

RESEARCH ARTICLE | JANUARY 29 2024

Robot bus low-level control system transformation to an open-source solution

Heiko Pikner ; Raivo Sell; Junyi Gu



AIP Conf. Proc. 2989, 020004 (2024)

<https://doi.org/10.1063/5.0189277>



CrossMark

AIP Advances

Why Publish With Us?



25 DAYS
average time
to 1st decision



740+ DOWNLOADS
average per article



INCLUSIVE
scope

[Learn More](#)

Robot Bus Low-Level Control System Transformation to an Open-Source Solution

Heiko Pikner,^{1, 2, a)} Raivo Sell,^{1, 2, b)} and Junyi Gu^{2, 1, c)}

¹*Department of Mechanical and Industrial Engineering, Tallinn University of Technology, 12616 Tallinn, Estonia.*

²*FinEst Centre for Smart Cities, Tallinn University of Technology, 12616 Tallinn, Estonia.*

^{a)}Corresponding author: heiko.pikner@taltech.ee

^{b)}raivo.sell@taltech.ee

^{c)}junyi.gu@taltech.ee

Abstract. This paper presents an approach to transfer the whole set of low-level control systems from one robot bus i.e. autonomous shuttle to another one that has different specifications in electronics and mechanical perspectives. In this work, we executed a series of experiments to test the reliability and safety of the autonomous shuttle after transferring the critical control systems related to the steering and brakes into the shuttle. To fulfill the requirements to register an autonomous shuttle as a legal vehicle on the road in Estonia, we carried out several vital tests of the shuttle's low-level control system. For example, we manually disconnected the different subsystems to simulate the sudden failure to check whether the shuttle acts with the corresponding protocols (i.e., when the steering Control Area Network fails, the shuttle should initiate the braking and cut off the high-voltage power). This paper proves the possibility of transferring the low-level control systems between the different models of the autonomous shuttle without risking encountering safety/reliability-related issues. Our open-source solution will be helpful for the practical promotion and commercialization of autonomous shuttles in the future.

INTRODUCTION

Recently, the development of automated vehicles (AVs) has led to the hope of a real driverless society with preferable performance in transportation efficiency, traffic safety, and energy saving. Regarding the evolution of AVs, one of the most well-known pioneering cases happened in the 1950s; General Motors embedded the automation system along the roads, not in the car because of the technical limitation at the time, but achieving the imagery of autonomous driving as the same [1]. The actual AVs started becoming a reality in the new century. The ARGO vehicle [2] performed a more than 2000 km driving test on an Italian highway in 1998, marking the beginning of the history of driverless vehicles. The modern definition of the AV is relying on sensors to perceive the surrounding environment and computer technologies to make decisions; the vehicles accord with this definition, first proved the practical usages in the DARPA Grand and Urban Challenges in 2005 to 2007 [3]. The blossom of AVs happened after the 2010s, and many companies and research groups have invested a large number of resources into commercialized technologies and experimental platforms [4] [5].

For the commercialized technologies, Advanced Driver Assistance System (ADAS) is one of the most successful technologies widely used in commercial vehicles to provide basic assisted functions such as distance control, lane keeping, and collision warning. ADAS represents a research direction mainly based on the objects perceiving to make decisions; corresponding techniques have significantly developed in recent years because of the breakthrough in the sensor industry and computers' processing capability. Light Detection and Ranging (LiDAR) sensors and computer vision technologies help vehicles to ignore the limitations of weather conditions and to make precise detection and classification of objects.

Compared with the mature technologies that were mainly adopted in the industry, experimental autonomous driving platform is the topic that has attained broad interest in research groups. [6] and [7] are examples that test the autonomous driving algorithms on vehicles for civil usage. A Low-speed AV shuttle, aka robot bus, is another attempt to exploit the practical potential of AVs in the real world. There is a possibility that the real-traffic deployed AV shuttles will reshape human transportation habits. Specifically, we would like to present the iseAuto shuttle (left one in Figure 1), which was designed and developed by the Autonomous Vehicles research group at TalTech, Estonia [8] [9].

Since the concepts of AV first appeared in research communities, reliability and safety are always the focus of AV-related technologies. For a human driver, the sentience and brain play the roles of sensors and computers to perceive the environment and make decisions. The controlling of the steering wheel and brake/throttle paddles by hands and feet guaranteed the vehicle's safety. For autonomous vehicles, a tremendous amount of research has focused on the perception-decision (sensor-computing) aspect, which commits to a comprehensive understanding of the environment



FIGURE 1. The iseAuto (left) and Navya (right) shuttles.

and flawless decision-making. However, in some perspectives, low-level control systems are more essential to AV's safety than the perception-decision stage. Very little tolerance for mistakes in the AV's critical steering, speed, and brake control systems. Therefore, the failure-proof and accuracy tests of the AV's low-level controlling system are necessary before deploying the vehicles into real traffic.

A common autonomous driving testing platform choice is commercial vehicles because their suspension, car frame, and other mechanical components have been tested and proven. However, several electronic modifications are necessary for the computer to operate the vehicles. In order to achieve the full range of autonomous features on a commercial vehicle, Wei et al. [10] implemented the multiple actuation/electronic control modules in a Cadillac SRX for brake, throttle, steering, and transmission shifting systems. For the low-speed AV shuttles, the requirements of the controlling systems are different as they do not have the steering wheel and brake/throttle paddles inside the vehicles. The manual controlling system mainly relies on the joystick controller, and the telecontrol uses the simulated steering wheel. Therefore, the whole low-level control system is heavily customized for each vehicle. One of the most important contributions of this work is transferring the low-level control system from the AV shuttle iseAuto [11] to the Navya AV shuttle (right one in Figure 1). The French-made self-driving AV shuttle Navya is a mature product on the AV shuttle market and has also been piloted previously on Estonian roads [12]. However, by the time the pilot was over, the vehicle's software was outdated, and the contract with the manufacturer had ended. The further reliability and performance testing for the Navya AV shuttle prove it is possible to migrate iseAuto's low-level control system to another type of AV shuttle with different hardware specifications.

LOW-LEVEL CONTROL SYSTEM TRANSFORMATION

Since iseAuto [13] and BoxBot [14] have been built by our team, we planned to transfer our low-level control system to an open-source solution that can be easily adapted to other types of autonomous vehicles. Moreover, a universal and modular low-level control solution is helpful for the promotion and large-scale deployment of autonomous vehicles. The tests and experiments were carried out on the Navya shuttle.

Before modifications, the self-driving shuttle Navya could drive autonomously on a pre-defined route when an expensive analysis and assessment process had to be carried out. The shuttle supplier recorded and edited the 3D LiDAR map and the driving path based on proprietary models and software [15]. Therefore deployment on a new route or even changing existing routes requires the arrival of a team from the vehicle manufacturer.

The existing self-driving shuttle transformation to an open-source solution involves multiple design steps. The vehicle manufacturer has yet to publicly release information about the vehicle's performance and technical solutions. The current shuttle was not capable of autonomous driving but was controlled over a joystick.

The first step involves checking and mapping the existing low-level architecture. Original control computers and their data connections to low-level vehicle systems were first located. The communication with the low-level vehicle system is based on the Controller Area Network (CAN). CAN is an existing multi-master broadcast serial bus communication protocol for connecting Electronic Control Units (ECU) in automotive applications [16]. In addition, the vehicle has an Ethernet network used by the two control computers to communicate with each other and with higher-level sensors, such as LiDAR sensors.

The second step involves logging the CAN message from all three identified networks. Each message has a unique

CAN ID for identification purposes. CAN network speeds were determined by trying different standard settings. A separate in-house built gateway controller was used to detect the direction of packets. For example, the packets sent out by the original control computer for each CAN network were mapped.

The third step involves identifying the data in the data field of the CAN packets. SavvyCAN DBC files store definitions for how data are formatted on the bus. It is possible to turn the raw data into parameters like RPM, odometer readings, and more [17]. Important parameters were found by changing them using the existing joystick or touchscreen and monitoring the changes in the CAN packets sent out by the original control computer. The existing data specified ranges, such as the minimum and maximum steering angle, speed, and other signals, were documented. Also, it is necessary to find feedback for each vital signal, which allows the use of a regulator such as a Proportional Integral Derivative controller (PID), and the execution of the command can be checked.

The last step involves sending and receiving all the messages needed to control the shuttle, as shown in Figure 2. An existing in-house developed Master controller is used [13]. New software layers are added to the Master controller for sending and receiving new vehicle-specific messages. Three CAN buses are identified and connected to the Master controller. CAN1 is for traction and battery, CAN2 is for steering, and CAN3 is for body-related systems. Also, a new control computer with open-source software (Ubuntu, ROS, Autoware) was added. A new Ethernet network was built to connect the new control computer, the Master controller, existing lidars, cameras, and a mobile internet access point. After that, the shuttle was able to drive, and work continued on tuning and testing the vehicle for autonomous driving.

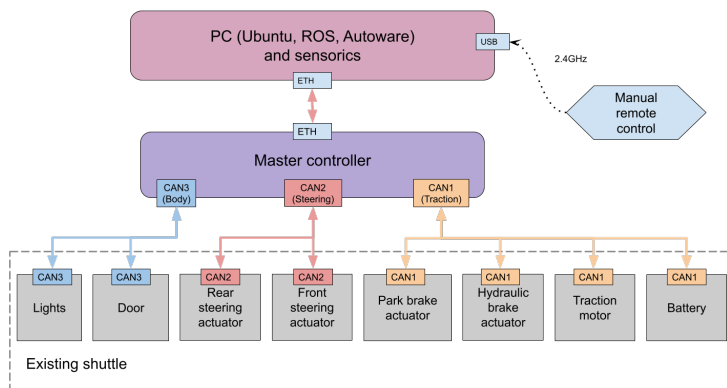


FIGURE 2. Updated hardware architecture for the shuttle.

EXPERIMENTS

Designing a safe and reliable solution requires several experiments. The first step involves implementing the steering system control signals for the Master controller. The range of the control and feedback signals has to be determined in this step. Logging data will be saved for further analysis. For example, Figure 3 shows that the control signal sent by the original control computer takes the steering system's movement speed into account. Both the front and rear axles of the vehicle can turn. Values of the rear and front axle control signals differ in sign. In addition, we found a solution to implement an independent control of the front and rear axles, which would be helpful in various future experiments. Figure 4 shows steering signals sent out by a Master controller. Tests were performed to measure how quickly the axle could imitate the control signal changes. As expected, the maximum axis movement speed is constant, and when it is reached, the actual position of the axis lags behind the required one.

Steering signal processing in the Master controller is straight forward. The UDP packet sent by ROS contains the desired steering angle, which was converted to degrees from radians, then a vehicle-specific CAN packet is later formed. The position signals of both axles were forwarded to the ROS as feedback.

The next step involves implementing the traction motor speed and control signals. The traction motor engine control unit (ECU) waits for a status signal, which can be in "use" or on "standby." Controlling the speed of the traction motor and braking is more complicated. The vehicle has a hydraulic brake and an electric handbrake that applies when the shuttle stops, as shown in Figure 5.

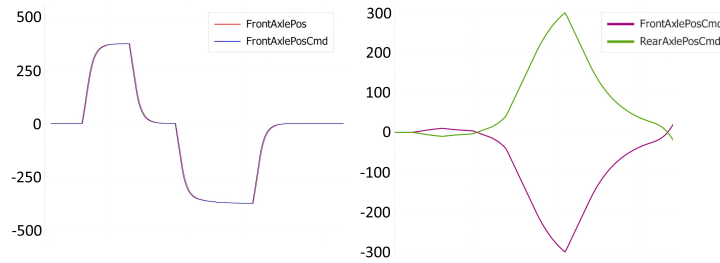


FIGURE 3. Steering signals with the original control system. The front and rear axle control command values differ in sign.

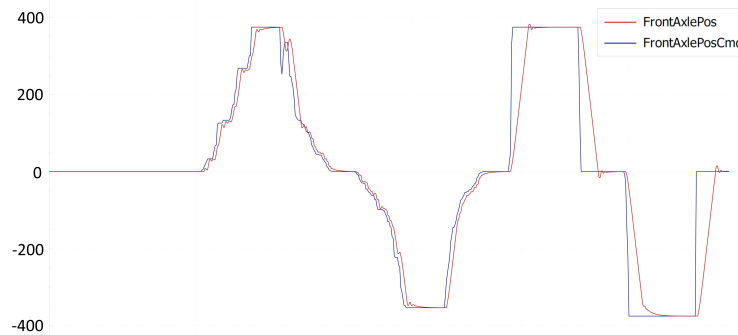


FIGURE 4. A custom Master controller sends out steering signals. Slow and high-speed movement is requested.

Examination of the packets and analysis of the logs shows that the corresponding ECU controls the brakes through the engine speed signal. Therefore, inverting the engine speed signal causing the vehicle to move in reverse. When the engine speed signal reaches zero, the hydraulic brakes applied first, and when the shuttle stops, the handbrake is also applied.

Speed signal processing inside the Master controller is more straightforward than the steering signal. The UDP packet sent by ROS contains the desired speed signal from a vehicle-specific CAN packet. Also, the traction motor control signal is a one-byte flag and was processed similarly to the iseAuto gear signal. The engine speed request sent by the Master Controller is shown in Figure 6. As expected, the speed of the traction motor follows the input signal. The Master Controller forwards the speed feedback signals to the ROS to allow basic telemetry and speed regulation.

RESULTS AND DISCUSSIONS

The new control system has passed all initial tests. Testing was took place according to the plan to ensure the shuttle's safety. First, the electronic control modules were tested separately on the bench under monitoring. Analyzed results are helpful for correcting data and module's operations. Second, the ECU was mounted on the shuttle and tested when the vehicle was lifted from the ground. Third, the driving tests of the shuttle were took place on an empty street to find out critical bugs and improve the software.

One example of discovered and fixed safety critical bug in the Master Controller software is the speed command packets were processed incorrectly. As a result, the shuttle brakes suddenly, which is dangerous to the passengers.

To register the autonomous shuttle as a legal vehicle in Estonian, it is intended to require the tests, including disconnection of the system components that prove the shuttle control system is reliable. These requirements were considered when developing the updated safety concept shown in Figure 7. Each module has a safety-related function. For example, an AI computer processes lidar and camera data and can execute most smooth breaking if it is safe and the distance is sufficient. The master controller in this safety concept is mainly a gateway. It can disable all control packets sent out over all three CAN networks if some databus connection is lost. In this way, low-level vehicle hardware detects it and can execute emergency brakes that contain shut-down traction motor power, making braking with regular brakes and applying the handbrake. The AI computer also sees the disappearance of feedback packets and stops active driving action.

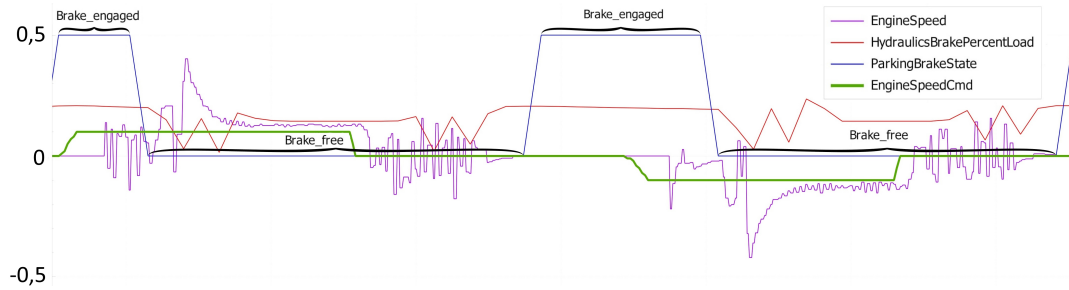


FIGURE 5. Engine speed request sent out by the original control system activates hydraulic brake and handbrake.

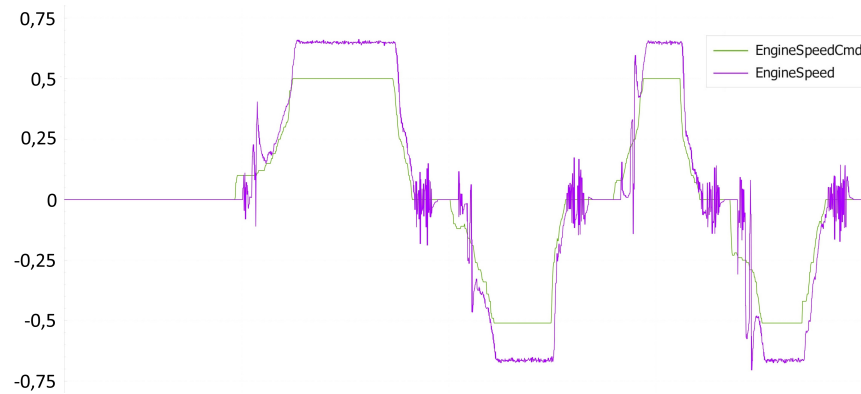


FIGURE 6. The engine speed requests sent by the custom master controller.

The new Master controller has three CAN connections and an Ethernet connection with the main computer. The test shows that if traction CAN1 is disconnected, the vehicle makes emergency braking and shuts down the high-voltage system. If steering CAN2 is disconnected, the steering stops, and the computer loses the position feedback package, then follows with a system shutdown. If body CAN3 is disconnected, all interior and exterior lights go out, and automatic doors stop operating. A special switch cuts off the power, and the doors can pull open manually. Finally, suppose the Ethernet connection between the Master controller and the new control computer is disconnected. All control packets disappear from all three CAN networks, and the shuttle applies emergency brakes and stops.

The operation of the emergency stop buttons was tested separately with the control signal generated by the new Master controller. The result was that the vehicle stopped immediately. Thus, the shuttle behaves under the safety concept described above. However, additional tests should be made to measure a braking force and check that it meets the requirements.

CONCLUSION

The existing self-driving shuttle transformation to an open-source solution involves multiple design steps like mapping the existing low-level architecture, logging and analyzing the CAN message, and identifying essential data. Then the new solution was implemented. Validate the result several experiments and tests with low-level and high-level components were performed to validate the safety. The most critical tests involved disconnecting various system components. Low-level tests included whether the life-critical actuators follow a given movement pattern accurately and if the selected action plan is triggered when the fault is artificially made. The new parallel-built shuttle TalTech iseAuto 2.0 can use experiment results and knowledge gained in this work to get it street-legal as soon as possible.

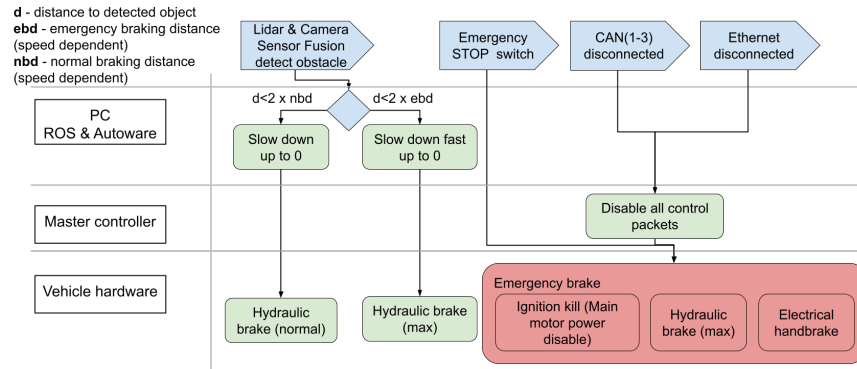


FIGURE 7. The shuttle planned safety concept.

ACKNOWLEDGMENTS

This research has received funding from two grants: the European Union's Horizon 2020 Research and Innovation Programme, under the grant agreement No. 856602, and the European Regional Development Fund, co-funded by the Estonian Ministry of Education and Research, under grant agreement No 2014-2020.4.01.20-0289. AI & Robotics Estonia – EDIH, project number 101083677. Master of Science in Smart, Secure and Interconnected Systems (MERIT) – Development of a new pan European educational ecosystem for training of digital specialists (project number VEU22048, DIGITAL-2021-SKILLS-01).

REFERENCES

1. J. Wetmore, "Driving the dream. the history and motivations behind 60 years of automated highway systems in america," *Automotive History Review* **7**, 4–19 (2003).
2. A. Broggi, *Automatic vehicle guidance: the experience of the ARGO autonomous vehicle* (World Scientific, 1999).
3. M. Buehler, K. Iagnemma, and S. Singh, *The DARPA urban challenge: autonomous vehicles in city traffic*, Vol. 56 (springer, 2009).
4. M. Bertozzi, L. Bombini, A. Broggi, M. Buzzoni, E. Cardarelli, S. Cattani, P. Cerri, A. Coati, S. Debattisti, A. Falzoni, *et al.*, "Viac: An out of ordinary experiment," in *2011 IEEE Intelligent Vehicles Symposium (IV)* (IEEE, 2011) pp. 175–180.
5. A. Broggi, M. Buzzoni, S. Debattisti, P. Grisleri, M. C. Laghi, P. Medici, and P. Versari, "Extensive tests of autonomous driving technologies," *IEEE Transactions on Intelligent Transportation Systems* **14**, 1403–1415 (2013).
6. J. Zhang and S. Singh, "Laser–visual–inertial odometry and mapping with high robustness and low drift," *Journal of field robotics* **35**, 1242–1264 (2018).
7. H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, and D. Li, "Object classification using cnn-based fusion of vision and lidar in autonomous vehicle environment," *IEEE Transactions on Industrial Informatics* **14**, 4224–4231 (2018).
8. A. Rassõlkin, T. Vaimann, A. Kallaste, and R. Sell, "Propulsion motor drive topology selection for further development of iseauto self-driving car," in *IEEE 59th Annual International Scientific Conference on Power and Electrical Engineering* (2018).
9. H. Pikner, R. Sell, J. Majak, and K. Karjust, "Safety system assessment case study of automated vehicle shuttle," *Electronics* **11** (2022), 10.3390/electronics11071162.
10. J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar, and B. Litkouhi, "Towards a viable autonomous driving research platform," in *2013 IEEE Intelligent Vehicles Symposium (IV)* (IEEE, 2013) pp. 763–770.
11. H. Pikner and K. Karjust, "Multi-layer cyber-physical low-level control solution for mobile robots," in *IOP Conference Series: Materials Science and Engineering*, Vol. 1140 (IOP Publishing, 2021) p. 012048.
12. M. Bellone, A. Ismailogullari, J. Mür, O. Nissin, R. Sell, and R.-M. Soe, "Autonomous driving in the real-world: The weather challenge in the sohjoa baltic project," in *Towards Connected and Autonomous Vehicle Highways* (Springer, 2021) pp. 229–255.
13. R. Sell, M. Leier, A. Rassõlkin, and J.-P. Ernits, "Self-driving car iseauto for research and education," in *2018 19th International Conference on Research and Education in Mechatronics (REM)* (IEEE, 2018) pp. 111–116.
14. H. Pikner, R. Sell, K. Karjust, E. Malayjerdi, and T. Velsker, "Cyber-physical control system for autonomous logistic robot," in *2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC)* (IEEE, 2021) pp. 699–704.
15. K. Rehrl and C. Zankl, "Digibus©: results from the first self-driving shuttle trial on a public road in austria," *European Transport Research Review* **10**, 1–11 (2018).
16. S. C. HPL, "Introduction to the controller area network (can)," Application Report SLOA101, 1–17 (2002).
17. H. Chi, J. Liu, W. Xu, M. Peng, and J. deGoicoechea, "Design hands-on lab exercises for cyber-physical systems security education," in *Journal of The Colloquium for Information Systems Security Education*, Vol. 9 (2022) pp. 8–8.