

# A HARDWARE-IN-THE-LOOP SIMULATOR BASED ON A REAL SKODA SUPERB VEHICLE AND RT-LAB/CARSIM

MILAN BIROS, KAROL KYSLAN\*, FRANTISEK DUROVSKY

*Technical University of Kosice, Department of Electrical Engineering and Mechatronics, Letna 9, Kosice, Slovak Republic*

\* corresponding author: [karol.kyslan@tuke.sk](mailto:karol.kyslan@tuke.sk)

**ABSTRACT.** This paper describes the design and realization of a hardware-in-the-loop simulator made from a real Skoda Superb vehicle. A combination of RT-LAB and CarSim software is used for real-time control and for handling the sensoric subsystems. The simulator provides almost realistic testing of driving cycles with on-line visualization. This unique device can be used in various fields of research.

**KEYWORDS:** hardware-in-the-loop, RT-LAB, CarSim, simulator, Skoda, Superb.

## 1. INTRODUCTION

In recent years, vehicle manufacturers and developers have focused on vehicle-human interaction, which has been leading towards eliminating the driver and replacing him/her by modern intelligent embedded systems. Manufacturers will have to establish new active safety systems due to mandatory European regulations and strict Euro NCAP criteria [1]. Developments in this field involve camera-based and radar-based assistance systems. These systems continuously monitor the surroundings of the vehicle, and provide considerable help for the driver during driving manoeuvres through visual or acoustic warnings or through autonomous reactions within a fraction of a second. Typical examples of such systems are Emergency Braking Assistant, Lane Keeping Assistant, Crossroads Assistant, and Automatic Cruise Control [2]. Developments of this kind cannot be made without advanced simulation systems. Hardware-in-the-Loop (HIL) simulation reduces development time and development costs, because it is not necessary to execute test rides with a real car in a real environment, and vehicle components can be tested separately. A controlled laboratory environment also improves accuracy and repeatability. Vehicle simulators have been used for this kind of research, often for designing and evaluating a wide range of control algorithms. These simulators can be classified as [3], [4] or [5].

According to their complexity:

- low-level: a car and a screen are fixed to the ground, a fixed base (FB) simulator
- mid-level: a car is accelerated (moved) in one degree of freedom (1DOF)
- high-level: a car is moved in at least six degrees of freedom (6DOF)

According to their use:

- training simulators: typically used to qualify for a driving license

- research simulators: a complex mechatronic unit is established for various research requirements

According to their cost:

- low-cost: using inexpensive graphic displays; the main applications are for screening the cognitive behavior of a driver
- mid-cost: employing real-time animation of a complete vehicle
- high-cost: used for evaluating electronic safety systems, or for accident reconstruction studies

According to the categories given above, this paper deals with a low-level, research, mid-cost driving simulator, based on a modified real vehicle. The aim is to connect a Skoda Superb vehicle to a real-time control system, and to use as many of the components of the vehicle as possible, resulting not only in a driving simulator, but also a complex mechatronic device that can be used for a range of research and educational requirements.

## 2. HARDWARE ARCHITECTURE

The essential component of the proposed simulator is the cockpit of a real Skoda Superb vehicle. Most of the electronic components and all important mechanical components remain unchanged inside the cockpit. Due to space limitations, the engine, the gearbox and the part of the vehicle behind the B-pillar were removed (see Fig. 2). This involved removing some electronic devices, such as the engine ECU, sensors for the engine and speed sensors on the wheels, which had to be replaced. In addition, some interior parts were removed or were slightly modified to improve visibility and access to the wiring, the harness connectors and other important components. Thanks to these modifications, users are able to observe the wiring loom and the placement of the components in a real vehicle in no time, without any tools. All adjustments were aimed at retaining the genuine arrangement of the electronics of the vehicle [6].

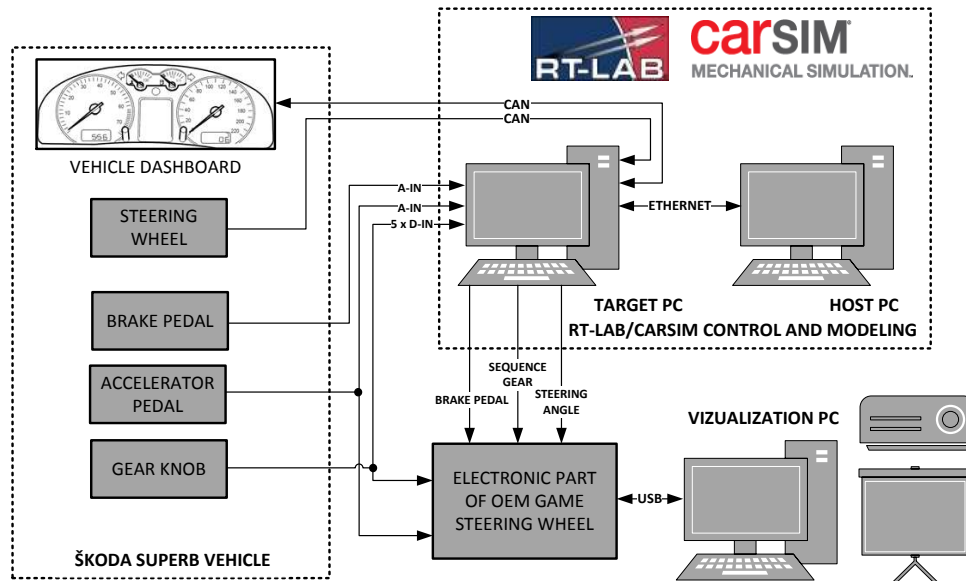


FIGURE 1. Hardware architecture of a vehicle simulator.

The *CarSimRT* real-time version by Mechanical Simulations is used as a simulation software for simulating the dynamics of the vehicle. A set of non-linear equations of the vehicle dynamics is calculated here, and enables the results to be seen in the form of time responses of particular mechanical variables. Likewise, the results can be animated by visualization in *Surfanim*. When on-line visualization is needed, the *LiveAnimator* tool provides real-time simulation, including the steering inputs by the car driver. It should be noted here that *LiveAnimator* was not available when the simulator was being constructed.

CarSim includes VehicleSim solvers, which calculate motion equations for mathematical models of vehicles. VS solvers use VS Lisp (originally named AutoSim) for symbolic solving of multi-body systems [7]. The current version of CarSim software includes a graphical user interface (GUI), which parameterizes the modelled vehicle in great detail. The user therefore does not need special knowledge of VS commands to create basic models. The current version also contains a large number of examples of vehicles and scenarios, which familiarize the user with the software functions and with the influence of the parameters on the manoeuvrability of the vehicle.

Only the basic parameters of the vehicle were obtained from the officially available documentation for the Skoda Superb 3U with a 1.8 l turbo charged spark-ignited engine (code designation AWT) and a 5-gear manual gearbox (code designation 01w-012). This is insufficient, due to the complexity of the CarSim software that is used. A default example from CarSim was therefore taken, and was supplemented by known parameters found in the documentation. The vehicle was constructed on a platform designated as B5 or



FIGURE 2. Skoda Superb vehicle simulator, rear view.

PL45, where P indicates a passenger car, L indicates longitudinal engine placement above the front axle, 4 is the size and 5 is the generation of the platform. This platform corresponds to CarSim dataset *D-Class, Sedan*.

In order to ensure precise timing with the vehicle components, calculations in CarSim software must be executed in real-time. The use of CarSimRT together with RT-LAB software creates the powerful HIL system, which meets the requirements for controlling the proposed vehicle simulator [8]. The RT-LAB system

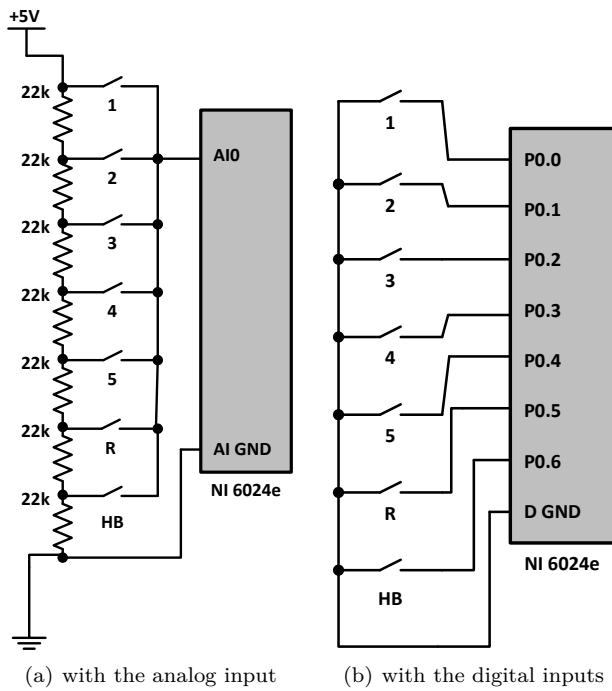


FIGURE 3. Measuring the position of the gear knob.

is an alternative to the widely-used dSpace [9], [10]. The control system consists of 2 personal computers (see Fig. 1).

*Host PC* acts as a console for controlling the simulation. It handles starting and stopping the execution, and in addition, the simulation results are displayed and stored here for further off-line evaluation. The mathematical model of the vehicle, developed or edited here in CarSimRT, is exported to MATLAB/Simulink as an S-function block, and is then compiled to C-language and sent to the *Target PC*.

*Target PC* with the QNX real-time operating system executes real-time computations and provides data acquisition (DAQ). This computer is equipped with a National Instruments 6024e DAQ card and a Softing CAN-AC2 PCI communication card. Signal scaling and processing is also performed here. Note that, with a few modifications, this HIL-based simulator can easily be extended for use in research on hybrid electric vehicles [11].

Without the CarSim *Live Animator* option, it is possible only to display the time responses of particular vehicle variables. The spatial orientation of the driver of the vehicle with monitoring of behavior of the vehicle only in the form of time responses would be unwieldy. In order to avoid the use of *Live Animator*, a commercial video game with a console and a steering wheel was introduced to convert the analog and digital signals from *Target PC* to another *Visualization PC*.

### 3. SENSORIC SUBSYSTEM

The original Superb dashboard was used to display some information necessary for the driver via CAN

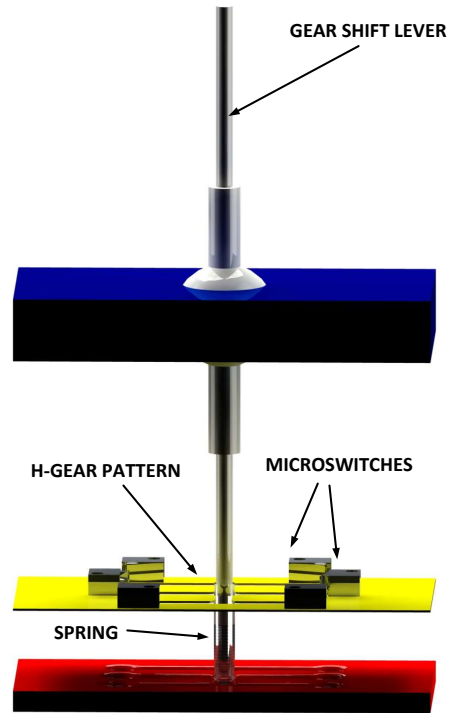


FIGURE 4. Structure of a position sensor for the gear.

bus, e.g. engine RPM (see Fig. 1). Many sensors have to be modified according to the notes in this section. These variables are measured in a real vehicle:

- gear knob position
- throttle position
- brake pedal position
- steering wheel angle

#### 3.1. POSITION OF THE GEAR KNOB

In most cases, vehicles with automatic transmission are equipped with a gear sensor. As was mentioned above, the Skoda Superb has a 5-gear manual gearbox, which is missing in the proposed simulator. The position sensor of the gear knob was constructed, but the main problem was where to put the knob. We mounted a small container with a spring at the bottom of a lever. The spring pushes the container against a plastic panel with slots and holes, which defines the range of movement of the lever. Holes assign the end position of the lever. We mounted another thin board above a plastic panel with a hole in the shape of the H-gear pattern. It was equipped with micro-switches in the end positions of the lever. The new structure of the gear shift lever is shown in Fig. 4.

At first, the micro-switches were connected as a resistor divider, as shown in Fig. 3(a). However, this was not ideal, because of the unidentified state in neutral gear. All the micro-switches were opened, so the residual voltage on the analog input of the DAQ card was gradually discharged. For example, when

1st gear was removed, the voltage slowly decreased from 5 V to 0 V. This means, that all the gears were detected in sequence, which is absolutely unacceptable. Another arrangement was therefore implemented, as in Fig. 3(b), where each micro-switch is connected directly to the one single digital input of the DAQ card. Thanks to this, the functionality of the sensor was re-established. However, the need for 5 digital inputs for the gear plus one for the hand brake takes up 6 of the 8 available digital inputs of the DAQ card, so there was limited capacity for adding new functions.

### 3.2. THROTTLE POSITION

The Skoda Superb is equipped with an electronic acceleration pedal. Its position is measured with a double-path potentiometer, which is also used in the simulator. It is not critical to operate this pedal during simulation, so it does not have to be redundant and only one path of the potentiometer is sensed. This sensor is supplied from an external voltage source, because there is no engine ECU. In a real vehicle, pedal failure detection is implemented. The voltage of the output sensor is in the range of 0.75–3.57 V, so it is offset for an analog input 0–5 V, and then is rescaled within the range of 0–1 for use in the CarSim S-function. In order to eliminate noise voltage that would cause engine RPM fluctuations, quantization with a 0.01 step is applied to the processed sensor output signal.

### 3.3. POSITION OF THE BRAKE PEDAL

In a real vehicle, information about the position of the brake pedal is obtained only by the circuit closer, so only an *on-off* signal is available. It is convenient to measure the position of the pedal in the simulator, which is later used in CarSim. For this purpose, the brake pedal system had to be equipped with a position sensor. Instead of an optical or linear position sensor, which would be expensive, we used a linear potentiometer modified for mounting to the pedal system. The path of the potentiometer did not match with the the range of movement of the pedal, so that the output voltage is processed in a similar way as in the accelerator pedal.

### 3.4. STEERING WHEEL

The original sensor from a real vehicle (G85) is used for measuring the steering wheel angle. It is an absolute quadratic encoder with a range of  $\pm 720^\circ$  directly connected to a CAN bus (CAN Antrieb). The value of the steering wheel angle in the CAN message is represented by 2 bytes. The lower 15 bits represent the angle, and the highest bit represents the direction of rotation. It is necessary to preserve this form of the signal for additional signal processing, e.g. scaling and direction detection in RT-LAB. In a real vehicle, this sensor is initialized after the key is switched to the first position and the steering wheel is turned about

$4.5^\circ$  by initializing a CAN message from the brake ECU. When the sensor is replaced or disconnected, it must be calibrated with the help of the VAG-COM diagnostic interface. The steering wheel angle is transmitted via CAN, then processed in RT-LAB and sent out to the game steering wheel electronics via analog output of the DAQ card.

## 4. VISUALIZATION

As was mentioned above, the *Live Animator* original real-time animation extension of CarSim was not available. An OEM gamer steering wheel and a PC game were used for real-time animation and visualization. The gamer steering wheel was completely disassembled, and the wheel itself is not used. Instead, its internal circuit board was disassembled and modified to connect to the DAQ card of the control system. The internal variables of the game can be accessed only by using the corresponding circuit board.

An advantage of this solution is that the steering wheel is connected to *Visualization PC* via USB, and is supplied with the existing USB driver, so it is not necessary to create a particular driver. However, the communication between *Target PC* and *Visualization PC* is only unidirectional, so the vehicle may behave slightly differently in the game and in the simulation. The main disadvantage of the OEM steering wheel is that its angle is only  $180^\circ$ . It is suitable for racing games, but is not appropriate as a driving simulator. Only way to remove this defect would be to use a different gaming device. For this reason, the steering angle signal from the vehicle's steering wheel  $\varphi$  has to be limited to  $\pm 90^\circ$  in software, and rescaled:

$$U_{in} = (\varphi + 90) \frac{5}{180}$$

Then the input voltage range  $U_{in}$  of the electronics of the gamer wheel is 0–5 V for the whole angle, and the steering wheel in the visualization is centered.

The original buttons on the gaming device were replaced by a position sensor made from gear lever micro-switches, because the DAQ card had an insufficient number of outputs. Classical gear shifting can therefore be used, if it is supported by the game. If not, two digital outputs of the DAQ card can be used and sequence gear can be applied, realized in MATLAB/Simulink (see Fig. 5). Then, however, there will be no other available digital output for sensing the position of the hand brake. If a sequence gear is chosen in the game, a model for the sequence gear, shown in Fig. 5, is active. An input 1 *Gear Model* is an output of a manual gear model, executed in CarSim. An input 2 *Seq Set* is a signal for manual synchronization of a gear in the game and in the vehicle. During execution, the game has information only about gear up or gear down, and no information about the current gear number. It has to be manually synchronized with input 2. In this part of the model, a final output

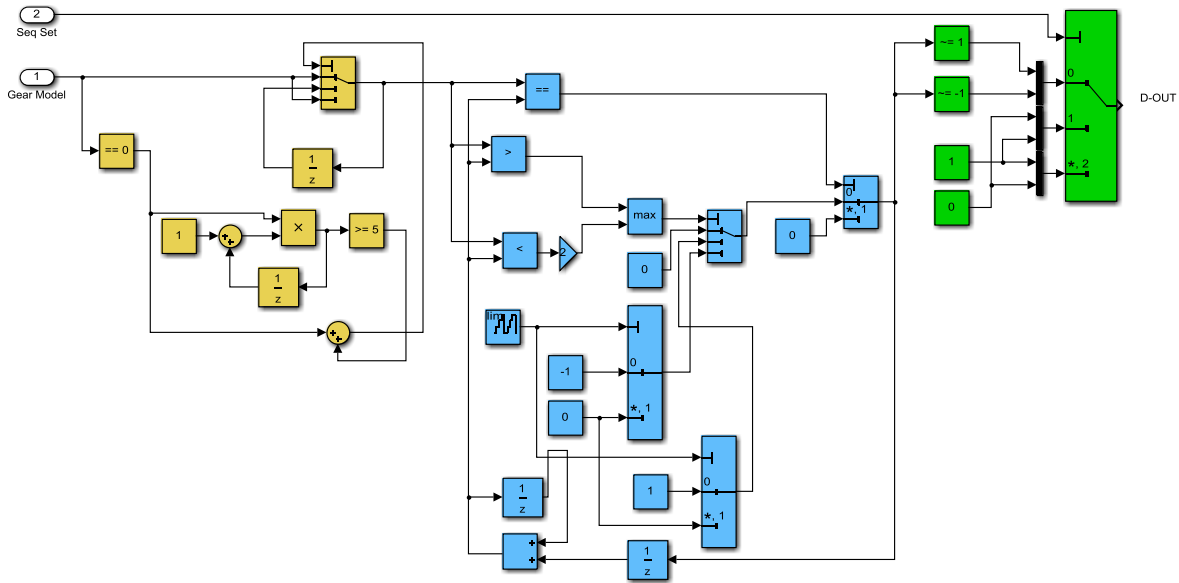


FIGURE 5. Sequence shifting algorithm.

vector transmitted to the digital outputs of the DAQ card is created.

The blue part of the model provides the main shifting algorithm. It has been verified experimentally, that gaming wheel buttons take at least 50 ms to detect pressing, and this causes delays. This value is implemented here as a sample time. Another delay is caused by adding detection of the neutral position during sequence gear. During shifting, e.g., from 1st gear to 2nd gear, the shifting in the game is sequenced as 1–0–1–2. Due to this, a waiting loop (yellow part of the model) was added to the algorithm. For neutral position of the lever, the algorithm waits 5 samples and then shifts to 0. This waiting loop causes an additional delay. However, this is not so critical, because if a gear change occurs within the waiting time, it is executed immediately.

### 5. SIMULATION RESULTS

The parameters of a vehicle model, running in CarSim, are slightly different from the parameters of a real vehicle. Only simulation results are presented in this section, and they are not compared with real vehicle measurements. The following simulation was performed with manual gear shifting, see in Fig. 6. The vehicle was driven by a user, and a real-time model of the vehicle was executed in the control system.

Gear shifting from 1st gear to 3rd gear can be observed in Fig. 6(a), when the vehicle accelerates smoothly to a speed of above 100 km/h. The speed of the vehicle at time  $t = 0$  s is not precisely zero, because unbraked vehicle tends to creep forward, due to the idle RPM engine vibration. The center of gravity speed of the vehicle reached 100 km/h at around  $t = 11$  s, which is about 1 s faster than the value specified for this vehicle by the manufacturer.

From  $t = 2$  s to  $t = 6$  s, there was heavy overspeeding on the front axle velocity. The front wheels skidded when low gears were shifted. A small slip can even be observed when there is a shift into 3rd gear. This is because the vehicle has the best power transmission from the engine to the pavement with a particular slip value included in vehicle model. When the gear lever traverses neutral position, discontinuous gear shifting behavior can be observed at the top of Fig. 6(a). This causes a temporary drop in velocity, and a negative acceleration peak can be observed at  $t = 4.3$  s.

The time responses during a double line change (DLC) manoeuvre are shown in Fig. 6(b). The acceleration of the vehicle is omitted, and only the DLC phase at an almost constant speed (60 km/h) is shown. At  $t = 12.1$  s the steering wheel is steered to an angle of approx.  $-90^\circ$ , and after one second it returns to zero. The vehicle has passed into another line. At  $t = 14.6$  s the wheel is steered to an angle of  $100^\circ$  and after a while, the vehicle is back in the original line.

The vertical forces in  $kN$  acting against the wheels are also shown in Fig. 6(b), where 1R, 1L are the forces acting on the front right wheel and on the front left wheel, respectively; and 2R, 2L are the forces acting on the rear right wheel and on the rear left wheel, respectively. At a constant speed, the forces acting on the front axle are greater than the forces acting on the rear axle, due to the placement of the engine in the front part of the vehicle. Centrifugal forces occur while the vehicle is steering to the left at  $t = 12.1$ – $13.1$  s. The forces acting on the wheels on the right side of the vehicle (1R, 2R) are therefore greater than the forces acting on the wheels on the left side (1L, 2L). Similar behavior can be observed while the vehicle is being steered to the right at  $t = 14.5$ – $15.5$  s.



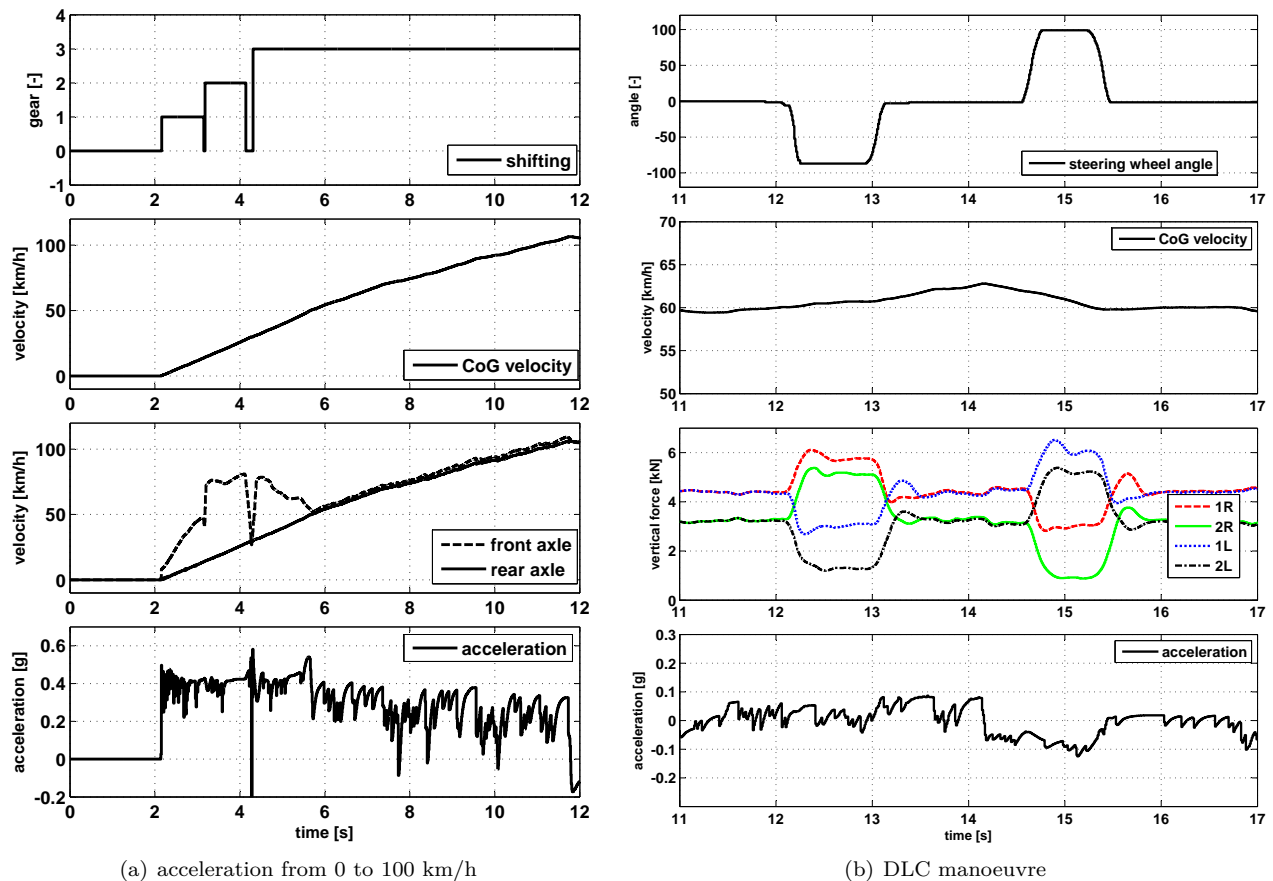


FIGURE 6. Simulation results.

## 6. CONCLUSION

The proposed vehicle simulator can be considered as a mid-cost, low-level research mechatronic device. It was built to verify its abilities when combined with a real vehicle. The aim was to use as many real vehicle components as possible. The main advantage of the simulator is that after the driving circuit geometry and the characteristics of the environment (pavement, weather condition etc.) have been designed, the user can drive a model of the simulated vehicle from a real car. There is no need to design a test procedure of the driver's behavior. The use of CarSim enables a huge variety of vehicle and environment parameters to be tested. However, the simulator has some drawbacks in its current state. The manual clutch from the real car is not available. In addition, the steering wheel is not equipped with feedback. Although the user can watch the behavior of the vehicle on a projector screen, this solution is not optimal. In future, the visualization will be upgraded to CarSim RT Animator or to Matlab Virtual Reality Toolbox, and the data acquisition channels will be extended to implement some new features.

## ACKNOWLEDGEMENTS

This work was supported by the Scientific Grant Agency of the Ministry of Education of the Slovak Republic and

of the Slovak Academy of Sciences (VEGA), under the project code 1/0464/15.

## REFERENCES

- [1] Euro NCAP. *NCAP*, 2014. <http://www.euroncap.com/technical.aspx>.
- [2] dSpace. *dSpace Magazine*, 2014. [https://www.dspace.com/en/pub/home/medien/product\\_info/eyes\\_on\\_the\\_road.cfm](https://www.dspace.com/en/pub/home/medien/product_info/eyes_on_the_road.cfm).
- [3] E. Blana. *A Survey of Driving Research Simulators Around the World*. Institute of Transport Studies, University of Leeds, Leeds, UK, 1996. <http://eprints.whiterose.ac.uk/2110/>.
- [4] J. Slob. *State-of-the-Art Driving Simulators, a Literature Survey*. DTC report. Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, Netherlands, 2008. <http://www.mate.tue.nl/mate/pdfs/9611.pdf>.
- [5] U. Eryilmaz, H. S. Tokmak, K. Cagiltay, et al. A novel classification method for driving simulators based on existing flight simulator classification standards. *Transportation Research Part C: Emerging Technologies* **42**(0):132–146, 2014. DOI:10.1016/j.trc.2014.02.011.
- [6] M. Biros. *SKODA Superb Vehicle Simulator in CARSIM*. Master Thesis, (in Slovak). Department of Electrical Engineering and Mechatronics, Technical University of Kosice, Kosice, Slovakia, 2014.

- [7] Mechanical Simulation. *Vehicle SIM Technology*, 2015. <http://www.carsim.com/products/supporting/vehiclesim/index.php>.
- [8] J. J. Kwon, T. W. Hong, K. Park, S. J. Heo. Robustness analysis of esc ecu to characteristic variation of vehicle chassis components using hils technique. *International Journal of Automotive Technology* **15**(3):429–439, 2014. DOI:10.1007/s12239-014-0045-3.
- [9] T. Haubert, T. Hlinovsky, P. Mindl. Modelling of electric vehicle dynamics using dspace dc 1103. *Transactions on Electrical Engineering* **3**(4):106–110, 2014. <http://www.transoneleng.org/2014/20144f.pdf>.
- [10] C. M. Kang, S.-H. Lee, C. C. Chung. Lane estimation using a vehicle kinematic lateral motion model under clothoidal road constraints. *IEEE 17th International Conference on Intelligent Transportation Systems (ITSC)* pp. 1066–1071, 2014. 8–11 Oct. 2014, DOI:10.1109/ITSC.2014.6957829.
- [11] Y. Inaba, S. Cense, T. O. Bachir, H. Yamashita. A dual high-speed pmsm motor drive emulator with finite element analysis on fpga chip with full fault testing capability. *Proceedings of the 14th European Conference on Power Electronics and Applications (EPE-2011)* 2011. Aug. 30–Sept. 1, 2011, Birmingham, United Kingdom.