Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Highly Automated Electric Vehicle Platform for Control Education ⋆

Árpád Fehér * **Szilárd Aradi** * **Tamás Bécsi** * **Péter Gáspár** *

* *Department of Control for Transportation and Vehicle Systems,*
Budapest University of Technology and Economics (e-mail:
{feher.arpad, aradi.szilard, becsi.tamas, gaspar.peter}@mail.bme.hu)

Abstract: The paper presents a small-scale electric vehicle framework for vehicle control education and research. The main goal of the project is to serve as a good experimental platform for the students on any level of vehicle mechatronics education. It offers wide range of possibilities for embedded system, control design and machine learning applications. The proposed system is a redesigned version of our former experimental vehicle framework. The project was scheduled for two main goals. One was to design and implement a Robot Operating System based vehicle with Ackermann steering. The other was to develop a platform, which can be highly integrated into vehicle control education. In the paper the system architecture, the sensors and the control units are detailed, furthermore the educational benefits and some use cases are presented.

Keywords: Control Education, Autonomous Vehicles, Autonomous Robotic Systems, Control Architectures in Automotive Control, Electric and Solar vehicles

1. INTRODUCTION

Nowadays the automotive industry is going through deep paradigm changes. The electric and hybrid drivetrains are coming in fore. The Advanced Driver Assistance Systems (ADAS) are evolving dynamically using increasingly complex algorithms. In the near future more and more driving functions will be taken away from the driver converging to fully automated driving. Because of these challenges the application of Artificial Intelligence (AI) became necessary beside classic rule-based algorithms. All this resulted in, that the IT giants such as Google entered the automotive industry and the conventional vehicle manufacturers and suppliers have to improve their knowledge in the fields of data science and machine learning.

Today's vehicle engineers need to be capable to integrate mechanics, electronics, vehicle dynamics, control and machine intelligence to form such complex mechatronics systems, hence the educational institutions need to develop their curriculum to meet these needs. At the Budapest University of Technology and Economics Faculty of Transportation Engineering and Vehicle Engineering several BSc and MSc programmes and specializations are also focus on these aspects with special regard to our newest Autonomous Vehicle Control Engineer MSc. The development of an experimental electric vehicle platform presented in this paper is suitable to students acquire knowledge related to and beyond education.

The cart project has been started in mid 2013 at the Department of Control for Transportation and Vehicle Systems. The goal was to develop a controllable, Ackermann steering vehicle (See Fig.1.) for implementing and testing advanced driver assistance and autonomous driving functions. Hence, the purpose of the development was not the building of a real 1:1 scale vehicle, which would raise several questions and needless extra design problems. For the purpose of rapid framework development a much simpler and more affordable vehicle basis has been chosen: a simple commercial go-cart frame with hydraulic disc brake, rigid rear axle and a chain drive. The early version of the electrical system, the driving and the steering were designed in the first phase of the project. Sensors, actuatros and ECUs were installed, the cart can be controlled by CAN messages. The first construction is published in (Becsi et al. (2015)). Besides several positive experience, some issues were revealed as well:

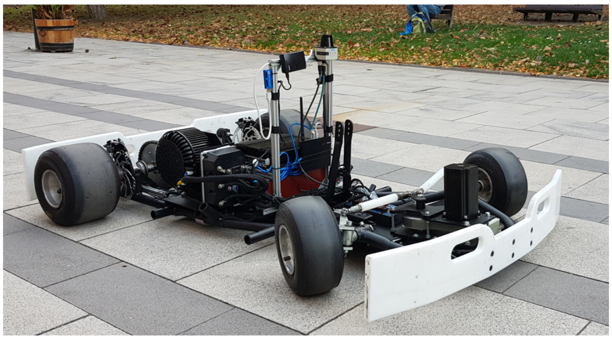
Fig. 1. The electric go-kart platform

- Weak steering mechanism
- Large turning circle
- Erratic wireless emergency stop

⋆ The research reported in this paper was supported by the Higher Education Excellence Program in the frame of Artificial Intelligence research area of Budapest University of Technology and Economics (BME FIKP-MI/FM).
EFOP-3.6.3-VEKOP-16-2017-00001: Talent management in autonomous vehicle control technologies- The Project is supported by the Hungarian Government and co-financed by the European Social Fund

- Lack of high computing central processing unit
- Lack of hydraulic brake system
- Inappropriate vision system

New low-level controls, a new steering gear, a differential gear, a mechanical braking system, a wireless remote control system and an environmental sensing camera were realized in order to solve these deficiencies.

Another motivation of redesigning our platform is to make it ROS-compatible, see Quigley et al. (2009). Over the past several years, Robot Operating System (ROS) has become widespread in the autonomous robot and vehicle research community. For more details see the two largest ROS-based autonomous vehicle projects: Baidu Inc. (2019) and Autoware Foundation (2019) ROS is a meta-operating system with lots of software libraries and tools to support the development of robot control applications. Several benefits of ROS can be listed:

- Distributed and modular
- Packages for robot applications such as mapping, localization, trajectory planning, perception etc.
- Python and C++ support. Python and C++ nodes can easily communicate
- Easy-to-use interface with Matlab and Simulink using the new ROS Toolbox
- Permissive open licensing

Finally if someone looks through the list of ROS-compatible robots (Open Source Robotics Foundation (2019)) , one will not find a medium-scale and affordable Ackermann-steered robot, only 1:1 scale or much smaller ones.

Several studies have already dealt with developing experimental vehicles for educational purposes. In Stebner et al. (2018) a three-wheeled lightweight chainless bicycle is presented. This vehicle is designed for education and research purposes. The drivetrain consists of a generator/motor combination. In Kiss and Trohák (2018) deals with design and implementation of the drivetrain of a hybrid cart, which is equipped with numerous sensors that provide the necessary information about its close environment. The drivetrain consists of an electric motor and an combustion engine. Most educational vehicles are small-sized robots. In Yu et al. (2017) the authors have developed an affordable, portable, and open source micro-scale mobile robot platform. As a complete and unique multi-vehicle platform enabled by 3D printing. In Giernacki et al. (2017) a quadrotor helicopter (quadcopter) as an open source experimental platform is presented. It is a low cost, easily expandable and upgradeable flying robot for research and education in robotics and control engineering. In Krauss (2016) the author presents low-cost vehicle platform based on the combination of a Raspberry Pi, an Arduino board, and a Zumo track-driven robot chassis. The article describes the construction of the vehicle without on-board sensors. Grepl et al. (2011) present a scaled four wheel steering and driven (4WS/4WD) vehicle. Basic kinematic and dynamic models are presented as the base for vehicle computer simulation as well as for the parameter estimation using variants of Kalman filter. Few educational purpose Ackermann steering vehicle platforms exists which, due to their size, allow placement of several industrial and autonomous sensors. Most articles focus on the drive chain and the construction design. For control educational purposes primarily small-scale differential drive robots and model cars can be found.

Section 2 details the system architecture and electronic systems of the new version experimental cart vehicle. In Section 3 our cart related educational programmes are presented. The Section 4 presents some realized tasks. Finally, in Section 5, the summary of the experiments and the future developments are concluded.

2. ARCHITECTURE AND ON-BOARD ELECTRONICS

As shown in Fig. 2, the architecture was designed on hierarchical way dividing the system into low-level and high-level control tasks. The system is consisted of several control units with different functions in order to separate the tasks in the education. The design of the control is based on the requirement that the architecture must be as modular as possible, so the cart could be easily reconfigured for different projects.

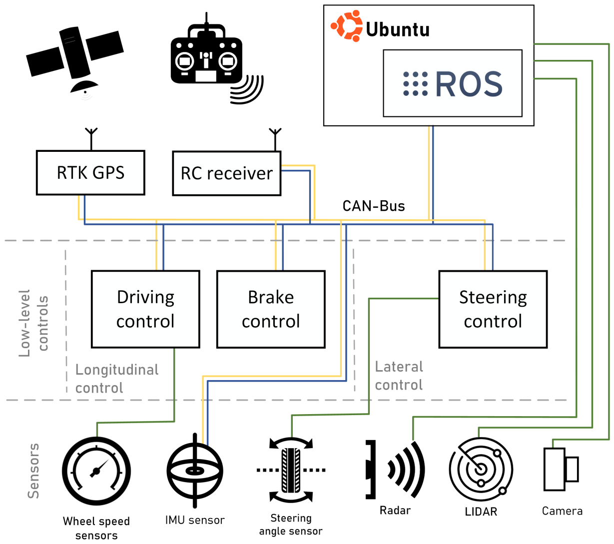

Fig. 2. Control architecture

The central element of the control architecture is an industrial PC, running Ubuntu Linux operating system. The high-level control algorithms are developed in Robot Operating System (ROS).

Fig. 2 shows that the experimental vehicle is also equipped with several sensors. The three main environment sensing units are connected to the main computer. Other sensors publish information to the common CAN network either directly or via a low-level control unit. The sensor set is detailed in table 1.

Among the listed environmental sensors, the camera is a self-developed module (Dániel et al. (2019)). Line and object detection is performed by a Raspberry Pi device using an Intel Neural Compute stick 2 module (See in Fig. 3).

For the localization-based functions, a self-developed differential RTK GPS module will be also included in the

Fig. 3. Environment sensing

future. The low cost device, which will be based on two u-blox ZED-F9P high precision GNSS modules, is currently under development. The preliminary tests showed that its accuracy is close to 1 cm at 20 Hz refresh rate. The location, heading and velocity information will be sent by the device via CAN bus.

According to past experiences, combining RC model based solutions with the Arduino system is very useful for developed experimental vehicles. Arduino is an open-source hardware and software ecosystem. The company offers a range of hardware platforms with with easy-to-use software libraries and documentation. A 6 channel Futaba T6L remote control and a Futaba R3106GF receiver were fitted to the system. The receiver signals is processed by Arduino Nano development board and sent them to the CAN network. The following tasks are performed:

- Manual control of the vehicle during the test.
- Triggering and parameter tuning of developed functions.
- External wireless emergency stop.

The low-level control tasks are performed by three electronic control units. These are responsible for controlling the three main actuators presented in the 2.1-2.3 subsections.

2.1 Electric Drive

The cart is driven by a 48V sensored BLDC motor. The rated power of the product of Golden Motor Company is 3KW-7.5KW. The HPB5000B motor has compact design, a water resistant body, a stainless steel shaft, and a self-cooling fan. A chain drive (H428 type with a gear ratio of 43/18) has been chosen to transfer the torque of the motor to the rear axle. A differential gear is mounted on the rear axle for easier maneuverability at low speeds. This drive train is able to accelerate at 2.5 m/s^2 up to 40 km/h with the maximum obtainable velocity of 60 km/h, which is more than satisfactory for our experimental needs. An HPC300H driver is designed to drive the BLDC motor. It is a high current programmable sinus wave driver with torque controller.

Table 1. Sensor set

Sensor	Type	Purpose	Position
Camera	Raspberry Pi module	Environment sensing	Top of the sensor frame
Lidar	Sick TIM571	Environment sensing	Top of the sensor frame
Radar	Continental ARS408-21	Environment sensing	Above the bumper
IMU	Bosch ESP sensor	Low-level control	Near COG
Steering angle sensor	ams AS5048A	Low-level control	Above the bumper
Wheel speed sensors	Automotive ABS sensor	Low-level control	Rear axle

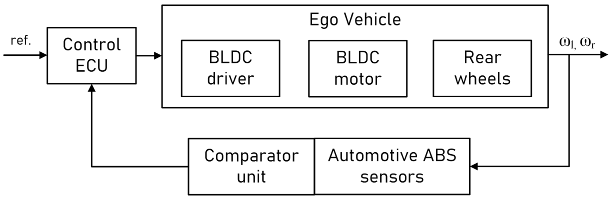

Fig. 4. Driving control

A self-developed ECU supervises the control of the vehicle drive. This well separated task does not require high computational power, so the microcontroller chosen for this control is a Microchip AT90CAN128, a low-power 8-bit AVR RISC-based unit with CAN controller. The speed of the rear wheels are measured to determine vehicle speed. Two sensor modules were installed on the rear axle because of the differential gear. These are classical automotive hall sensor based ABS encoders with a ferromagnetic ring in front of them. Their outputs are proportional to the current. In order to the microcontroller be able to receive signals, a resistance divider and a comparator unit are required to be integrated into the system.

2.2 Steering

A NEMA 34 size 86HS156-6204A stepping motor has been chosen for the steering actuator, having a torque of 12Nm with the rated current of 6.2A. The measured required torque is achieved with a T10 toothed belt drive having 2.75 transmission value. The motor is powered by an NDC 06 high current stepper driver.

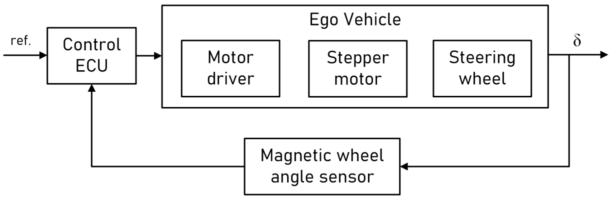

Fig. 5. Steering control

As shown in the Fig. 5 a separated control ECU is responsible for steering control. It is also a self-developed AT90CAN128 microprocessor based control unit. The module has been designed so to be mounted at the bottom of the steering column. The ECU contains the necessary basic circuit elements and an ams AS5048A magnetic angle sensor is placed to the middle of the printed circuit. The high precision 14-bit angular position sensor measures the absolute position of the magnets rotation angle, which is placed at the bottom of the steering column. The absolute position information of the magnet is directly accessible over a standard SPI interface.

2.3 Braking

The electric BLDC motor also has braking mode, which provides a relatively slight deceleration. A mechanical braking system was built into the cart in order to increase the possible deceleration range. It combines the Arduino environment with the modeling and cycling solutions. An SLX SM-RT66 203 mm diameter brake disc, an XT BR-M8020 four-piston caliper and an XT BL-M8000 brake lever are fitted on both sides. It is an Deore series hydraulic brake system from Shimano. The brake levers are actuated by two SRT DL 3017 digital model servos. The central unit of the low-level brake control is an Arduino Nano board with a CAN shield. The sensor signals of the control algorithm can be the Inertial Measurement Unit (IMU) and wheel speed information from the CAN network.

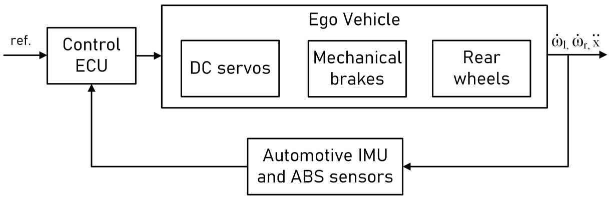

Fig. 6. Mechanical braking control

3. EDUCATIONAL BENEFITS

Our platform was designed especially for vehicle engineer education on different levels. First the rough curriculum of our educational programmes are presented, then the topics related to the control technology are detailed.

The Faculty of Transportation Engineering and Vehicle Engineering offers 3 BSc programmes.

- Transportation Engineering BSc
- Logistics Engineering BSc
- Vehicle Engineering BSc

In the Transportation Engineering BSc the main focus is on traffic modelling and control, furthermore the design and operation tasks of freight and passenger transport systems. Logistics Engineering BSc is basically similar to the previous programme, but it contains less about traffic modelling and it is extended with the corporate logistics and supply chains.

Finally, and the most important for the scope of the paper is the Vehicle Engineering BSc. It offers a Vehicle Mechatronics Specialization whose curriculum basically built on automotive embedded systems, on-board communications and vehicle control. Students learn about the development and testing of automotive microcontroller architectures with a special attention on the on-board bus systems such as CAN, LIN and FlexRay. Furthermore the students will be able to deal with vehicle dynamics and control design.

On MSc level the Vehicle Automatization specialization provides knowledge about systematic design and development of vehicle automation systems such as driver assistance and autonomous vehicle control systems and environmental perception systems. During the design phase safety and reliability issues are taken into account. Furthermore vehicle dynamics, system simulation, optimal planning, solving of complex vehicle automation tasks are assessed.

Our newest programme is the Autonomous Vehicle Control Engineer MSc. The aim is to provide the required knowledge to graduates in the fields of vehicle technology, control theory, intelligent systems, information technology and communications. The students will be prepared to take part in designing, developing and manufacturing highly automated vehicles, simulate on-board and V2X networks, testing and validation processes and work in a complex environment with various sensor data. The engineers graduated here will also be able to facilitate the creation of safe and energy-saving operation of modern transportation systems considering the requirements of environmental protection and sustainability.

The students use different simulation and development tools during their studies. In control theory education Matlab/Simulink is used, in embedded system lectures the students learn to program in C and C++, and they debug the communication networks with Vector interfaces and softwares.

As it can be seen, in our Vehicle Engineer programmes very extensive competencies can be obtained. One of our most important experience in vehicle engineer education that this complex knowledge can be used best, if the students learn to integrate them in a real-world control application

Our experimental vehicle is especially suitable to realize control tasks on different levels and using different software platforms from the basic PID control to the complex navigation, control methods even combining with machine learning applications.

Table 2. Educational usage

Lecture	Subsystem	Software
BSc Degree		
Vehicular On-board Systems	Low-Level ECUs	Arduino, C, C++
Sensors and Actuators	Low-Level ECUs, Wheel speed sensors, Steering angle sensor, IMU	Arduino, C, C++
Vehicle Control	Central PC	ROS (Python, C++) and/or Matlab
Vehicular Communication Systems	Low-Level ECUs, CAN	C, C++, Vector CANalyzer
MSc Degree		
Automotive Environment Sensing	Central PC	ROS (Python, C++) and/or Matlab, SICK SOPAS, Vector CANalyzer
Control Theory	Central PC	ROS (Python, C++) and/or Matlab
Discrete Control Design	Low-Level ECUs, Central PC	ROS (Python, C++) and/or Matlab
Mechatronics and Micorcomputers	Low-Level ECUs	C, C++

4. USE CASES

Several student research society papers, theses and scientific publications have been written about all tasks of developing the cart. These included the mechanical design of the actuators, PCB design of the custom ECUs, and the basic platform software. Tasks were well-separated and perfect for BSc and MSc thesis topics.

In the following chapters some typical use cases are introduced, which can be applied specifically in the control education.

A typical basic control design task is an automotive radar based adaptive cruise control (ACC) function. The low level control was run on an 8-bit controller, the ACC control (See Fig. 7) was implemented by a PC. It is a ideal BSc-level control experiment, in which the students can learn the design, implementation testing and tuning of a PID-controller.

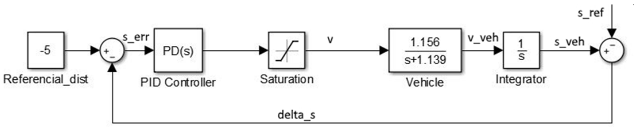

Fig. 7. Longitudinal control loop

Using the early state of the experimental vehicle, Törő et al. (2016) has published an automotive camera based lane keeping function. To achieve this task, two solutions were used, a double-loop control with feedforward load disturbance compensation (See Fig. 8) and a nonlinear method. The control algorithms were designed and tuned in a Hardware-In-The-Loop framework. The nonlinear algorithm was implemented on two different hardware devices and validated in CarSim–Matlab software environment. The camera attached to the go-cart was a first generation Multi-Purpose Camera (MPC1) from Robert Bosch GmbH. The device, equipped with CMOS sensor, is commonly used in passenger cars involved in road traffic.

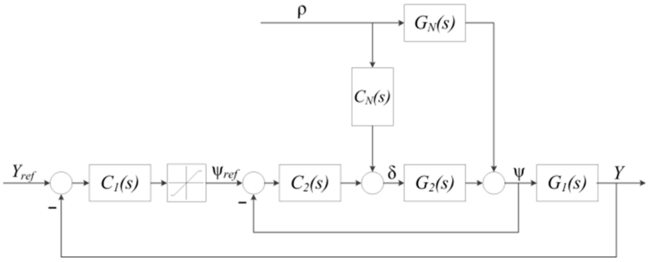

Fig. 8. Double loop control with feed forward compensator

MPC1 is especially designed for lane detection on highways, thus it cannot work reliably in a significantly different environment. Therefore a camera-based sensor has also been developed for the vehicle., which can be fitted to the vehicle. The goal was to provide high-level lane information (distance of lane-markings, lane geometry etc.). The hardware is a Raspberry PI 3B+ with a CMOS camera module. The lane detection algorithm was developed in both Python and C++ programming languages as MSc student scientific projects. Figure 9 shows the debug window of the lane detection software.

In later works, an odometry and ultrasonic sensors based home zone assist function was developed. The system solved a Simultaneous Localization And Mapping (SLAM) problem with routing. The function was enabled the vehicle to autonomously navigating on a predetermined route by continuously mapping the environment. For more details see Bécsi et al. (2017). These projects can be placed on MSc and PhD student level.


Departure Warning System with a Monocular Camera

Warning: offcenter > 0.6m (use higher threshold in real life)

Road Status
Lane info: Straight
Curvature 348838.0 m
Off center: Left 0.1m

Fig. 9. Lane detection software

Fig. 10. Developed SLAM function

Finally based on the improved vehicle, an automated emergency braking function has been implemented. According to the new concept the central processing unit was a Linux-based PC with ROS. The control logic sends low-level control signals to the longitudinal controller via the CAN network based on the object detection of the lidar sensor .

5. CONCLUSION AND FUTURE DEVELOPMENTS

In the above chapters our improved medium-scale electric vehicle platform was introduced. Its hardware and software architecture is based on Arduino units, C/C++ programmable custom embedded systems and Robot Operating System on an industrial PC. Unlike most ROS robots it is an Ackermann-steered vehicle. As it can be seen our platform can support the vehicle control education of both the undergraduate as well as postgraduate students including PhD studies. Furthermore it is capable for experiments with areas on the borderline to other fields such as localisation, mapping and perception.

In addition it provides space for different implementation techniques on different system levels.

- Fast prototyping using MATLAB/Simulink and ROS Toolbox
- ROS nodes in Python and C++
- Arduino platform
- C/C++ and Assembly on microcontrollers

In the near future it will be extended with an automotive 3D lidar sensor and later with an Nvidia GPU-based central computer. In the meantime the students work on several demonstration projects with different difficulty levels such as obstacle detection and avoidance, traffic jam assist and valet parking.

REFERENCES

Autoware Foundation (2019). Autoware.AI. URL `https://autoware.ai`.

Bécsi, T., Aradi, S., Fehér, ., and Gáldi, G. (2017). Autonomous Vehicle Function Experiments with Low-Cost Environment Sensors. *Transportation Research Procedia*, 27, 333–340. doi:10.1016/j.trpro.2017.12.143.

Baidu Inc. (2019). Apollo Auto Autonomous Driving Solution. URL `https://apollo.auto`.

Becsi, T., Aradi, S., and Gaspar, P. (2015). Educational Frameworks for Vehicle Mechatronics. *IEEE Transactions on Intelligent Transportation Systems*, 16(6), 3534–3542. doi:10.1109/TITS.2015.2480958.

Dániel, D., Árpád, F., and Szilárd, A. (2019). Sávdetektáló algoritmus teljesítményének összehasonlítása C++ és Python nyelven. In *IFFK 2019: XIII. Innováció és fenntartható felszíni közlekedés. (2019)*, Paper 18.

Giernacki, W., Skwierczyński, M., Witwicki, W., Wroński, P., and Kozierski, P. (2017). Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering. In *2017 22nd International Conference on Methods and Models in Automation and Robotics, MMAR 2017*, 37–42. Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/MMAR.2017.8046794.

Grepl, R., Vejlupek, J., Lambersky, V., Jasansky, M., Vadlejch, F., and Coupek, P. (2011). Development of 4WS/4WD experimental vehicle: Platform for research and education in mechatronics. In *2011 IEEE International Conference on Mechatronics, ICM 2011 - Proceedings*, 893–898. IEEE. doi: 10.1109/ICMECH.2011.5971241.

Kiss, D. and Trohák, A. (2018). Hybrid drivetrain of a self-driving Go-Kart. In *Lecture Notes in Mechanical Engineering*, 9783319756769, 81–86. Pleiades Publishing. doi:10.1007/978-3-319-75677-6_7.

Krauss, R. (2016). Combining Raspberry Pi and Arduino to form a low-cost, real-time autonomous vehicle platform. In *Proceedings of the American Control Conference*, volume 2016-July, 6628–6633. Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/ACC.2016.7526714.

Open Source Robotics Foundation (2019). ROS Robots. URL `https://robots.ros.org/`.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A.Y. (2009). ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, volume 3, 5. Kobe, Japan.

Stebner, G., Hartwig, C., and Kohlhauen, M. (2018). Lightweight Vehicle as multipurpose Platform for Education and Research. In *Proceedings of the 2018 19th International Conference on Research and Education in Mechatronics, REM 2018*, 105–110. Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/REM.2018.8421788.

Törő, O., Bécsi, T., and Aradi, S. (2016). Design of Lane Keeping Algorithm of Autonomous Vehicle. *Periodica Polytechnica Transportation Engineering*, 44(1), 60–68. doi:10.3311/PPtr.8177.

Yu, J., Han, S.D., Tang, W.N., and Rus, D. (2017). A portable, 3D-printing enabled multi-vehicle platform for robotics research and education. In *Proceedings - IEEE International Conference on Robotics and Automation*, 1475–1480. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/ICRA.2017.7989176.