sciendo RIGA TECHNICAL UNIVERSITY 1862
Electrical, Control and Communication Engineering
ISSN 2255-9159 (online)
ISSN 2255-9140 (print)
2018, vol. 14, no. 1, pp. 81–88
doi: 10.2478/ecce-2018-0009
https://www.degruyter.com/view/j/ecce

Development Case Study of the First Estonian Self-Driving Car, ISEAUTO

Anton Rassõlkin* (*Research Scientist, Tallinn University of Technology, Tallinn, Estonia*),
Raivo Sell (*Senior Research Scientist, Tallinn University of Technology, Tallinn, Estonia*),
Mairo Leier (*Research Scientist, Tallinn University of Technology, Tallinn, Estonia*)

Abstract **– The rapid development of intelligent control technology has also brought about changes in the automotive industry and led to development of autonomous or self-driving vehicles. To overcome traffic and environment issues, self-driving cars use a number of sensors for vision as well as a navigation system and actuators to control mechanical systems and computers to process the data. All these points make a self-driving car an interdisciplinary project that requires contribution from different fields. In our particular case, four different university departments and two companies are directly involved in the self-driving car project. The main aim of the paper is to discuss the challenges faced in the development of the first Estonian self-driving car. The project implementation time was 20 months and the project included four work packages: preliminary study, software development, body assembly and system tuning/testing of the self-driving car. This paper describes the development process stages and tasks that were distributed between the sub-teams. Moreover, the paper presents the technical and software solutions that were used to achieve the goal and presents a self-driving last-mile bus called ISEAUTO. Special attention is paid to the discussion of safety challenges that a self-driving electrical car project can encounter. The main outcomes and future research possibilities are outlined.**

Keywords **– Automotive engineering; Autonomous vehicles; Vehicle driving; Vehicle safety.**

I. INTRODUCTION

ISEAUTO [1] is the first Estonian self-driving car developed in cooperation between Tallinn University of Technology (TTÜ) and Silberauto AS. The ISEAUTO last-mile bus is designed to be a minibus that is going to operate mainly on the campus of TTÜ, therefore the speed of the car was limited to 10–20 km/h. The project started in June 2017, when the sides agreed in Tallinn to jointly develop a self-driving car which will see its public demonstration a year later. The ISEAUTO project has a range of objectives from both sides as well as a very practical outcome. The purpose on the part of the company was to get involved with the self-driving technology as the future of the automotive industry and obtain experience in manufacturing a car body structure, as it is one of the branches where the company is active. Moreover, the company also values engineering education and wants to support young talented students in their engineering studies as well as attract new students to studying engineering. The main goals of the university are to increase competence in the field of self-driving vehicles and enable interesting yet practical projects for engineering students.

The outline of the paper is as follows: Section II presents the project flowchart regarding the time stages and the main tasks for the ISEAUTO team. Section III describes a technical solution of the test platforms used in the project. Section IV discusses the safety challenges of the project. Section V describes the student involvement in project. The conclusions and the future work outlooks are presented in the last section.

II. PROJECT FLOWCHART

The project timeline is divided into four stages, the length of which varies from 3 to 5 months. The development follows the V-model development process methodology specified in the VDI2206 guideline [2] as shown in Fig. 1. Different project management tools, computer-aided design tools for mechanical, electrical and software design and implementation are used, as well as documentation and process management.

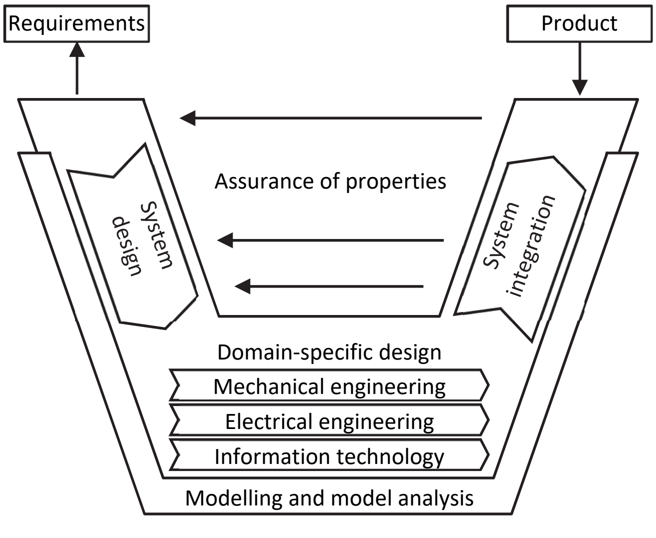

Fig. 1. V-model development process methodology.

A. Stage I

The project stated with the three-month-long Stage I, which focused on the development of the project requirements, state-of-the-art analysis and initial design task formalisation. In addition, the project management structure and tools were defined and put into action. The output of Stage I consisted of reports on the following domains:

---

* Corresponding author.
E-mail: Anton.Rassolkin@ttu.ee

- similar projects of autonomous cars [3] at universities and small and medium-sized enterprises (SMEs),
- software frameworks and architecture offered at the time,
- sensors for autonomous vehicles, their scope, availability and characteristics.

Reports were formatted according to the template and ended with a short analysis and a list of recommendations on the topics in question. All the reports are available on the project website [1]. Based on the findings of the analysis and the requirement specification, a detailed initial task was developed as a source for actual development.

B. Stage II

The next stage started with the forming of sub-teams for the project and distribution of the tasks. All the sub-teams defined the short-term and long-term goals as well as the tools and procedures for the development and documentation process. On the higher level, project coordination tools and the documentation process were defined. The stage lasted for four months and as an outcome, ~~a~~ computer simulations were produced, the initial algorithms were tested on the first test platform (a modified electric toy car for children), which was able to drive autonomously inside buildings. No safety measures were needed as the toy car did not constitute a potential risk to anyone. The main outcome consisted in the proving of the concept that the selected technology actually worked at limited conditions and that the decisions made at the beginning of the stage had been correct. In addition, simulation environments were set up and final solution simulations were able to show how the actual road will looks like and the vehicle will drive.

C. Stage III

Stage III was the most development-intensive time and lasted for 5 months, starting from January 2018. During that time, the tests were switched from the indoor toy car to medium-class outdoor robot UKU (shown in Fig. 2).

Fig. 2. Medium-class experimental robot UKU.

As it happened to be a winter and early spring on the same valuable weather information from different sensors (like LiDAR and cameras) was collected. The area where the vehicle should start driving was fully mapped with LiDARs and several tests were performed. At the end of the stage, the next test platform (a full-sized electric vehicle Mitsubishi i-MiEV) was utilised as the main testing solution. The Mitsubishi i-MiEV vehicle already had an identical control system and the final bus as well as software set-up and hardware. The result of the stage is finalised development and proof-of-concept on the same hardware as for the final car.

D. Stage IV

The last stage is a three-month period when the control system is moved to the final platform and an intensive testing period begins in the demo track. Safety and reliability tests are mostly performed with few functional updates. The summer time was also preferable for Stage IV due to safety reasons, there not being many people at the university campus at that period of time. Based on the test results, bug fixes and improvements for the algorithms and the vehicle in general were made.

III. TECHNICAL SOLUTION

There are three test platforms that were developed for the ISEAUTO project: a modified electric toy car, a civilian medium-class robot, UKU [4], and an i-MiEV test platform.

UKU is an ATV-frame-based fully electrical mobile robot for research and practical experiments in its latest configuration. UKU was built as a student project in various configurations, starting from 2007. The current electromechanical solution was developed in 2009 and the first application demo involved a street cleaning task. Since then, the robot has served as a test platform of mobile applications and a number of different control algorithms have been developed. UKU was a safe platform for Stage III, to start outdoor tests with still experiential control algorithms. For that task, UKU was slightly redesigned as regards its control system and fully reprogrammed to meet new project software architecture. The control operating system was changed from Windows to Linux, Ubuntu 16.04, which is a base operating system for the Robot Operating System (ROS) and the Autoware application developed for ROS [5].

The i-MiEV test platform was used to test the software developed for object recognition and a tune controller was used for propulsion. The propulsion system of Mitsubishi i-MiEV, designed as a highway-capable mass production electric car, requires significant optimisation. The size and parameters of the propulsion system should be verified and optimised, to provide a more stable and effective platform for future development steps of the project.

The mechanical design and body structure development has a significant impact on the performance of the vehicle. Therefore, it is highly important to consider all the physical parameters, which affect most of the vehicle behaviour on the road. A large part of the efforts on the ISEAUTO last-mile bus was devoted to the body, which was designed and developed by the project's industrial partner, Silberauto Estonia. All the moulds and panel frames have been specifically designed for the project. The picture in Fig. 3 shows one of the building stages of the main body of the ISEAUTO last-mile bus. The body design is symmetrical, meaning that the vehicle can drive in both directions without having a specific front and back side. Sidebar lights are implemented by using the RGB LED matrix where all individual LEDs are independently driven. It means that the vehicle can easily change lights from red to white at both ends. The direction indicators and other visual elements can easily be provided on the same panels. The body design of ISEAUTO also takes into account the location of sensors required for autonomous cruising.

The ISEAUTO car has been designed to be a minibus that is going to operate primarily on the territory of the university campus, therefore the speed of the car was limited to 20 km/h. Due to lack of time, the project team decided to build the self-driving car on the Mitsubishi i-MiEV trolley, which is based on a Y4F1 permanent magnet synchronous motor (PMSM) and a parallel-shaft-type two-step reduction F1E1A transmission, with an overall reduction ratio of 6.066 : 1. The car control signals were taken over from the manual control mode and replaced by the signals from the developed controller board.

Fig. 3. One of the building stages of the ISEAUTO last-mile bus.

The main challenge of control system modification was to prevent errors and faults in the electrical vehicle electronic control unit (EV-ECU). An EV-ECU sends control signals to the traction motor only if no alerts or errors have accrued in the control algorithm; at the same time connections should be placed as close as possible to sensors and actuators. For this reason, a relay box was installed to switch from manual to autonomous operation mode with one simple click. Additionally, a signal flashlight was installed to inform the surroundings that the car is cruising in the autonomous mode. Modifications that were made to the electrical diagram of the i-MiEV test platform are presented in Fig. 4.

A. Electronics and Propulsion Motor Drive

The electronics framework of the self-driving car that provides communication between the controller software with the propulsion motor drive system has three main tasks: braking, steering and acceleration. The braking task, similarly to a conventional car, consists of a parking brake and deceleration. The parking brake is used to keep the vehicle stationary during passenger boarding and absence of operation mode. To provide remote control, the manual parking brake of the i-MiEV test platform was replaced with the BMW electronic parking brake (Part ID 34436862906). The deceleration task could be solved by proper control of the electric motor; however, to provide the required safety conditions to passengers and people outside the car, deceleration motor braking should be combined with a hydraulic unit. The hydraulic unit of Mitsubishi i-MiEV is equipped with an ABS/ASC control unit and a brake pressure sensor; moreover, it is directly connected to the four wheel sensors and connected through a CAN bus line to the EV-ECU, the steering angle sensor, the angular velocity (Yaw and G combined) sensor and the indicators panel. While the speed of the self-driving car is limited [6], there are no possibilities to use regenerative braking at the low driving speeds, as had been estimated in earlier studies [7].

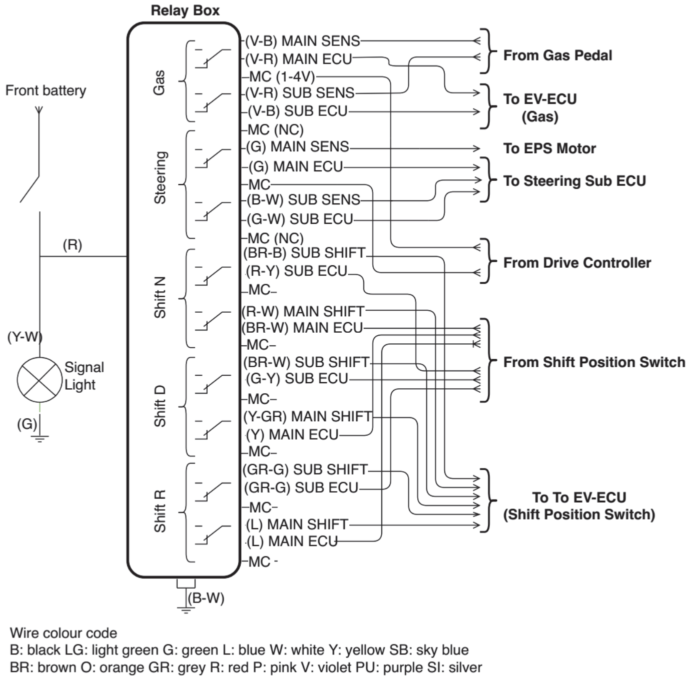

Fig. 4. i-MiEV test platform electrical modification.

The electric power steering (EPS) of Mitsubishi i-MiEV has been adopted, which improves the steering feeling in all ranges from low to high speed and enhances the vehicle stability during high-speed driving. The steering gear is mounted on the front suspension frame via the left and right bushings with inner

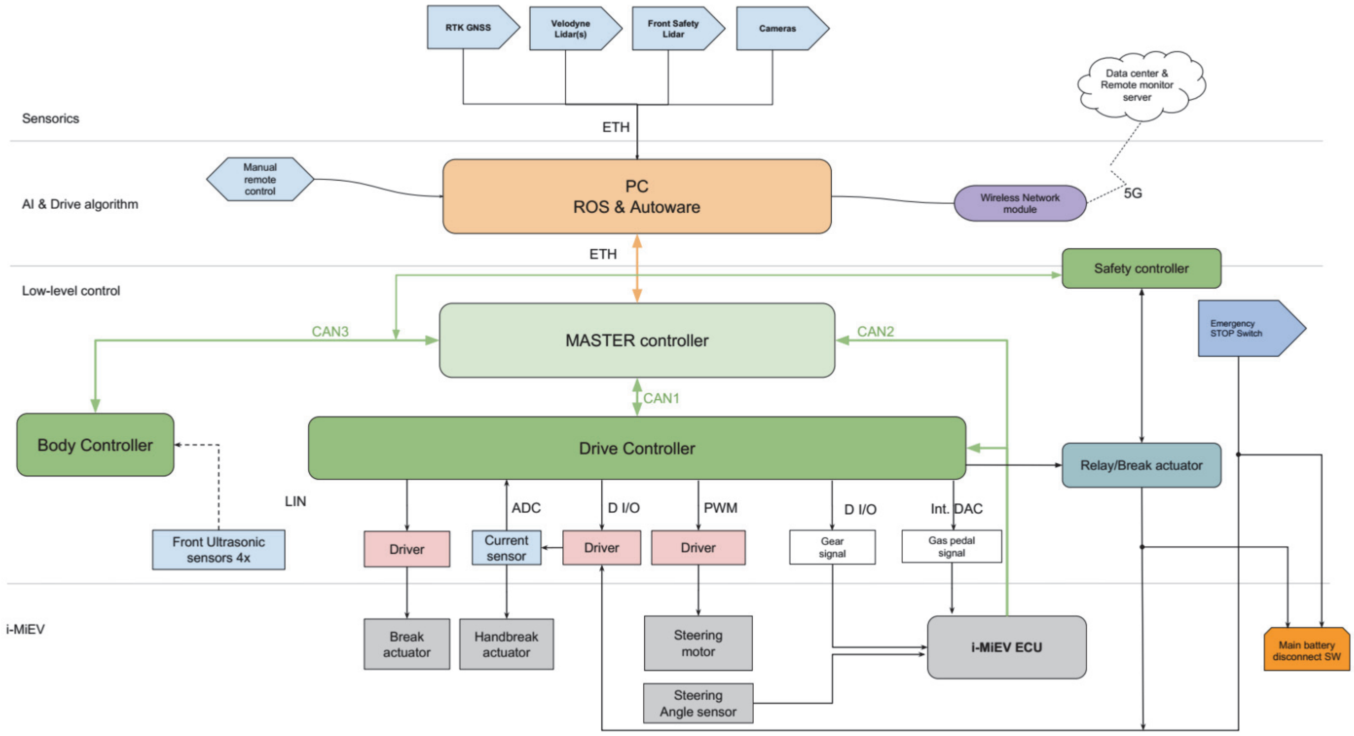

Fig. 5. Architecture of the vehicle control system.

cylinders. This mounting method achieves higher rigidity and improves the steering feeling. [8] EPS allows a light steering force during stationary steering manoeuvre or low-speed driving, which is important for a low-speed last-mile bus. In Mitsubishi i-MiEV, EPS-ECU controls the steering motor current according to the vehicle speed and the steering force of the steering wheel. In the i-MiEV test platform, the steering wheel torque sensor was replaced with a smart motor driver that directly controls the power steering motor with a Pulse Width Modulation (PWM) signal with an accuracy of 0.08 degrees. The accelerator system of Mitsubishi i-MiEV detects the accelerator pedal depression amount via an accelerator pedal position sensor. The gas pedal signal is a doubled analogue signal that varies depending on the position between 1 V and 4.5 V for the main channel and between 0.5 V and 2.25 V for the sub-channel. For i-MiEV test platform, the accelerator pedal position sensor signal was replaced with a microcontroller acceleration control signal.

In Mitsubishi i-MiEV, no gear shifting mechanism exists but the shift position detected by the shift position switch is transmitted to the EV-ECU. An original shifting mechanism has six positions: parking (P), reverse (R), neutral (N), drive (D), regenerative brake (B) and comfort (C). For the i-MiEV test platform, only four of them are used (P, R, N and D); the switching is made through a drive controller developed for test platform. The ISEAUTO car is designed to be a minibus where the passengers are travelling in a standing position, which brings additional challenges to determine an acceptable acceleration and braking force to ensure a comfortable ride to the passengers. The braking and steering of the car play a major role in the safety of a self-driving car, which means that they should meet all the safety requirements.

Mitsubishi i-MiEV is equipped with a 330 V main battery (16 kWh Li-ion) and a 12 V auxiliary battery; additionally, the test platform is equipped with one extra 12 V battery that has a higher capacity to feed additional load like PC, controllers, actuators, sensors etc.

B. Control Architecture

The architecture of the car's control is divided into four layers as described in Fig. 5. The upper sensorics layer provides input to the AI and the drive algorithm, which makes autonomous driving decisions based on this. The vehicle speed and direction commands are sent to the low-level control layer that has a mission-critical functionality to take care of the car's real control by controlling different actuators and reading feedback from the car CAN network on the i-MiEV layer.

The autonomy is achieved by running Autoware in a PC on top of the ROS that communicates with the controllers over Ethernet to minimise delays. Mission-critical controllers are divided into two layers – the master controller layer and the drive controller layer. The main task of the master controller is to forward information from and to the PC with a minimum delay. It handles communication with ROS, listens to data from the EV CAN network and communicates with the other controllers over different CAN networks. The CAN1 network is dedicated to the mission-critical drive controller that manages the gearbox, the brakes, the hand (parking) brakes, the steering wheel and the gas pedal. The CAN3 network is dedicated to other controllers that control other actuators, interior actuators and low-level sensors, e.g. the ultrasonic sensors at the door side of the car. The number of slave controllers may vary depending on the progress of the development. The software of all the controllers is based on the kernel of the real-time operating system FreeRTOS [9].

To analyse and control the management of the CAN network messages, the project team have decided to divide all the messages sent by the controllers into groups. Separate groups were defined starting from the highest priority for car velocity, braking, steering, gear shifting, ultrasound sensors, lights and doors. The last byte of each message contains a 1-byte counter that is increased each time after the message has been sent to track the messages.

Most of the controller software functionality is covered with embedded C unit tests to ensure the quality and functionality of the software. Among the most used testing frameworks, a Unity framework was selected to be used. For safety reasons, a separate safety controller has been developed, which is also presented in Fig. 5. Its General-Purpose Inputs and Output (GPIO) pins are connected to the drive controller's analogue and digital pins for real-time signal analysis to determine whether the expected signal levels are in the appropriate range. If signal is out of range, then it automatically activates the brake signal, which immediately stops the EV.

The most sophisticated low-level functionality is integrated into the drive controller. Fig. 6 depicts the controller's first PCB version, which has 12 V power supply, 2 CAN interfaces and connectors for brake, gas and steering actuators and sensors. There are PID controllers implemented for speed and steering control on the drive controller. The coefficients for the PID controllers were chosen by means of MATLAB environment simulations. The speed control PID controller was modified to exclude its integral part that is usually used to rule out a static error. Additionally, a gas pedal change limitation was added to the PID controller, which slowed down the speed reaction time for smoother speed changing. Initially, the speed signal was taken from the odometer, which provides an accuracy of 1 km/h and a small delay. Replacing the signal with an electric motor RPM, we obtained a decreased reaction time and a 55 times better accuracy to control speed. Other slave controllers have a very similar structure on the hardware and software levels as the drive controller.

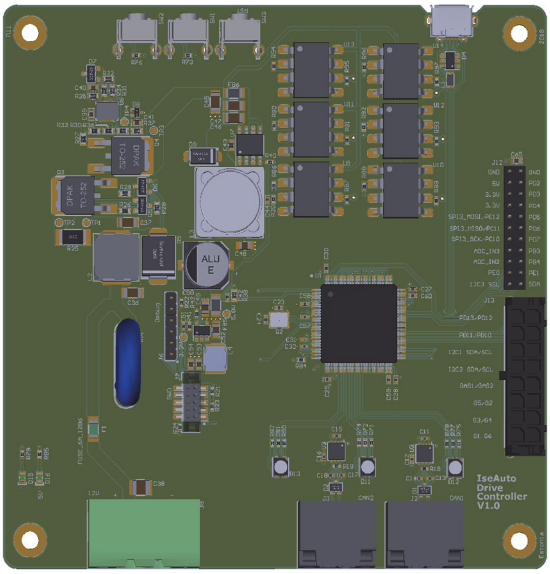

Fig. 6. Drive Controller v1.0.

The master controller acts as a central gateway between all the nodes. All the data messages between the controllers and the ROS are divided into different priority groups. This ensures that the most critical car driving messages are always handled with the highest priority and with the smallest delay. Most of the delays are caused by the speed of the EV CAN network. For example, some critical messages from the EV are sent with a frequency of 50 Hz. The ROS is configured to send driving messages with a frequency of 100 Hz. Most of the controller software functionality is covered with C unit tests to ensure the quality and functionality of the software.

The controller hardware is chosen based on the STM32 ARM, which has a wide range of different types of controllers, a user-friendly configuration interface and good previous experience. ST Microelectronics also provides special automotive controllers for future developments.

Some part of the controller programming and debugging tasks are made in cooperation with the Estonian branch of ABB.

C. Data Monitoring and Communication

The purpose of the data monitoring and communication system is to transmit the parameters and video streams from the ISEAUTO last-mile bus to the interface where those can be monitored. In addition, this system should be used in the future for sending commands to ISEAUTO. The monitoring and communication system must meet the following requirements:

- the data extraction does not influence the Linux machine that controls ISEAUTO;
- the data must be secured with encryption during the data transfer;
- ISEAUTO's visualization must present the basic parameters (such as alert indications, speed, weather conditions etc.), live camera streams and LiDAR point cloud streams.

The system is limited by the following factors: the cost of the hardware components; the computing power of the computer (hereafter referred to as the 'control computer') which is intended to control ISEAUTO; the maximum upload speed provided by 4G (theoretical maximum: 150 Mbps; the real-life expectation is 25 Mbps); it is not possible to use servers which are not owned and managed by TTÜ; it is possible to use a limited number of Internet of Things (IoT) platforms free of charge.

The Real Time Streaming Protocol (RTSP) was chosen for the ISEAUTO project because it enables switching from the Transmission Control Protocol (TCP) to the User Datagram Protocol (UDP) with the purpose of providing a better streaming experience. The UDP is delivering media data during the shortest time range, since it will not spend extra efforts on fixing errors related to the delivery, yet the UDP transport protocol might be blocked by some firewalls. The RTSP is a stateful protocol and is capable of tracking simultaneous media sessions. The ROS-supported video_stream_opencv uses this protocol. However, delivering of media data is not accomplished by means of the RTSP protocol. In order to ensure transmission of media files, RTSP servers rely on the

Real-Time Transport Protocol (RTP) and the Real-Time Transport Control Protocol (RTCP).

ISEAUTO's main computer is running on Linux Ubuntu 16.04. The ROS is used for sensing and integration of control devices. The Autoware application is used on top of the ROS for urban autonomous driving. Data and video streams are extracted from ISEAUTO's control system. The Autoware application is used on top of the ROS for urban autonomous driving. Autoware supports the following functions: 3D localisation, 3D mapping, path planning, path following, acceleration control, brake control, steering control, data logging, car detection, pedestrian detection, object detection, traffic signal detection, traffic light recognition, lane detection, object tracking, sensor calibration, sensor fusion, cloud-oriented maps, connected automation, smartphone navigation, software simulation and virtual reality. The main structure of the Autoware application is presented in Fig. 7.

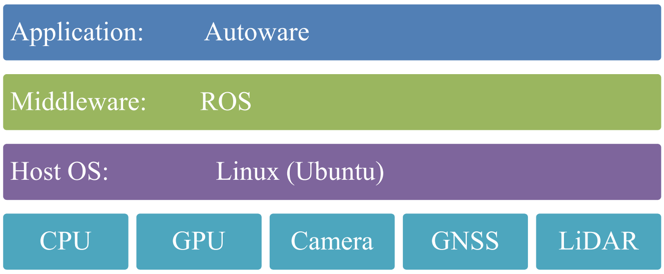

Fig. 7. The structure of Autoware application.

Different options for creating the desired user interface were compared and data specification was performed. IoT platforms such as Cumulocity and Kaa offer great functionalities to build visualisation and online data logging services for the ISEAUTO project. However, due to the limited time for creating the dashboard, the webpage option was chosen as more reasonable. Furthermore, the webpage could be used as a front end in further development where IoT platforms are used. The data provided with the webpage user interface are as follows: the location (LiDAR and GPS) and speed of the vehicle, video from the camera, the state of charge of the main battery, readings form the ultrasonic sensors.

Autoware has integration with different object detection methods [10]–[13] and it is important to evaluate how well these methods perform in a safety-critical application with different light and weather conditions. Object detectors use a trained model to detect objects on an image or a video frame. Training a model is a method of supervised learning that requires a considerable amount of labelled data. Usually the context where people need to detect objects is in some way specific and a self-trained model could have an advantage when designed for a specific task and trained with relevant data.

Next, object detection methods were compared: the Fast Region-Based Convolutional Neural Network (Fast-RCNN), the Single-Shot Detector (SSD) and You Only Look Once v2 (YOLOv2). To understand how reliably the available object detection methods can detect cars and pedestrians, an analysing and evaluating performance method was developed for different object detectors in the autonomous self-driving car context. The general idea of the developed method is using our own benchmark dataset containing relevant data and relevant, context-specific metrics to measure the performance of different methods. Firstly, scenes on the planned route were recorded at different weather and light conditions. Six scenes that provide a good sample set were selected. Next, ground truth data were created by annotating selected scenes, each object detector ran on each annotated scene and gathered performance statistics by comparing the output to the ground truth, using context-specific metrics. Finally, the statistics of different object detectors and the annotation tool used to visualise prediction of object detectors together with ground truth data were compared to get a better understanding of the statistics. YOLOv2 [13] showed the best performance and efficiency and the results clearly indicate that none of the analysed object detectors is able to detect objects with the reliability required for safety-critical applications like self-driving cars. To guarantee safety, it is not enough to rely only on the object detector but it is necessary to use it in combination with other sensors that can fill in the weak spots.

IV. VEHICLE SAFETY

Autonomous vehicles have made significant advances in recent years. To become more suitable for practical applications, they are required to operate safely and reliably even under baffling driving conditions. There is a lot of debate nowadays about ethical considerations related to autonomous vehicles, but along with that, EVs represent a completely different technology compared with traditional internal-combustion engine vehicles, which means that different safety hazards related to the characteristics of high-power electric equipment should be presented. Standards already exist for the construction of such vehicles in terms of reducing the potential risk to the passengers and the rescue team who could be exposed to hazards. [14], [15] However, ISO 26262 does not address unique electrical and/or electronic systems in special-purpose vehicles such as vehicles designed for drivers with disabilities. This is why different risks associated with the EV technology must be carefully assessed and taken into account [16]:

- electric system safety;
- functional system safety;
- battery charging safety;
- vehicle maintenance, operation and training.

Electric system safety includes protection against electric shocks that can be divided into protection against direct or indirect contacts. This means all the parts and the frame of the EV has to be protected against contact with persons inside or outside the vehicle. Functional system safety is mainly concerned with hardware failures and software bugs [17]. This means that the EV-ECU must ensure reliable and safe operation of the vehicle in case of system warnings, avoid possible damage through excessive torque, overcurrent or fierce accelerations, provide emergency disconnection of devices, provide fail-safe operation (incl. cases of power surge prevention, frame faults, electromagnetic compatibility etc.). The battery is the most critical item on board the EV and represents the following potential hazards [16]: electrical, mechanical, chemical and explosion hazards. There have been numerous real-world examples of EVs catching fire after a crash and in the garages where they were being stored; in some cases, this may have occurred while the vehicle was being charged. [14] To rank the risk factors and determinate the risk level, a risk analysis for the ISEAUTO last-mile bus was provided.

The ISEAUTO last-mile bus will operate in the campus of TTÜ, including parking and pedestrian areas. That means that an obstacle should be detected at a very short distance. It is difficult to have an accurate and fast prediction of the behaviour of pedestrians around the car in order to react properly. In case of a stop and a recalibration of the sensors, situations can occur when a pedestrian is not detected properly due to overlapping with another pedestrian. In case of a sudden reaction of a person, the system must be able to react and remain in a safe state. This holds also for people lying on the ground or small animals. After all, these are not imaginary but fully realistic situations; in case a collision occurs, the system has to be able to stop so that no injury occurs [18]. Next, sensors are used in the ISEAUTO project to observe the environment and for mapping, localisation and navigation:

- LiDAR Velodyne VLP-16 ×2;
- ultrasonic sensors at the front and back ×8;
- ultrasonic sensors at the door side ×6;
- a short-distance radar;
- cameras ×8;
- an RTK-GNSS;
- an IMU sensor.

[19] presents a statistical analysis of the number of miles that autonomous vehicles must travel in order to produce precise estimates of the crash rate for autonomous vehicles based on observational data. To demonstrate that fully autonomous vehicles have a fatality rate of 1.09 fatalities per 100 million miles (with a reliability of 99.9999989 %) with a probability confidence level of 95 %, the vehicles would have to be driven 275 million failure-free miles (ca. 440 million km). With a fleet of 100 autonomous vehicles that are test-driven 24 hours a day, 365 days a year, at an average speed of 25 mph (40 km/h); this would take about 12.5 years [19], which makes 25 years considering the speed limit of ISEAUTO, 20 km/h. Due to the lack of time, Stage IV of the ISEAUTO project is mainly dedicated to safety and reliability tests.

The introduction of EV and an autonomous vehicle requires compliance with security rules that are inherent for the electrical drive, people who are professionally working in this area must be trained on the subject of how to react in case of an accident. For this reason, during the operation of ISEAUTO, a specially trained operator will accompany the last-mile bus, also for the purpose of vehicle safety.

V. STUDENT INVOLVEMENT AND MOTIVATION

Building a self-driving car is an undertaking that requires multidisciplinary collaboration from mechanical and electrical engineering, mechatronics, electronics, computer systems and software engineers. Thus, participation in such a project helps to develop various important skills such as cross-discipline communication, working on large projects, and understanding large and complex legacy systems. There are currently fifteen students involved in various aspects of software development, eleven students in electronics and low-level microcontroller software development.

Integration of the project work into the studies is taking place in two main ways: final theses and project courses. The achievement is supported by ten final theses (4 B. sc. and 6 M. sc. theses), students pursue their theses related to various aspects of the ISEAUTO self-driving car.

VI. CONCLUSIONS AND FUTURE WORK

It is important to highlight that the ISEAUTO self-driving car is an interdisciplinary project that includes contribution from different fields. Several local and international companies are following the project progress and discussing possible cooperation plans. The final design of the ISEAUTO last-mile bus is presented in Fig. 8.

Competence and know-how about autonomous vehicles and self-driving algorithms are being developed at the university and valuable practical experience for future engineers in Estonia is provided. A very important deliverable is a multi-purpose educational and research platform for students at all levels of education. Multidisciplinary (electrical propulsion, autonomous driving, software development etc.) courses, students' final theses or other project-based educational activities can rely on it. Many fresh ideas and features proposed by students were applied during the project implementation, some of which will be used in future projects.

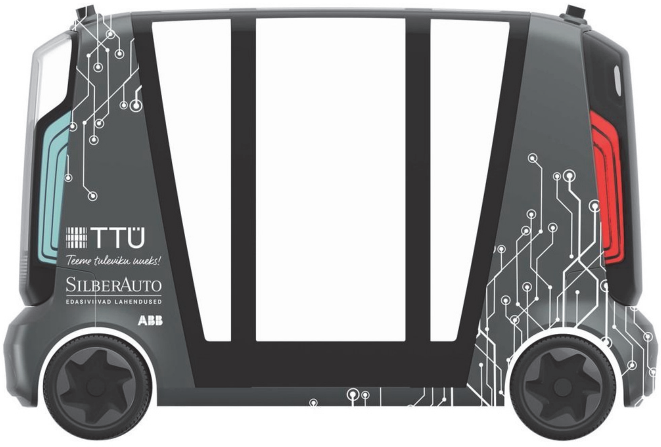

Fig. 8. Final design of ISEAUTO last-mile bus.

The long-term objective of the ISEAUTO project is to establish a smart city testbed at a university campus where different types of studies about future urban mobility can be conducted. The smart city testbed is a real-life environment in which self-driving cars, delivery robots and smart infrastructure objects are placed. The ISEAUTO last-mile bus can be a base for the Vehicle-to-Everything (V2X) platform, a vehicular communication system that incorporates other more specific types of communication as V2I (Vehicle-to-Infrastructure), V2V (Vehicle-to-Vehicle), V2P (Vehicle-to-Pedestrian), V2D (Vehicle-to-Device), V2G (Vehicle-to-Grid), or any other entity that may affect the vehicle. Many stakeholders (e.g., 3GPP, IEEE) have already announced the draft standards or technical reports that detail new opportunities of V2X-enabled vehicles for the cellular and automotive industries [20]. V2X is able to warn drivers about dangerous situations and intervene through automatic braking or steering if the driver is unable to avoid an accident; besides, platooning (road-trains) and highly automated driving can reduce travel time, fuel consumption and CO_2 emissions while improving road safety and traffic efficiency [21].

ACKNOWLEDGEMENT

The project is supported by Tallinn University of Technology and SilberAuto joint grant SS419.

REFERENCES

[1] ISEAUTO – TTÜ & Silberauto AS. [Online]. Available: http://iseauto.ttu.ee/en/mainpage/ [Accessed: 4 Jul. 2018].

[2] VDI-Fachbereich Produktentwicklung und Mechatronik, *VDI 2206 Design Methodology for Mechatronic Systems*. Düsseldorf, Germany, p. 118, 2004.

[3] R. Sell, A. Rassõlkin, M. Leier, and J.-P. Ernits, “Self-Driving Car ISEAUTO for Research and Education,” in *19th International Conference on Research and Education in Mechatronics (REM 2018)*, 2018.

[4] S. Seiler and R. Sell, “Semantics on Mobile Robot Algorithms Development,” in *Proc. 7th International Conference Mechatronic Systems and Materials (MSM 2011)*, Kaunas, Lithuania, 2011, vol. 7.

[5] Autoware.ai. [Online]. Available: http://autoware.ai/ [Accessed: 4 Jul. 2018].

[6] A. Rassolkin, L. Gevorkov, T. Vaimann, A. Kallaste, and R. Sell, “Calculation of the Traction Effort of ISEAUTO Self-Driving Vehicle,” in *2018 25th International Workshop on Electric Drives: Optimization in Control of Electric Drives (IWED)*, 2018, pp. 1–5. https://doi.org/10.1109/iwed.2018.8321397

[7] V. Vodovozov, A. Rassõlkin, N. Lillo, and Z. Raud, “Energy Saving Estimates for Regenerative Braking and Downhill Driving of Battery Electric Vehicles,” in *14th Biennial Baltic Electronics Conference*, Oct. 6–8, 2014, pp. 237–240. https://doi.org/10.1109/bec.2014.7320600

[8] MY, *Mitsubishi i-MiEV Service Manual, Technical Information Manual & Body Repair Manual*, 2013.

[9] *FreeRTOS – Market Leading RTOS (Real Time Operating System) for Embedded Systems With Internet of Things extensions.* [Online]. Available: http://www.freertos.org/ [Accessed: 5 Jun. 2018].

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object Detection With Discriminatively Trained Part-Based Models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010. https://doi.org/10.1109/tpami.2009.167

[11] R. Girshick, “Fast R-CNN,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448. https://doi.org/10.1109/iccv.2015.169

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single Shot MultiBox Detector,” in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, M. Welling M. (eds.). Springer, Cham, 2016, pp. 21–37, 2016. https://doi.org/10.1007/978-3-319-46448-0_2

[13] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognition (CVPR)*, Jul. 2017, pp. 6517–6525. https://doi.org/10.1109/cvpr.2017.690

[14] S. O’Malley, D. Zuby, M. Moore, M. Paine, and D. Paine, “Crashworthiness Testing of Electric and Hybrid Vehicles,” in *24th Enhanc. Saf. Veh. Conf.*, 2015, no. 15-0318, pp. 1–16.

[15] *ISO 26262-1:2011: Road vehicles -- Functional safety*, p. 23, 2011.

[16] P. Van den Bossche, “Safety Considerations for Electric Vehicles,” in *12th International Electric Vehicle Symposium (EVS-12) and Electric Vehicle Exposition*, 1994.

[17] R. Mariani, “An Overview of Autonomous Vehicles Safety,” in *2018 IEEE International Reliability Physics Symposium (IRPS)*, 2018, p. 6A.1-1–6A.1-6. https://doi.org/10.1109/irps.2018.8353618

[18] M. Kocsis, N. Susmann, J. Buyer, and R. Zollner, “Safety Concept for Autonomous Vehicles That Operate in Pedestrian Areas,” in *2017 IEEE/SICE International Symposium on System Integration (SII)*, 2017, pp. 841–846. https://doi.org/10.1109/sii.2017.8279327

[19] N. Kalra and S. Paddock, “Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?,” *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, Dec. 2016. https://doi.org/10.1016/j.tra.2016.09.010

[20] K. Bian, G. Zhang, and L. Song, “Security in Use Cases of Vehicle-to-Everything Communications,” in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Sept. 24–27, 2017, pp. 1–5. https://doi.org/10.1109/vtcfall.2017.8288208

[21] J. Lianghai, A. Weinand, B. Han, and H. D. Schotten, “Applying Multiradio Access Technologies for Reliability Enhancement in Vehicle-to-Everything Communication,” *IEEE Access*, vol. 6, pp. 23079–23094, 2018. https://doi.org/10.1109/access.2018.2829606

Anton Rassõlkin received the Ph. D. degree in electric drives and power electronics from Tallinn University of Technology in 2014. His main research interests are in the field of electric drives and their control systems as well as in the fields of electrical machines and electric transportation.

He works as a Research Scientist at the Department of Electrical Power Engineering and Mechatronics at Tallinn University of Technology.

Department of Electrical Power Engineering and Mechatronics, Tallinn University of Technology, Ehitajate tee 5, 19086 Tallinn, Estonia.

E-mail: Anton.Rassolkin@ttu.ee
ORCID iD: https://orcid.org/0000-0001-8035-3970

Raivo Sell graduated from Tallinn University of Technology with a doctoral degree in product development in 2007. His research interests cover early design issues on mobile robotics and self-driving car autonomous driving.

He works as a Senior Research Scientist and as a R&D manager at a robotic-design-oriented SME in Estonia.

Department of Mechanical and Industrial Engineering, Tallinn University of Technology, Ehitajate tee 5, 19086 Tallinn, Estonia.

E-mail: Raivo.Sell@ttu.ee
ORCID iD: https://orcid.org/0000-0003-1409-0206

Mairo Leier received the Ph. D. degree in computer systems from Tallinn University of Technology in 2016. His research interests include embedded systems and computing, low-power wireless networks and IoT solutions.

He works as a Research Scientist at the Department of Computer Systems at Tallinn University of Technology.

Department of Computer Systems, Tallinn University of Technology, Akadeemia tee 15A, 12618 Tallinn, Estonia.

E-mail: Mairo.Leier@ttu.ee
ORCID iD: https://orcid.org/0000-0002-9878-4477