed (ECEF) coordinate system, using ENU coordinates is much more intuitive, since they provide simple map-like 2D planar projections. Unlike map-projection-based systems such as the Universal Transverse Mercator (UTM) coordinate system, the transformation from three-dimensional geodetic coordinates to the ENU coordinate system is invertible. Moreover, the local origin of the ENU coordinate system can be changed dynamically so that the 2D navigation plane is always accurate.

### B. Perception

Perception and scene understanding are among the most important and challenging tasks for autonomous driving. In particular, object detection and tracking have been widely studied in the computer vision, machine learning and robotics

communities [1]. In robotics, one of the most accurate sensors for range and bearing detection is LiDAR (Light Detection And Ranging). Model-based approaches [11] and supervised object detectors [12] are the most widely used methods in LiDAR-based detection.

In GCDC the perception task is highly simplified, as the environment is restricted to a straight two-lane highway with the other participants' vehicles the only possible obstacles. Moreover, for highway platooning, only the distance to the vehicle with which the host vehicle is platooning needs to be computed. In the GCDC scenarios, platooning was done with respect to two possible vehicles: the vehicle immediately in front of the host vehicle, referred to as the MIO (Most Important Object), and the vehicle ahead of the host vehicle in the other lane (either the left or the right lane, depending on the host vehicle's current lane), referred to as the forward MIO. Nominal platooning was done with respect to the MIO, but during a merging maneuver it was done with respect to the forward MIO.

A four-layer Sick LD-MRS LiDAR was used in our experiments. The sensor was set so that the third layer was approximately parallel to the ground. The first two lower layers were used to detect the ground plane, while the two upper layers were used for object detection. Because of the high quality of the sensor, with a low false alarm rate and the limited complexity of the task, we chose to use a simple geometrical approach, rather than build a more complex general object detection method.

Given a point cloud measured by the LiDAR, a geometrical clustering was used to detect the different objects in the scene. Two consecutive LiDAR points can be considered to belong to the same object if their relative distance is below a given threshold. Once the points are clustered, rectangular bounding boxes are computed around the clusters. These clusters are then filtered based on their size, shape and distances with respect to the different lanes and to the host vehicle.

Let $d^*$ be the distance to a given vehicle of width $w$, and let $\phi$ be the angular resolution of the LiDAR. The maximum number of laser beams hitting the back of the vehicle is given as

$$n = w/\Delta d \quad \text{with} \quad \Delta d = d^* \cdot \tan \phi, \quad (1)$$

where $\Delta d$ represents the minimum distance between two consecutive points. Inversely, if the minimum number of points in a cluster is set, then the greatest distance at which a vehicle can be detected can be computed by inverting Eq. (1).

Fig. 4 illustrates a typical perception configuration. The goal of the perception module is to compute the two distances $d_1$ and $d_2$ which represent the longitudinal distances from the host vehicle to the MIO and the forward MIO, respectively. The lane margin $p$ controls the width of the spatial zones, i.e., navigation corridors, in which the detected objects can be considered as the MIO or the forward MIO. The two computed distances serve as input to the control module.

## C. Control

On the experimental vehicle, the control was limited to sending three possible commands to the vehicle CAN bus:
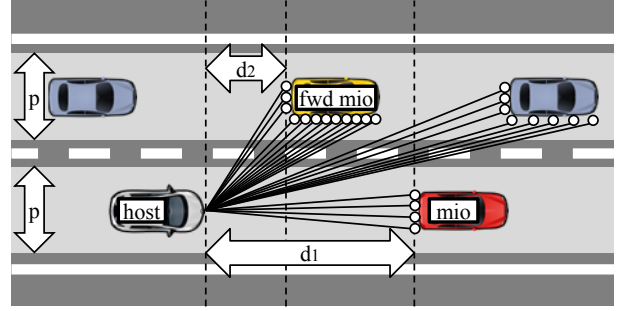


Fig. 4. Typical LiDAR-based perception configuration. The lines and white dots represent the LiDAR beams and impacts on obstacles. The parameter $p$ represents the width of the navigation corridors in which the MIO (bottom gray corridor) and the forward MIO (top gray corridor) should be contained.

acceleration torque, braking torque and steering wheel angle. Since the challenge was carried out on a motorway, the longitudinal and lateral controls of the vehicles were implemented in a decoupled way [13]. In the scope of the platooning scenario of the GCDC, the nominal speed was set to 40 km/h with maximum allowed acceleration and deceleration set to 2 m/s$^2$. In terms of lateral dynamics, the maximum steering wheel angle for lane keeping and lane merging was no more than 10 degrees which corresponds to an angle less than 1 degree for the vehicle front wheels. Within this context, i.e., low speeds and small angles, two main assumptions were made: the slip of the wheels was neglected and we assumed a linear relationship between the angle of the steering wheel and the one of front wheels. A bicycle-based model was therefore sufficient to encompass the kinematics of the used vehicle.

*1) Longitudinal control:* When the car was not platooning, for example when approaching the intersection, the longitudinal control was handled through a speed regulator controlling the acceleration and braking torques. The use of a two-wheeled vehicle model [14] simplifies the model of the longitudinal dynamics. We took into account the vehicle equivalent mass $m_e$ and the aerodynamic drag force $F_a$ defined as

$$m_e = m + 2I_w/R_e^2 \text{ and } F_a(t) = \frac{1}{2}\rho \cdot c_d \cdot s \cdot v(t)^2, \quad (2)$$

with $I_w$ the rotational inertia of the wheels, $R_e$ the effective radius of the wheels, $\rho$ the mass density of the air, $c_d$ the aerodynamic drag coefficient, $s$ the frontal area of the vehicle and $v$ the longitudinal velocity.

The longitudinal model of the car being an integrator with an input disturbance, a proportional-integral (PI) controller with a direct compensation term computes the torque input $T(t)$ as

$$T(t) = m_e R_e \left( K_p e_v(t) + K_i \int_{t_0}^t e_v(\tau)d\tau \right) + R_e F_a(t), \quad (3)$$

with $e_v(t)$ the velocity error at time $t$ w.r.t. a desired velocity $v_d$ and $K_p$, $K_i$ the proportional and integral terms. The closed-loop transfer function is basically a second order system for which it is straightforward to compute gains for given response time and damping ratio.
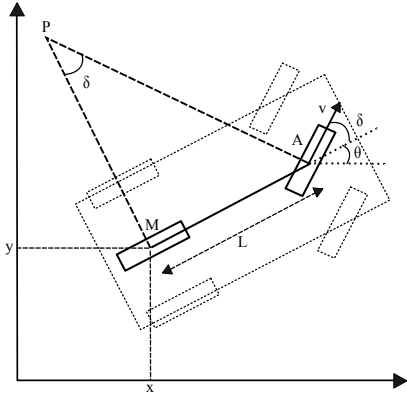
Fig. 5. Vehicle kinematic model. The $M$ and $A$ points represent the center of the virtual wheels located in the middle of the rear and front wheel axles, respectively. The point $P$ is the instant center of rotation and $L$ is the wheelbase length. The angle $\theta$ is the heading of the vehicle and $\delta$ is the angle of the virtual front wheel.

Depending on the scenario, a desired speed profile was set that the longitudinal controller had to respect. In platooning, cooperative driving was defined in the challenge by a distance $d$ to be respected between vehicles:

$$d = r + h \cdot v, \qquad (4)$$

where $r$ is a standstill distance (typically 6 meters) and $h$ is a time headway (typically 1.5 seconds).

Given a range measurement provided by the perception module, the desired speed was simply derived by inverting Eq. (4), i.e., $v_d = (d-r)/h$. Since our perception system based on LiDAR did not provide a relative speed, we had initially planned to use the speed communicated by the vehicle ahead to fix the damping ratio, but we noticed during the trials that the information coming from the other cars was not sufficiently reliable for control operations. If the coefficients $r$ and $h$ are set smaller, communication between vehicles can be used in a better way as in [15].

*2) Lateral control:* A number of lateral control laws for vehicles can be found in the literature [16], in particular the Stanley control law [17] is well known for following paths. Most of these laws base the kinematic model of the vehicle on the classical Ackermann model. This model neglects slippage which is a reasonable assumption in our case. In this model, the car is modeled as a bicycle, as illustrated in Fig. 5. In our case, we implemented the steering controller as a state feedback with pole placement. For a front-wheel drive car with a wheelbase $L$ running at speed $v$, which is considered constant as we are considering an decoupled control law, a kinematic bicycle model is given by:

$$\dot{x} = v \cos \delta \cos \theta \; ; \; \dot{y} = v \cos \delta \sin \theta \text{ and } \dot{\theta} = v \sin \delta / L, \quad (5)$$

where $\delta$ corresponds to the angle of a virtual front wheel located at the middle of the car and is proportional to the steering wheel angle for small angles. This kinematic model can be refined by taking into account a drift angle for high speeds [18] which was not necessary for GCDC 2016.

In a Frenet frame such that the lane to follow is horizontal, we can write:

$$\dot{y} = v \cos \delta \sin \theta \quad \text{and} \quad \dot{\theta} = v \sin \delta / L. \qquad (6)$$

Here, $y$ corresponds to the cross-track distance at the center of the rear axle and $\theta$ is the heading error. For simplicity, we assume that $\delta$ and $\theta$ remain small (a reasonable assumption on a motorway): $\cos \delta \approx 1$, $\sin \delta \approx \delta$ and $\sin \theta \approx \theta$. The model becomes:

$$\dot{y} = v\theta \quad \text{and} \quad \dot{\theta} = v\delta / L. \qquad (7)$$

Taking the derivative of the first equation for a constant speed:

$$\ddot{y} = v \cdot \dot{\theta} = v \cdot v\delta / L = v^2 \delta / L. \qquad (8)$$

The system is a double integrator for which a proportional-derivative control law is known to work efficiently. For a desired path to follow $(y_d, \dot{y}_d, \ddot{y}_d)$, the control law is therefore given by:

$$\delta = \frac{L}{v^2} \left( \alpha_1 \left( y_d - y \right) + \alpha_2 \left( \dot{y}_d - \dot{y} \right) + \ddot{y}_d \right) \text{ and } v \neq 0, \quad (9)$$

where $\alpha_1$ and $\alpha_2$ are the gains. Since $\dot{y} = v\theta$ and since the path is a straight line, the control law becomes:

$$\delta = \frac{L}{v^2} \left( \alpha_1 \cdot e_l + \alpha_2 v \cdot e_\theta \right) \text{ and } v \neq 0, \qquad (10)$$

where $e_l$ and $e_\theta$ are respectively the lateral and heading errors of the vehicle with respect to the reference path.

This control law is time-triggered and it is easy to tune the gains $\alpha_1$ and $\alpha_2$ to get a critical mode (without any overshoot) corresponding to a chosen time constant. Moreover, thanks to this law, a smooth lane change between two adjacent lanes can be implemented by switching from the current lane to the new lane without path planning. Finally, the singularity of Eq. (10) when the car was motionless was addressed in the implementation of the control law.

*D. Communication*

The wireless communication used in GCDC was based on ETSI C-ITS standards (Cooperative-Intelligent Transport System). It covered both Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications. The implementation followed the ETSI ITS-G5 standard, including the GeoNetworking protocol and BTP (Basic Transport Protocol) [19]. The standard CAM (Cooperative Awareness Message) [20] and DENM (Distributed Environment Notification Message) [21] were used during the challenge. In addition, a non-standard set of messages, *iCLCM* (*i-GAME Cooperative Lane Change Message*) was added to provide an extension to CAM and DENM.

CAM messages mainly convey information about the vehicle itself, such as its localization, heading, speed, acceleration, size, type and role. CAM messages were sent by all the vehicles. DENM messages essentially provide information about road safety related events. In GCDC, DENM messages were used to notify the presence of road works and were only sent by the roadside unit. Using only CAM and DENM was not enough to enable complex interaction between vehicles.

Fig. 6. Human Machine Interface implemented on an Android based tablet.
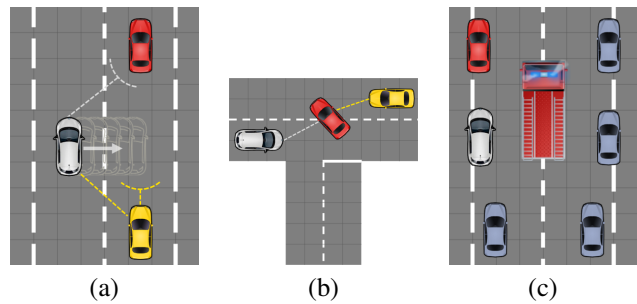


(a)        (b)        (c)

Fig. 7. A sample of images shown on the HMI tablet for the three different scenarios: Merging (a), Crossing (b) and Emergency (c). The host vehicle is the light gray car. (a) The host vehicle is doing the merging maneuver. (b) The red vehicle, which has the highest priority, crosses the intersection. (c) The emergency vehicle passes through the two lanes.

One of the main purposes of iCLCM is to provide an extended message set to vehicles in order to interact within a pre-defined protocol during a merging maneuver. They were also used by the RSU to start and stop the scenarios.

A Cohda Wireless communication device was used to provide communication capabilities. This modem served as an UDP IPv4 to 802.11p gateway for sending and receiving messages. Messages were encoded and decoded on a computer communicating with the Cohda via UDP. The basic ETSI ITS-G5 stack was implemented in C++ within the PACPUS framework. The encoding and decoding of this stack was implemented with the help of the *asn1c* compiler (http://lionet.info/asn1c/compiler.html).

### E. Human-Machine Interface

We implemented a Human Machine Interface (HMI) on an Android-based tablet for displaying useful information and interacting with the safety driver (see Fig. 6). The main concern in the design of the HMI was to display sufficient information while preventing the safety driver from being too distracted. For this purpose, a single image was used to illustrate each step of a given scenario. Additionally, each image was associated with a vocal message describing it, so that the safety driver could know the current step without having to look directly at the HMI screen. The tablet audio output was connected to the audio system of the car through a Bluetooth connection. A voice synthesizer was used to create vocal messages, meaning that new messages could be generated on the fly with no pre-recording. Fig. 7 shows a sample of images shown for the three different scenarios.

### F. Supervision

As shown in Fig. 2, the supervision module was connected to all the other modules. The main role of the supervisor was to follow a specific cooperative interaction protocol with the other vehicles. Several state diagrams were defined for each scenario.

Fig. 8 illustrates the state diagrams of the merging scenario. After the vehicle has finished its initialization, it waits for an iCLCM "StartPlatoon" message from the RSU. Once the message is received, the vehicle starts platooning in its current lane. When a DENM "roadworks" message is received, the

vehicle has to pair up with its forward MIO by exchanging their station IDs through iCLCM messages. If the host vehicle is in the left lane, it needs to wait for all the vehicles ahead of it to merge before being becoming the leader in its lane. As a leader, the vehicle then needs to wait for its backward pair to send an iCLCM "SafeToMerge" message and to wait for the driver to manually confirm the start of the merging maneuver. After merging, the vehicle returns to a normal platooning mode. If, on the other hand, the host vehicle is in the right lane, after the pairing step, the vehicle starts doing platooning w.r.t. its forward MIO. Once the distance to its MIO is large enough, the vehicle sends an iCLCM "SafeToMerge" message and keeps doing platooning w.r.t. its forward MIO, which will become its MIO once it has merged.

Each state defines a driving profile with a desired speed that the longitudinal controller must respect. In order to avoid being blocked indefinitely in a given state as a result of errors that may be independent of the host vehicle, the safety driver always has the option to force the state transition through the HMI.

Before producing the communication messages, the supervisor needs to retrieve information from all the other system components and write messages according to the state diagrams. Another important task of the supervisor is time-stamping each message. It was therefore important for all the vehicles and infrastructures to be well synchronized. For this purpose we used an additional GPS receiver (u-blox) supplying 1PPS (one pulse-per-second) output to continuously synchronize the whole system to the GPS time. An open-source implementation of NTP, *chrony* (http://chrony.tuxfamily.org/), was used for synchronization. One advantage of this solution was that it was designed to work even after long periods offline, i.e., without a GPS signal in our case. A sub-microsecond accuracy was attained using *chrony* with the u-blox GPS receiver.

## IV. PACPUS FRAMEWORK

The major challenge when deploying a complex system such as an autonomous vehicle is integrating the different software elements. The main target features are modularity of the components, a correct I/O definition, a simple way to configure the system at run-time, and a flexibility that will
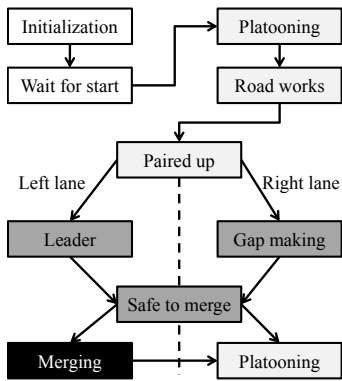
Fig. 8. State diagram of the merging scenario. The color of the blocks corresponds to different control commands. White blocks: stationary state. Light gray boxes: platooning with respect to the MIO. Dark gray boxes: platooning with respect to the forward MIO. Black boxes: changing lane. The "Leader" and "Merging" states are attained if the host vehicle is in the left lane while the "Gap making" state is only attained if it is in the right lane.

allow the system to evolve without the need to redefine the low level architecture. The PACPUS framework developed in our lab addresses these technical issues.

The Heudiasyc Laboratory began to develop the PACPUS framework in 2006. The objective was managing system integration in the CARMEN intelligent vehicle. CARMEN was equipped with a multi-sensor perception system, and an initial application for pedestrian recognition powered by the PACPUS framework was demonstrated at the 2008 IEEE Intelligent Vehicles Symposium in Eindhoven [22]. In 2013, the C++ source code of the framework and of some components (sensors interfaces, CAN connectivity, displays, etc.) was released under the free open-source license CECILL-C, a LGPL-like license. More information can be found on the website of the project (https://pacpus.hds.utc.fr).

PACPUS is a multi-platform (Windows, x86-Linux, ARM-Linux) framework similar to other robotics middleware such as ROS [23] and proprietary software such as RTMaps [24] and ADTF [25]. All of these are modular and with a data flow communication system. However, our focus has been the effectiveness of the framework at run-time. For example, the data player is based on a loading of data "on-the-fly", and data are exchanged in binary format. PACPUS is based on a component architecture that offers modularity and dynamic loading of components. Therefore, at run-time, it is possible to load plugins and instantiate new components. Each component can be launched as a single process or with other components as threads in the same application sharing the same memory segment. It is also possible to distribute the components through a network and run the system on several computers.

The class diagram in Fig. 9 illustrates the typical object architecture of a PACPUS application. The management of the components life is delegated to the *ComponentManager* class and each component provides a component factory to build the object. The key principles of the PACPUS framework are:

- A modular, dynamic, component-based architecture. Each component must provide its type at compile time and must implement a minimum life cycle:
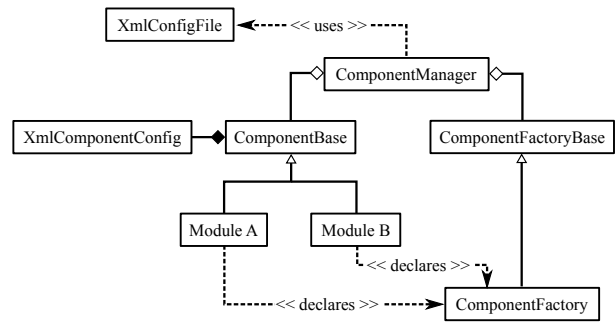


Fig. 9. Typical object architecture of a PACPUS application.

  – construction of the object: definition of the C++ constructor
  – configuration step: at run-time the component is able to process an XML node sent by the component manager which contains the parameters of the algorithm
  – an activity start function: the developer initializes the algorithm and launches the processing threads if necessary
  – an activity stop function: similar to the above
  – destruction of the object: equivalent to the C++ destructor, mainly useful for freeing memory

- An absolute time-stamping of each piece of data. Unlike in other popular robotics middleware, the time-stamp is absolute, so it is easy to retrieve the date and time of a dataset. This also solves problems of date matching in distributed systems.
- A strong typed input/output feature to exchange data between components. This mechanism is based on an event-triggered scheduling.

## V. EXPERIMENTAL VEHICLE

The experimental vehicle used by the Heudiasyc team was a fully electric car that had been modified for the purposes of academic research. We observed a safety policy whereby our vehicle was limited to a maximum speed of 50 km/h while driving autonomously.

### A. Automated control

The automated control of the vehicle was done through a dSpace MicroAutoBox prototyping hardware. This was designed to send commands through the vehicle CAN bus relating to acceleration, braking and steering wheel angle.

Several physical switches, well in reach of both the safety driver and the assistant operator, were previously added to the vehicle for safety reasons (see Fig. 10). When designing the experimental vehicle a specific mode reserved for the Heudiasyc Laboratory (UTC) was added to the (a) switch. This mode allows research teams at the University of Technology of Compiègne to access the vehicle CAN bus. The (b) switch was designed to select a driving mode, the three available modes being manual, cooperative and autonomous (see Sec. V-B). The (c) switches could be used to manually disable autonomous capabilities such as acceleration, braking or steering
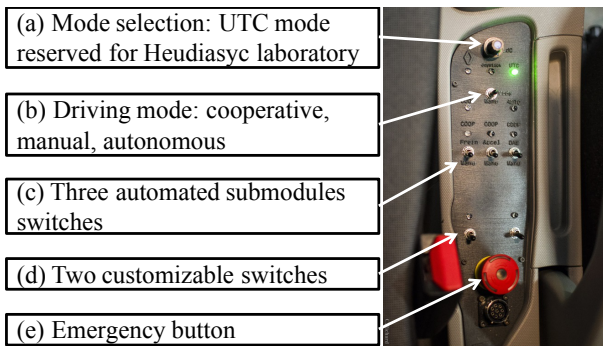
Fig. 10. Manual switches to select driving mode.

TABLE I
POSITIONING MODES WITH THEIR EXPECTED ACCURACIES [26].

| Positioning mode | Accuracy |
|---|---|
| RTK fixed ambiguities solution (RTKFIXED) | $\sim 0.01$ m |
| RTK floating point ambiguities solution (RTKFLOAT) | $< 1.0$ m |
| Pseudorange Differential Solution (PSRDIFF) | $< 2.0$ m |
| Satellite-Based Augmentation System (SBAS) | $\sim 3$ m |
| Pseudorange Single Point Solution (PSRSP) | $\sim 5$ m |

sent by the system. In this case, the car was also switched to manual mode. In cooperative mode, the driver's actions on automated sub-modules overrode the commands of the system and only set these particular sub-modules to manual mode.

## VI. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, we report several results from the data recorded during GCDC, focusing on the merging scenario, and discuss the current limitations of our solution. Because our car was limited to a speed of 50 km/h in autonomous driving mode, half of the merging heats were done at a lower speed (40 km/h). The Heudiasyc team took part only in the low speed merging heats and in the crossing scenarios. The high speed merging heats and the emergency scenario were conducted in manual driving mode. In every scenario with a speed less than 50 km/h, both longitudinal and lateral controls were done autonomously, including in the lane changing maneuver. All the code for the developed modules as well as the raw data recorded during the challenge are available for download (https://pacpus.hds.utc.fr/software-datasets/datasets/).

*a) Localization:* The Kalman filter-based localization of the vehicle was done using a NovAtel SPAN-CPT that combines inertial measurements with GNSS information received from both GPS and GLONASS satellites as well as geostationary satellites such as the European Geo-Stationary Navigation System (EGNOS). GNSS-based localization is classed into five modes with different accuracies, as detailed in Tab. I. The accuracies reported in Tab. I do not consider IMU information and are only valid in open sky conditions with several satellites in view and without multi-path. In highway scenarios, these conditions are in general satisfied, except when driving under bridges. We remarked that dead-reckoning was sufficient for overcoming these limitations during the challenge.

The best accuracy is reached when RTK corrections are available and with several satellites in line-of-sight. To fix the ambiguities in the solution, at least five GPS satellites (or four GPS and at least two GLONASS satellites should be in line-of-sight). To maintain a fixed ambiguity solution, at least five satellites, including at least two GPS satellites, are necessary. When both GPS and GLONASS satellites are used, the accuracy is improved by a factor of two. When RTK is lost, a RTK floating point ambiguities solution is obtained, but only if at least five satellites (GPS+GLONASS) are available [26].

Fig. 11 shows the distribution of positioning modes during a complete heat of the merging scenario. For more than 80% of the time, a sub-metric accuracy was obtained directly from GNSS information. The high quality localization results achieved at GCDC were, however, exceptional and only made

independently. During GCDC, the two extra customizable switches (d) were connected to two rooftop lights, one red and one green. These lights were used as visual indicators so that other participants could know whether the vehicle was in autonomous mode or not. The emergency button (e) could be used to shut down external electrical components including the MicroAutoBox and computers. When pressed, the vehicle would go into a degraded mode, the safety driver would regain the control of the vehicle but could no longer accelerate. This button was for emergency purposes only, and not for simply switching back to manual mode.

### B. Driving modes

The experimental vehicle's three driving modes are the following:

- Manual: the driver has a full control, no commands can be sent to the vehicle.
- Cooperative: the driver has control over at least one sub-system (acceleration, braking, steering). For example, the driver might be in charge of the lateral control while the longitudinal (acceleration and braking) part is done autonomously.
- Autonomous: the system manages both longitudinal and lateral controls of the vehicle.

A number of safety precautions were in force relating to the transition to non-manual mode. Switching to non-manual mode could only be done if all the following conditions were satisfied:

- the manual switches are in the correct positions
- the hand brake/parking brake is off
- the car engine is on
- the gearbox is in the D or R position
- all four doors and the car trunk are closed
- the emergency button is released
- there are no errors in the vehicle's self diagnosis
- there are no errors in the MicroAutoBox diagnosis

For the vehicle to be able to activate non-manual modes, it must be stopped or driven very slowly (less than 10 km/h, i.e., park assist speed). While driving in non-manual modes, if one of the aforementioned criteria ceased to be met, the vehicle was switched back to manual mode.

In autonomous mode, any action on the acceleration, the brake or the steering from the driver overrode the commands
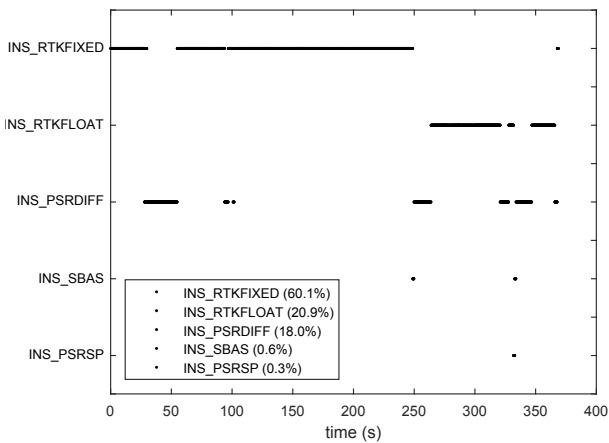
Fig. 11.   Positioning modes during a complete merging heat.

possible by the almost ideal conditions during the challenge. In practice, having high-end IMU, RTK corrections and almost 100% GNSS covering are not reasonable hypothesis. The lack of GNSS signal or the reception of multi-path signals over extended periods of time could lead to catastrophic bias in localization. The combination of GNSS localization with local perception information, e.g., lane markings or image features-based localization, could alleviate such issues [27]. Since this was not necessary during the challenge, it was not implemented.

*b) Perception:* Regarding the perception aspects, the LiDAR based approach combined with a map performed successfully in most cases. The LiDAR estimated effectively the ranges and bearing angles with respect to the vehicles immediately ahead of the host vehicle in both its own lane and the other lane. During the preparation week preceding the challenge, the perception parameters were set as follows:

- minimum number of points per cluster: 5
- lane margin width: 4 meters
- maximum distance between two points: 1 meter

These parameters resulted in no miss detections and no false positives in the merging scenario at 40 km/h, which corresponded to a safety distance between vehicles of about 22 meters. Fig. 12 shows the miss detection and false positive rates obtained during one heat for different values of the parameters.

Varying the minimum number of points per cluster, we can see from Fig. 12 (a) that when this number was small, the false positive rate increased. Small clusters composed of only one to three points, corresponding to possible noisy measurements or parts of objects, could be considered as vehicles and induce false positive detections. Starting from five points, the false positive rate dropped to 0%. However, when it reached 11, the miss detection rate started to increase. This could have been predicted from Eq. (1) as the maximum number of points hitting a two meters wide vehicle at 22 m with a 0.5º resolution LiDAR is about ten.

Regarding the lane margin width, we can see from Fig. 12 (b) that when the margin was only two meters, the miss detection rate was about 50%. When the detected object was not perfectly centered in the navigation corridor, it was filtered

out and thus not detected. The best results were obtained with a margin of four meters, approximately the width of the highway lanes. When the margin became too wide, some false positives appeared, since objects outside of the road could be confused with vehicles.

Finally, Fig. 12 (c) shows the results when varying the distance of separation below which two consecutive points are considered to be in the same cluster. We can see that this parameter had no influence on the false positive rate when the minimum number of points per cluster and the lane margin width were set up correctly. When this distance was too small, the clusters tended to be very small and were then filtered out by the minimum size of cluster criteria, leading to a high rate of miss detections. Considering Eq. (1), we can see that the distance between two consecutive points is about 0.2 meters. Therefore, if the threshold was set to 0.4 meters, then a single miss detected LiDAR point was enough to cut one cluster into two distinct clusters. Conversely, if the threshold was too high, large clusters appeared that were then filtered out by the lane margin criteria, which also led to a higher miss detection rate.

Like the localization aspects, the implemented perception module was specific to GCDC. Although it was sufficient for handling perception tasks related to platooning, it was not *per se* an obstacle detection module. Apart from the MIO and the forward MIO, our perception module would not have detected any other obstacles. In order to have more robust perception capabilities, more generic object detection and tracking algorithms should be considered as well as multi-sensor based methods [28].

*c) Control:* In terms of vehicle control, the tasks were to maintain a safety distance with respect to the forward vehicle, to perform lane keeping, and to execute a merging maneuver autonomously. In a steady platooning state at 40 km/h, the average longitudinal velocity error was 0.71 km/h with a standard deviation of 0.03 km/h, and the average distance error w.r.t the preceding vehicle was 1.07 meters with a standard deviation of 0.04 meters. This means that the host vehicle was driving at a slightly lower speed and keeping a safety distance about one meter above the minimum safety distance ($\approx 22.7$ m), which is perfectly reasonable.

To show the behavior of our longitudinal controller in the case of highly dynamic scenarios, Fig. 13 reports results recorded during the preparation week prior to the challenge. The scenario comprised three decelerating and two accelerating stages. The dotted curve represents the measured inter-vehicle distance from which the desired velocity (dashed curve) is computed following Eq. (4). The solid curve represent the actual host vehicle speed. We can see that the system had a response time of several seconds (less than 4s) to catch up to the desired speed. This may cause critical issues if the preceding vehicle brakes too hard. During the second deceleration stage, the distance between the two vehicles was about five meters below the nominal safety distance. In this particular test, the preceding vehicle accelerated and decelerated at an absolute value of about 3.8 m/s$^2$. During the challenge, the maximum allowed acceleration was, however, set to 2 m/s$^2$. In order to handle such dynamic situations properly, the controller should integrate the speed of the
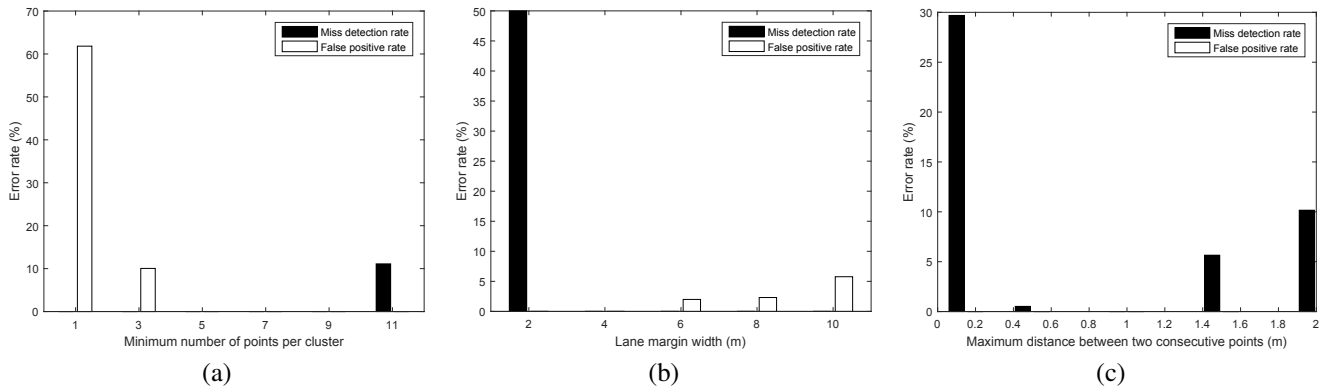
Fig. 12. Miss detection (black bars) and false positive (white bars) rates for the LiDAR based object detection module. The results are given for varying values of (a) the minimum number of points per cluster, (b) the lane margin width and (c) the maximum distance between two consecutive points.

preceding vehicle, for example, by using a sensor such as a radar. Velocity and acceleration information coming directly from the vehicle ahead through wireless CAM messages could also be used under the hypothesis than they are reliable.

In terms of lateral control, in a steady platooning state, the average lateral error was $0.29$ meters with a standard deviation of $0.18$ meters, and the errors were contained within the interval $[0.05; 0.60]$ meters. The average heading error was $-0.06$ degrees with a standard deviation of $0.27$ degrees.

Fig. 14 shows the lateral and heading error profiles with respect to the left and right lane during a real merging maneuver during the challenge. It can be seen that the lateral error was not symmetrical, but always positive. This may be because the A270 highway curves slightly to the right throughout the competition zone. The lateral error nevertheless remained within a reasonable range, since the usual lateral distance between two vehicles is about two meters. Fig. 14 also shows the steering wheel angle during the merging maneuver. The steering wheel angle is approximately 15 times the wheel angle $\delta$. We can see in the bottom graph of Fig. 14 that the steering wheel command is set up to handle discrete integer values, in terms of wheel angle resolution, one degree corresponding to approximately $0.07°$. During the whole merging phase, the angle $\delta$ remained small with an absolute value smaller than $1°$.

We remark that the errors w.r.t. the two lanes had slightly different shapes, as the two recorded reference trajectories were not perfectly parallel. On the top graph in Fig. 14, the circled line shows the lateral error sent to the lateral control. We can see a jump at $t \approx 30$ which corresponded to the start of the merging maneuver. The merging was completed in a bit less than 10 seconds at a speed of 40 km/h.

Fig. 15 shows a bird's eye view of the recorded trajectory during the lane changing maneuver in a Frenet frame. The trajectory was relatively smooth even through no path planning was implemented.

*d) Communication:* Regarding the performance of the V2V communications, Fig. 16 shows the average CAM message reception rates. During the challenge, the CAM message emission rates were set to $25$ Hz. The top graph shows the box plots for each team separately (the team IDs have been anonymized). We can see that for all the teams the median
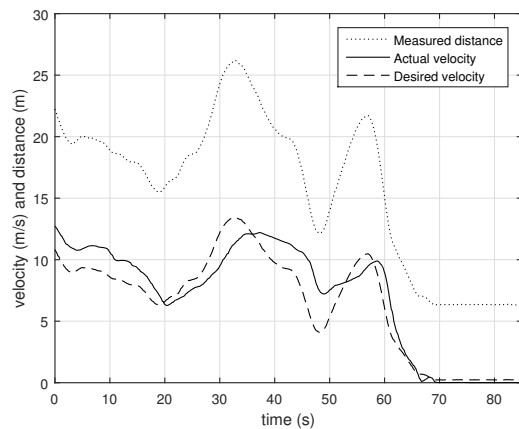


Fig. 13. Longitudinal control results. The dotted curve represents the measured inter-vehicle distance from which the desired velocity (dashed curve) is computed. The solid curve represents the actual host vehicle speed.

reception rates were above $20$ Hz. Except for team 3, more than $75\%$ of the reception rates were above $10$ Hz, which is the frequency recommended by the ETSI standards. Among the nine other vehicles, the reception rates for teams 2, 7 and 8 were very high. The other vehicles had relatively similar performances in terms of communication.

The bottom graph shows to what extent the distance w.r.t. to the host vehicle influenced reception rates. We show the box plots of the reception rates of all the vehicles for different distance ranges. The reception rates tended to decrease slightly with distance, but the difference does not appear to be significant. Also, even at a relatively close range, the reception rates could in some limited cases still be very low.

Overall, the quality of communications during the challenge was relatively good. Or, at least, good enough to manage the different scenarios in GCDC.

## VII. CONCLUSION AND FUTURE WORK

The system architecture described in this paper was successfully implemented and demonstrated within the context of GCDC 2016. Although most of the basic modules rely on fairly simple approaches, they were able to handle complex scenarios such as cooperative merging and crossing. As the
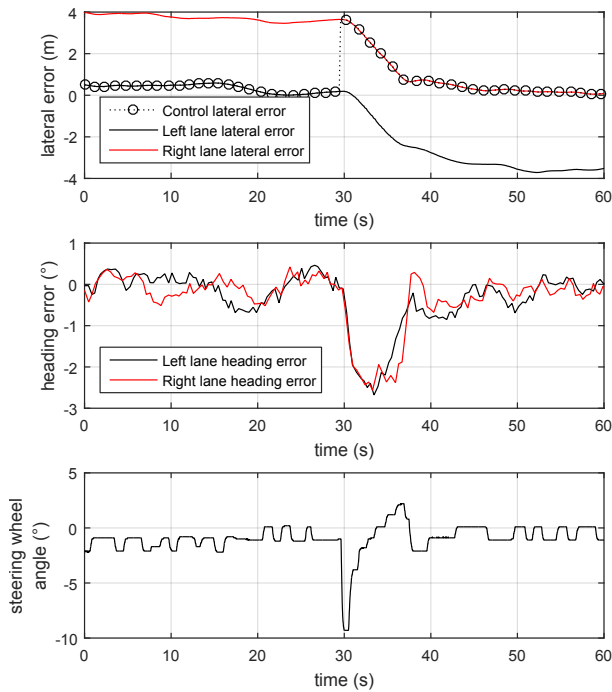
Fig. 14. Lateral errors (top graph), heading errors (middle graph) and steering wheel angle (bottom graph) during a merging maneuver. In the error figures, the black curves represent the errors w.r.t. the left lane while the red curves are the errors w.r.t. the right lane.
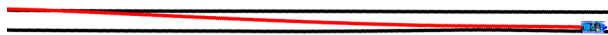


Fig. 15. Lane changing trajectory. The host vehicle is moving from left to right and changing from the left lane (top black curve) to the right lane (bottom black curve). The red curve represents its trajectory during the merging maneuver.
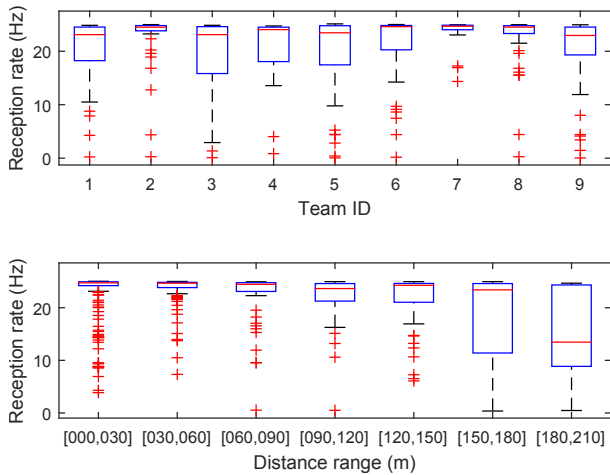


Fig. 16. Box plot of CAM messages reception rates. The top graph shows the reception rates for each team for all the merging heats. The bottom graph show the average reception rates for all the teams for different distance ranges.

solution considered in this paper remains relatively costly, one research direction for future work will be to reproduce similar results at a lower cost, both in terms of localization and perception. In particular, the use of high-definition maps and wireless communications is one way to have enhanced information for almost no additional cost. More powerful and

more robust vision-based perception algorithms and control laws will be implemented in future work. The possibilities offered by wireless communications for automated distributed understanding of driving scenes are numerous and largely unexplored. We believe that the main challenges in cooperative driving are, first, defining standardized interaction protocols for driverless cars and human driven vehicles, and, second, managing the integrity and trustworthiness of information exchanged.

## REFERENCES

[1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[2] J. Ziegler *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] G. Seetharaman, A. Lakhotia, and E. Blasch, "Unmanned vehicles come of age: The DARPA grand challenge," *Computer*, vol. 39, no. 12, pp. 26–29, Dec 2006.

[4] C. Berger and B. Rumpe, "Autonomous driving - 5 years after the Urban Challenge: The anticipatory vehicle as a cyber-physical system," in *Proceedings of IEEE/ACM International Conference on Automated Software Engineering*, Essen, Germany, Sep 2012, pp. 789–798.

[5] E. van Nunen, M. R. Kwakkernaat, J. Ploeg, and B. D. Netten, "Cooperative competition for future mobility," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1018–1025, 2012.

[6] J. Bennett, *OpenStreetMap*. Packt Publishing, Limited, 2010.

[7] D. Bétaille and R. Toledo-Moreo, "Creating enhanced maps for lane-level vehicle navigation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 4, pp. 786–798, Dec 2010.

[8] P. Bender, J. Ziegler, and C. Stiller, "Lanelets: Efficient map representation for autonomous driving," in *Proceedings of IEEE Intelligent Vehicles Symposium*, Ypsilanti, MI, USA, 2014, pp. 420–425.

[9] National Imagery and Mapping Agency, "Department of defense world geodetic system 1984: its definition and relationships with local geodetic systems," St. Louis, MO, USA, Tech. Rep. TR8350.2, Jan. 2000.

[10] Z. Tao and P. Bonnifait, "Sequential data fusion of GNSS pseudoranges and Dopplers with map-based vision systems," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 3, pp. 254–265, Sep. 2016.

[11] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Autonomous Robots*, vol. 26, no. 2-3, pp. 123–139, 2009.

[12] K. O. Arras, O. M. Mozos, and W. Burgard, "Using boosted features for the detection of people in 2D range data," in *Proceedings of IEEE International Conference on Robotics and Automation*, Roma, Italy, 2007, pp. 3402–3407.

[13] 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 Transactions on Control Systems Technology*, vol. 8, no. 4, pp. 695–708, 2000.

[14] R. Rajamani, *Vehicle Dynamics and Control*.  Springer, 2006.

[15] A. Ali, G. Garcia, and P. Martinet, "The flatbed platoon towing model for safe and dense platooning on highways," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 58–68, 2015.

[16] S. Dominguez, A. Ali, G. Garcia, and P. Martinet, "Comparison of lateral controllers for autonomous vehicle: Experimental results," in *Proceedings of IEEE International Conference on Intelligent Transportation Systems*, Rio de Janeiro, Brazil, Nov. 2016, pp. 1418–1423.

[17] J. M. Snider, "Automatic steering methods for autonomous automobile path tracking," Robotics Institute, Pittsburgh, PA, USA, Tech. Rep. CMU-RITR-09-08, 2009.

[18] R. Lenain, B. Thuilot, C. Cariou, and P. Martinet, "High accuracy path tracking for vehicles in presence of sliding: Application to farm vehicle automatic guidance for agricultural tasks," *Autonomous Robots*, vol. 21, pp. 79–97, 2006.

[19] ETSI, "Intelligent transport systems (ITS); Vehicular communications; GeoNetworking; Part 5: transport protocols; Sub-part 1: basic transport protocol," Tech. Rep. ETSI EN 302 636-5-1 v1.2.1, May 2014.

[20] ——, "Intelligent transport systems (ITS); Vehicular communications; Basic set of applications; Part 2: specification of cooperative awareness basic service," Tech. Rep. ETSI EN 302 637-2 v1.3.1, Sep. 2014.

[21] ——, "Intelligent transport systems (ITS); Vehicular communications; Basic set of applications; Part 3: specification of decentralized environmental notification basic service," Tech. Rep. ETSI EN 302 637-3 v1.2.1, Sep. 2014.

[22] F. Fayad, V. Cherfaoui, and G. Dherbomez, "Updating confidence indicators in a multi-sensor pedestrian tracking system," in *Proceedings of IEEE Intelligent Vehicles Symposium*, Eindhoven, Netherlands, 2008, pp. 156–161.

[23] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009.
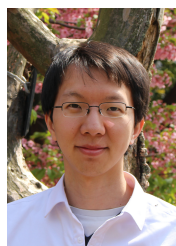
[24] B. Steux, P. Coulombeau, and C. Laurgeau, "RTmaps: a framework for prototyping automotive multi-sensor applications," in *Proceedings of IEEE Intelligent Vehicles Symposium*, Dearborn, MI, USA, 2000.

[25] J. Messner, "EB assist ADTF automotive data and time triggered framework," Tech. Rep., 2015.

[26] NovAtel, "Positioning modes of operation application note," Tech. Rep. APN-051, Oct. 2013.

[27] Z. Tao, P. Bonnifait, V. Frémont, J. Ibanez-Guzman, and S. Bonnet, "Road-centered map-aided localization for driverless cars using single-frequency gnss receivers," *Journal of Field Robotics*, vol. 34, no. 5, pp. 1010–1033, Aug. 2017.

[28] S. A. R. Florez, V. Frémont, P. Bonnifait, and V. Cherfaoui, "Multi-modal object detection and localization for high integrity driving assistance," *Machine vision and applications*, vol. 25, no. 3, pp. 583–598, 2014.

**Gérald Dherbomez** received his M.S. degree in Computer Science, Microelectronics and Automatic from The Polytech'Lille Engineering School in 2003. He is research engineer at the CNRS and he is member of the Heudiasyc laboratory, at the Université de Technologie de Compiègne, France. He works as project manager of the experimental platform PACPUS. His research topics deal with real-time and embedded applications in the field of Intelligent Vehicles, in particular, in the development and validation of Advanced Driver Assistance Systems and autonomous driving.



**Elwan Héry** is a Ph.D. student in Heudiasyc UMR 7253, a joint research unit between Université de Technologie de Compiègne (UTC) and CNRS in France. He works on cooperative localisation for autonomous and communicating cars. He obtained his engineer's degree in mecatronics in the Ecole Nationale Supérieure d'Ingénieur de Limoges (ENSIL), France, and his M.Sc. degree in algorithmic, symbolic computation and numerical optimization in the Université de Limoges, France, in 2016.



**Abderrahmen Abidli** graduated as a computer science engineer from the National School of Computer Science (ENSI) in Tunisia. He did a research internship at the Heudiasyc laboratory in which he worked on cooperative and autonomous driving. Since 2016, he is a research engineer at Navya Technology in the field of autonomous vehicles.



**Philippe Xu** received his B.S. and M.S. degrees in computer science from the Ecole Normale Supérieure de Cachan, France, in 2009 and 2011, respectively. He received his Ph.D. degree from the Université de Technologie de Compiègne (UTC), France, in 2014. Since 2015, he is an associate professor in the Computer Science department of UTC and is carrying out his research in Heudiasyc UMR 7253, a joint research unit between UTC and CNRS. His research interests cover information fusion, computer vision and machine learning applied to autonomous vehicles.
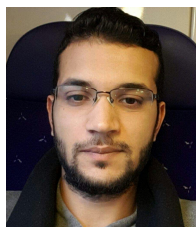


**Philippe Bonnifait** is a professor in the Computer Science department of the Université de Technologie de Compiègne (UTC) in France. He obtained his Ph.D. in automatic control and computer science at the École Centrale de Nantes in 1997. Since 1998, he has been with Heudiasyc UMR 7253, a joint research unit between UTC and CNRS. His research interests are Intelligent Vehicles, and high-integrity positioning and map-matching for autonomous navigation in structured outdoor environments.