Development and implementation of a software and hardware complex of the controls automation into the self-driving car system based on the "GAZelle NEXT" vehicle

M M Voronin, P A Anisimov, A V Zolotov
Alekseev Nizhniy Novgorod State Technical University, 603155 Minina st., 24, Nizhniy Novgorod, Russian Federation

e-mail: mindroz_prosperous@list.ru

Abstract. This article describes the stages and the results of automation of the car controls, as part of the self-driving car based on the light commercial vehicle (LCV) car class. Some aspects of automation of the car equipped with a manual transmission are touched upon. A block diagram of the automation system of the vehicle controls is provided. The variant of integration of modules of controls automation in the base car is shown. The description of the lower level program is given.

1. Introduction
On the base of NNSTU n.a. R. E. Alekseev and Private Institution of Further Vocational Education «Corporate University “GAZ Group” was formed a project team to create a self-driving vehicle based on LCV-class car “Gazelle NEXT”.

Figure 1. Self-driving “Gazelle NEXT”
The first step in creating an unmanned vehicle based on car “Gazelle NEXT” is the automation of its controls - the steering system, gearbox, and pedals, particularly.

One of the most important tasks in creating a self-driving car is to ensure road safety, and this can’t be done without the reliable and fast functioning of all vehicle components. The safety of the designing control automation system was emphasized in its design.

2. General structure description
The proposed structure of the control automation system has a hierarchical structure (see Fig. 2), and it is possible to distinguish the upper and lower levels in general. The lower level is represented by direct current motor and stepper motor controllers equipped with sensors. The upper level is represented by a computer with a movement planner program and a program that processes the movement planner commands, implementing the control algorithm of the lower level.

In the early stages of the design, it was decided to use a personal computer for upper level implementation. The components of the lower level were car controls' drivers - the controllers of the pedal unit, gearbox, and electric power steering. To provide a communication channel between the upper and lower levels, it was decided to use a CAN interface along with USB-CAN Converter, which was used as a CAN transceiver for the PC. CAN-packets generated at the upper level in response to the movement planner's instructions and the feedback are considered as control commands (see Fig. 1). Following that, CAN-packets like module status messages, telemetry packets and sensor readings are considered as feedback from the lower level and can be transferred only by request from the upper level.

After specifying the elements of the structure, the control automation system takes the following form (see Fig. 3):

![Figure 3. Controls automation system structure](image)

On-Board CAN-bus is used as a source of telemetry of the car: the speed of the machine (in km/h) and engine rate.
3. Description of the lower level automation units

3.1. Steering automation unit.
To automate the steering in the steering column of the base vehicle was introduced electric power steering «Avtoelectronica» (further EPS). EPS is equipped with a built-in control unit with CAN-interface, as well as direct current motor and absolute steering angle sensors. EPS can operate in two modes: in the manual mode, in which a steering wheel can be controlled directly by the driver, and the engine of EPS performs an auxiliary function, and remote control mode, in which steering is carried out by the control unit that receives commands from the CAN-network.

EPS safety in the remote mode is carried out in the following way: the command source is required to transmit the control synchronization package to the EPS control unit cyclically with a period of not more than 1 second, otherwise, the EPS will go to the manual mode and will inform about the loss of control. Also, the steering wheel goes into manual mode on command from the upper level in case of a critical situation. All this was taken into account when writing the algorithm of the main control program of the lower level.

3.2. Pedal block
One way to automate the pedal assembly is to control the pedals with an electric motor converting the rotational motion of the pulley into pedal motion. A stepper motor was used to control the pedals. To ensure the safety of the future system during emergencies, an Autonomous program was created for each of the automation units. For this purpose, the network controller based on the STM32F103 microcontroller was designed. The absolute encoder and stepper motor driver are connected to the controller board. The stepper motor is controlled by CAN commands. The main objective of the CAN-controlled motor driver controller is to ensure a clear and uninterrupted operation of the self-driving car in the presence of a control signal from the upper level and to perform an emergency stop of the vehicle in case of a system failure.

Figure 4 shows a 3D model of the network controller board, assembled with the case.

![Network controller board](image)

**Figure 4.** Network controller board

3.3. Gearbox automation unit
The car "GAZelle NEXT" is equipped with a manual transmission. A gearbox automation module is an automated unit that replaces the gearshift knob. The most reasonable and technologically correct solution in this direction is using an automatic or robotic gearbox. However, such a solution for the basic vehicle is not possible, because the car "GAZelle NEXT" is not equipped with such transmission. In this regard, as a module of automation of the gearbox, it was reasonable to develop a unit that simulates gear shifting using the gearbox handle.

Two linear actuators of the TGA-100 model equipped with DC motors were used as an actuating element. The actuators were controlled by CAN-interface using a network controller. This network controller is responsible for timely gear switching on the command from the CAN-bus. The current gear is determined by the limit switches connected to the controller.
4. Constructive implementation of control automation system modules

4.1. Electric power steering mounting design
To ensure the transmission of torque from the EPS to the steering shaft, the mechanical connection of the EPS output shaft with the standard steering shaft and the steering wheel of the basic vehicle was performed. As a tool for mounting the module into the steering column of the base vehicle, adapter brackets for mounting the module were designed and manufactured (shown in black in Fig. 5, a).

A three-dimensional model of the steering control automation module is presented in figure 5, a. Integrated into the steering column EPS is shown in figure 5, b.

![Figure 5. Steering automation unit.](image)

4.2 Gearbox automation unit design
To solve the problem of integration of the gearbox automation unit into the control system, the existing gearbox handle was dismantled, and the gear shift was performed directly with the gearbox drive cables. Brackets were designed to secure the cables (see Fig. 6). They were installed on the base support plate of the module - a molded plate of steel sheet with connecting holes. The model of the support plate is shown in figure 7.

Figure 8 shows the model of the assembled gearbox automation unit.

![Figure 6. Cable bracket 3D model](image) ![Figure 7. Support plate 3D model](image)
4.3 Pedal automation unit design

The development of pedal drivers was part of the task of the pedal automation unit integration into the self-driving car control system. The downshift was designed for the implementation of a connection between the stepper motor and the encoder shafts. A single-stage cylindrical gearbox with a gear ratio of 3 was used for this purpose. The model of the reducer with encoder is shown in figure 9. The model of the assembled drive unit is shown in figure 9, b.

The base plate is installed under the driver's seat is shown in figure 10. A metal cable was used as a kinematic link between the drive and the pedal, which was laid under the floor and the motor shield of the car.
5. Lower-level control program

The lower-level control program was intended for coordinating the operation of the resulting system modules. The control algorithm assumes the following:

1) Setting up the USB-CAN Converter at program startup;
2) Sending telemetry requests periodically;
3) Processing of received CAN-packets, i.e. telemetry decoding, processing of movement planner commands, error arbitration, etc.
4) Sending command packets to the CAN-bus in real-time;

The program was realized in Python. It is divided into the initialization module and an infinite loop represented by several parallel threads:

1) The main control module, which contains the algorithm of the pedals, gearbox and EPS functioning;
2) Telemetry request thread, which contains the telemetry request sequence;
3) USB-CAN converter dialogue thread, which organizes continuous receiving and sending of CAN-packets;
4) Movement planner commands receiving thread;

In the case of emergency, an emergency message with the highest priority in CAN network is sent, and if it’s necessary further stop of the vehicle can be made without the participation of the movement planner, but only with network controllers’ possibilities. If the connection between the lower and upper levels is broken, the vehicle stops without the necessary participation of the upper level. Also, the command for switching the EPS to manual mode in case of emergency is sent from the upper level.

6. Conclusion

As a result of this phase of the self-driving car project, the safety of the car automation system was ensured. The reliability of the self-driving car systems was tested in real conditions of operation in the car and showed the stable performance of the tasks. Testing was carried out not only in the conditions of normal system operation but also in case of emergency situations, which indicates the stability of the designed system.

This research done with the financial support from Ministry of Science and Higher Education of the Russian Federation (the unique project identifier is RFMEFI57717X0268).

References

[1] H Asama T Arai T Fukuda T Hasegawa 2002 *Distributed Autonomous Robotic Systems 5* Springer-Verlag Tokyo
[2] D S Vdovin and I O Khrenov 2019 IOP Conf. Ser.: Mater. Sci. Eng. 534 012016
[3] Pfeiffer O Ayre A Keydel C 2008 *Embedded Networking with CAN and CANopen* Copperhill Technologies Corporation
[4] Waschtl H Kolmanovsky I Willems F 2019 *Control Strategies for Advanced Driver Assistance Systems and Autonomous Driving Functions* Springer International Publishing
[5] Leandro C Fernandes, Jefferson R Souza, Gustavo Pessin, Patrick Y Shinzato, Daniel Sales, Caio Mendes, Marcos Prado, Rafael Klaser, André Chaves Magalhães, Alberto Hata, Daniel Pigatto, Kalinka Castelo Branco, Valdir Grassi Jr, Fernando S Osorio, Denis F Wolf (2014) *CarINA Intelligent Robotic Car: Architectural design and applications* Journal of Systems Architecture Volume 60, Issue 4, April 2014, Pages 372-392
[6] Ionita Silviu 2017 IOP Conf. Ser.: Mater. Sci. Eng. 252 012098
[7] Barabás I 2017 IOP Conf. Ser.: Mater. Sci. Eng. 252 012096
[8] Shahian-Jahromi Babak et al 2017 IOP Conf. Ser.: Mater. Sci. Eng. 224 012029