

Research Article

Design Method of ADAS for Urban Electric Vehicle Based on Virtual Prototyping

Katarzyna Jezierska-Krupa  and **Wojciech Skarka** 

Institute of Fundamentals of Machinery Design, Faculty of Mechanical Engineering, Silesian University of Technology, Gliwice, Poland

Correspondence should be addressed to Wojciech Skarka; Wojciech.Skarka@polsl.pl

Received 25 June 2017; Revised 17 October 2017; Accepted 12 November 2017; Published 5 March 2018

Academic Editor: Rongxin Cui

Copyright © 2018 Katarzyna Jezierska-Krupa and Wojciech Skarka. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Since 2012, the Smart Power Team has been actively participating in the Shell Eco-marathon, which is a worldwide competition. From the very beginning, the team has been working to increase driver's safety on the road by developing Advanced Driver Assistance Systems. This paper presents unique method for designing ADAS systems in order to minimize the costs of the design phase and system implementation and, at the same time, to maximize the positive effect the system has on driver and vehicle safety. The described method is based on using virtual prototyping tool to simulate the system performance in real-life situations. This approach enabled an iterative design process, which resulted in reduction of errors with almost no prototyping and testing costs.

1. Introduction

Conducting the design process of Advanced Driver Assistance Systems (ADAS) in an optimum way requires a specific approach to defining and solving problem [1]. Profound research and analysis have to be carried in order to prepare well for this task. Sometimes it is impossible though to identify and avoid certain mistakes in the process, without using a prototype [2–4]. Currently, there are more and more tools available for engineers that enable creating virtual prototypes and therefore optimizing the design process in terms of its effectiveness and fault minimization [5]. Nevertheless, in order to use these tools efficiently, there needs to be a proper approach to design process adopted. The aim of this article is to present the approach to ADAS design process, which includes the use of simulation methods and virtual prototyping. Preceding the actual method description, there are some considerations included, whether there are reasonable grounds for using virtual prototyping tool and what other tools should be considered before making the final decision. The subsequent sections of this article include the method description and the use case, that is, design process of Blind Spot Information System for urban vehicle. This article is an extended version of the article [6] presented at the International Conference on Transdisciplinary Engineering 2016 in Curitiba, Brazil.

2. Consideration of the Rationale for Using Virtual Prototyping Tool

2.1. Defining the Needs. The first step that needs to be taken into system of design process needs definition [4, 7, 8]. What must be identified are the problems that a driver is facing and possibilities of finding the right solutions for them. The most effective ways of finding them are interviews with actual drivers and analysis of traffic situations of great risk for driver, vehicle, or third parts safety. The most common conditions in which the driving task is performed [1, 5] need to be analysed in this step. For vehicles dedicated for specific goals, such as competitions, these conditions can be described very precisely, which enables finding most current problems and most effective solutions. The choice of prototyping tool, which would be used in further steps, needs to be based on tool possibilities in terms of reflecting real environmental and situational conditions.

2.2. Defining the External Constraints. Before deciding on using virtual prototyping tool for ADAS design, it needs to be considered whether there is such a need at all. To answer this question, the external constraints must be considered and it must be verified how the design process will be affected by using simulation methods.

There are multiple constraints that must be considered, when thinking about designing ADAS. One of the most basic ones are costs, time, and quality. According to a well-known business rule of thumb, it is accepted that only two of these three constraints can be fulfilled at the same time.

2.3. Discussion of Possible ADAS Design Methods. There are various methods of ADAS design available and it is crucial to choose the one that fits the needs of the project best. ADAS can be designed in the following ways:

- (1) No prototype: which does not enable us to verify its performance in real-life situations until the system is implemented
- (2) Virtual prototype: which, accordingly to a chosen tool, enables us to simulate the natural conditions in really accurate but not perfectly accurate way, generates low costs, and enables the iterative design process [3, 5]
- (3) Real prototype: which enables absolute verification but has also many drawbacks such as costs, production, and construction time or difficult interaction [3]

When considering the virtual prototyping tool, the above-mentioned aspects must be included.

3. Design Method

3.1. Project Analysis. In order to minimize the need for changes in further steps of the ADAS design process, it is recommended to define all internal constraints before working on an actual model. This approach enables us to precisely set the goal of the designed ADA system and define the required constraints early enough to be included in the model. Using a virtual prototyping tool for such complicated systems as ADAS requires splitting considerations about the whole system into subsystems.

3.2. Designing and Testing. Testing part of designed system is a step that must not be neglected. Thorough tests are the last step before real system implementation that allow error detection without generating additional costs in manufacturing process. Correctly conducted testing enables also very accurate error identification and makes its correction easier.

In order to properly define test cases for ADAS virtual prototype testing, three separate system aspects must be considered, in which an error could appear:

- (1) Data processing: software
 - (1.1) What are the system inputs?
 - (1.2) What are the system outputs?
 - (1.3) What is the path of converting inputs to outputs?
 - (1.4) What values are independent from the system design and may not be tested?
- (2) Functional requirements: UI and usability
 - (2.1) What is the system supposed to do?



FIGURE 1: Bytel vehicle during Shell Eco-marathon race in Rotterdam (2015).

- (2.2) What information is it supposed to communicate to the driver?
- (2.3) What must the system not do?

(3) Models design: the idea

- (3.1) How should the system work?
- (3.2) What inputs and outputs should it accept/generate?
- (3.3) What functions are the system described by?

Such analysis enables us to create test cases sheet that should be used in test experiments outcome evaluation. Only when every test case will end successfully, that is, the values will be corresponding with template values and the visual verification will be fulfilled, the system can be implemented.

For correct testing process of ADAS virtual prototype, it is also needed to cover all possible situations in simulations, in which an error could appear. The test experiments should include both typical traffic situations of a higher risk and nonrisky situations in order to test the system against false alarms.

4. Use Case: BLIS for Urban Electric Vehicle

The above-described design method was used in design process of Blind Spot Information System (BLIS) [2, 6, 9, 10] for urban electric vehicle, Bytel [6, 11, 12] (Figure 1). Bytel is a vehicle created to participate in Shell Eco-marathon (SEM) [13] competitions, the event which aims to encourage design of highly efficient vehicles by students and scientific organizations. Bytel has participated in SEM in 2014 and 2015 in two power source categories: battery electric and hydrogen (hydrogen fuel cell stack).

The steps described above were taken to successfully finish the process at minimal time and maximum quality, keeping the costs reasonable at the same time.

4.1. Historical Determinants of BLIS System Development. The concept of this BLIS system is based on the team's experience in this field, taking into account specific conditions and upon existing solutions. The basic assumption of the system is to inform the driver about the appearance of another vehicle in

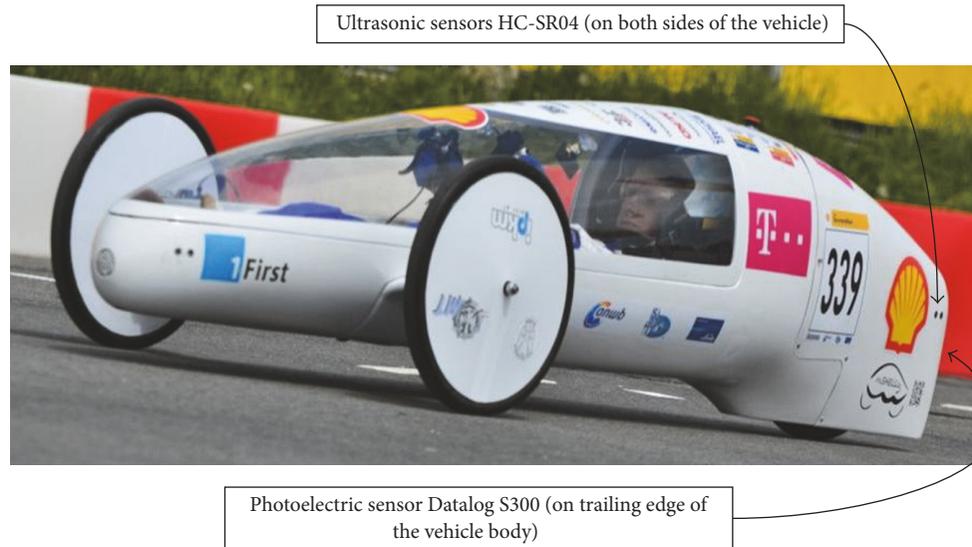


FIGURE 2: Vehicle MuSHELLka at Shell Eco-marathon 2013 in Rotterdam. Placement of BLIS system sensors in MuSHELLka.

the blind spot. The final form of the system was influenced by experience with previous ADAS systems, including the BLIS system, which was fitted with another vehicle [11] (MuSHELLka) developed by the team (Figure 2).

Previous technical solution of BLIS system [6, 11] is composed of a controller which is based on Microcontroller ATmega8 and 3 sensors. The main task of the controller is to handle sensors and to transmit information about the detected threat to ACS controller using RS232. Two ultrasonic sensors HCSR-04 and diffusive photoelectric sensor Datalogic S300 are used for scanning area behind the vehicle. The HC-SR04 ultrasonic sensor uses sonar to determine the distance from an object in the range of 2 cm–400 cm, with 3 mm resolution. Sound wave that is used for measurement has a frequency of 40 kHz [11]. Datalogic S300 is an advanced photoelectric sensor with a detection range of 0–500 cm for white objects and 0–350 cm for black objects. Infrared radiation and triangulation are used for measurement. The sensor response is binary, and the detection range can be adjusted in the full range of measurement. Due to the use of the sensor, a critical parameter is the response time. Maximum response time for this sensor is shorter than 2 ms [11]. All system components are placed in the rear fairing, which was possible thanks to the compact design and minimum weight.

This solution proved to be a much smaller vehicle because of its simple construction, low cost, and high operational reliability. It is also suitable for small size vehicles and small mass. It is important to use the BLIS virtual prototyping method in the development of this system. As part of the development of the previous system, a virtual race track system was used, a vehicle model in the TASS PreScan [14] environment, and a numerical simulation model of the BLIS system in this environment using the MATLAB/Simulink environment. As part of the development of the previous system, a full simulation test was conducted in the simulation environment and subsequent verification was performed on

the actual site. These tests confirmed the compatibility of the system parameters with the assumptions and the experience of the design process, virtual testing, and racing operation are the basis for the development of the new BLIS system.

As the next designed vehicle [15, 16] (Bytel) is only equipped with side mirrors and does not have a rear window or a central rear-view mirror, it is not only the monitoring of the area on the side of the vehicle, but also the area immediately behind the vehicle. A number of conceptual solutions have been analysed, of which three are presented below (Figure 3).

The first concept (Figure 3(a)) was based on the use of photoelectric sensors. The experience of using this type of sensor in previous solutions indicated the potential for such a system, but ultimately, due to the need to use a large number of such sensors, the vehicle aerodynamic effects were adversely affected, and its relatively low efficiency and range, especially in variable weather conditions, and insufficient information obtained from such a system has been rejected.

The second concept (Figure 3(b)) was based on the use of a vision system based on the Kinect. The design assumes the use of the depth camera included in the kit to determine the distance of the approaching vehicles. If larger areas than a single sensor have to be scanned, more Kinect devices can be used. This, however, complicates the connection and requires data analysis from two or more different sensors. Creating a system, based on two or a few of this type of sensors, which would recognize an object with satisfying accuracy and calculates position, is a difficult task and thus it requires great processing power. This affects not only the recognition of the image, but also the angle at which the vehicles will be detected. Taking into account the parameters of a commonly available Kinect device, a sensor placement scheme was used when using four devices. The advantages of this solution are the innovation and the use of ready-made modules and the large potential of the system. The disadvantages include the large dimensions of the device, relatively small scanning angle

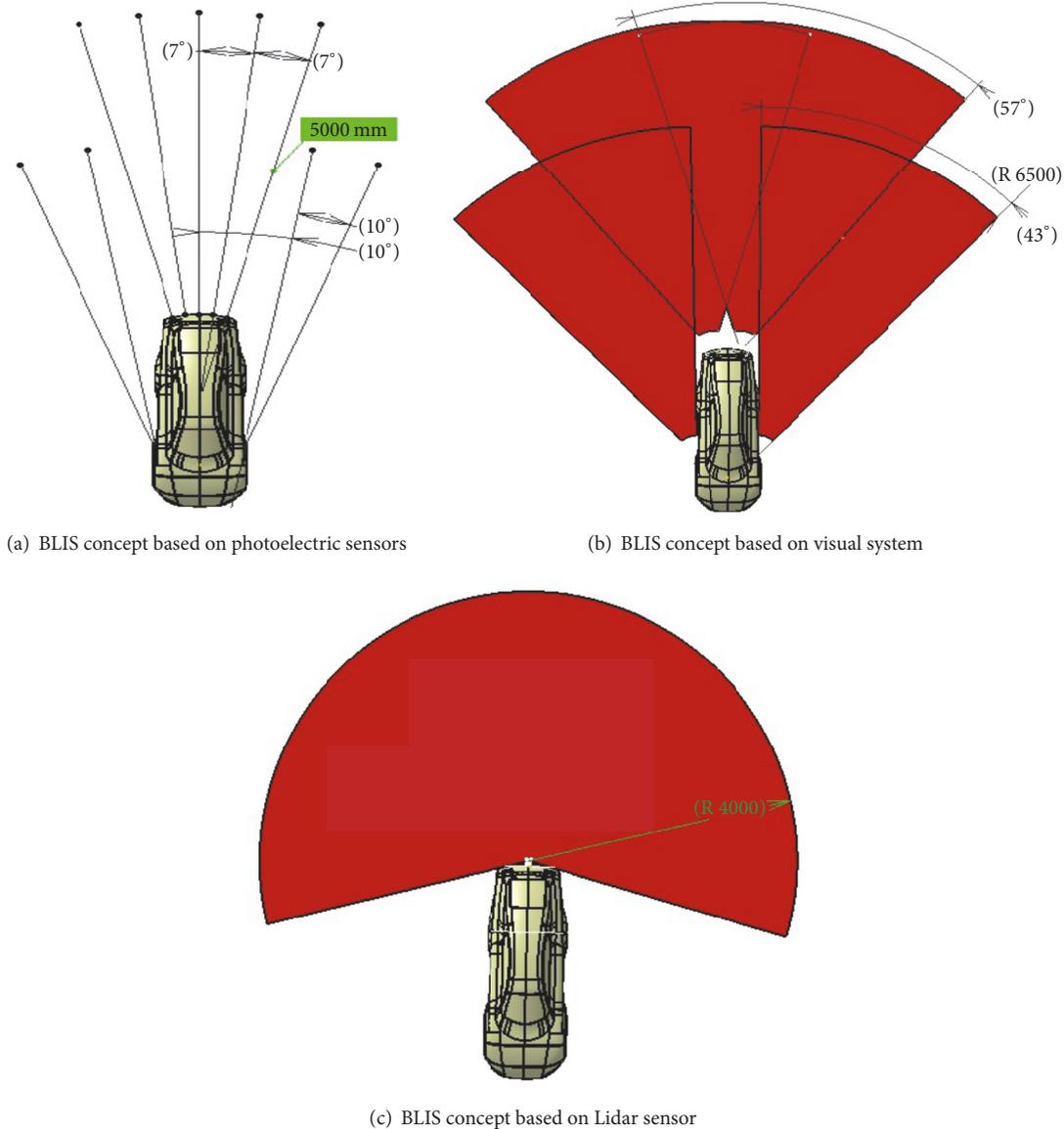


FIGURE 3: Alternative concepts of BLIS system.

in case of a single sensor, the possibility of disturbance of measurements in real conditions (sun rays), and the need to use a PC.

The third concept (Figure 3(c)) assumes the use of a Hokuyo laser sensor. This sensor is characterized by high resolution (1080 samples per 270° , corresponding to 4 samples per 1°), high sample rate (40 Hz), and wide scan area (270°) [17]. The laser used in this sensor fulfils the 1st safety class, which allows it to be used during SEM competition and is characterized by low sensitivity to atmospheric agents (sun, rain, snow, and fog). The advantages of this solution are simple assembly, single device, and high reliability, whereas the main disadvantage is the high cost of the sensor and the possibility of problems arising from the analysis of so much data. This, however, provides a field for the use of complex algorithms to analyse the data to eliminate errors and increase operational certainty, which was also used and presented later

in this paper. Ultimately, the laser sensor solution has been used in further development work.

The reflections and experience gained from previous development work led to the new BLIS solution to modify both the system itself and the design methodology. The new design method was shortened by abandoning the cost-intensive and time-consuming verification of the real object. Each of the newly constructed vehicles is treated as a prototype vehicle and a shortened test of the actual object is justified. These tests are designed to be carried out on the track and during races. This does not mean that the system will not be verified. The emphasis was placed on virtual prototyping, a formal verification phase was established during these studies, and new methods were introduced to verify the results of the simulation tests by introducing new independent computational methods to duplicate the results of standard simulations.

As far as the concept was concerned, it was decided to change the concept of the system completely by introducing a new interface in the driver's field of vision, providing much more information but not distracting the driver. The use of a single Lidar sensor with a range of measurable resistance to ambient conditions and at the same time to meet the rigorous requirements of the vehicle aerodynamic performance is critical to ensure high reliability of the proposed system of the development method using virtual prototyping methods and it provides the possibility of using this sensor and integration of the BLIS system into other ADAS systems [18, 19].

4.2. Consideration of the Rationale for Using Virtual Prototyping Tool. The decision to develop BLIS for urban vehicle was made based on low viewing range, which is characteristic for common conditions for these kinds of vehicles [18], that is, vehicles designed specifically for Shell Eco-marathon competitions. BLIS is the system that informs the driver about vehicles coming from the back of the vehicle, which are invisible to the driver due to its low viewing range. The need of creating such a system has been recognized based on analysis of many accidents and dangerous situations happening during the race and also after discussing the problem with drivers participating in SEM.

While deciding to use virtual prototyping tool, the crucial reason for using it was the necessity for parallel production process. As the vehicle was still in development phase during the design of BLIS, there was no possibility of using real prototype in order to conduct tests and work on data processing improvement. This is equivalent to accepting time as the main external constraint that applies to this project.

The second constraint was quality of the system, as the vehicle purpose was to take part in SEM competition; that is, in the race, it was considered extremely important to provide the driver with ADAS of the best possible quality. The point was to help the driver keep safe and avoid active or passive participation in accidents.

Last but not least, the adequate tool needed to be chosen for designing and conducting BLIS simulations. There were a few virtual prototyping tools especially dedicated to ADAS design which were considered. The final choice was TASS PreScan [14] due to its wide range of predefined sensors, very friendly user interface, and great visualization capabilities.

4.3. Project Analysis

4.3.1. The Scope of Development Works Performed Using Virtual Prototyping Methods. As part of the development work on the new BLIS system for the Bytel vehicle, the following scope of work is planned:

- (i) Analytical analysis of previous versions of BLIS and competing solutions
- (ii) Development of detailed functional requirements and requirements for integration with vehicle and other vehicle systems, as well as internal and external constraints
- (iii) Development of a detailed BLIS concept

- (iv) Construction of a simulation environment involving simulation models of an example race track, designed vehicle, and other vehicles
- (v) Construction of the mathematical simulation model of the BLIS system
- (vi) Identification and development of simulation scenarios
- (vii) Development of alternative algorithms for methods of verification of virtual prototyping simulations
- (viii) Carrying out a series of simulations with verification tests and improving the parameters and structure of the developed BLIS solution in iterative mode until the results are achieved
- (ix) Performance analysis and production and integration plan

According to the designers of the vehicle, a set of detailed planning simulation tests and independent allowing algorithms to verify the results of simulation studies are particularly important for the new method. Correct planning and carrying out of this part of the research guarantee the possibility of doing without the physical prototype. Therefore, further attention was paid to the detailed description of these fragments of the research.

4.3.2. Defining the Internal Constraints. Following the described method, the first step after recognizing the need of creating BLIS and defining the external constraints that lead to decision of using virtual prototyping tool was defining the internal constraints of the designed system.

- (1) The system cannot affect actively the breaking and/or steering system.
- (2) The system functions must be restricted to information/warning functions.

The self-imposed constraints of the project were also directly connected to Shell Eco-marathon assumptions, as the purpose of Bytel vehicle was to achieve the best result possible in this competition (SEM is about developing highly energetic vehicles). In order to satisfy this goal, there were the following constraint sets:

- (1) The weight of the system must be minimized.
- (2) The system should not consume relevant amount of power.

These constraints enabled us to define the general approach to designed system.

At this point, the decision regarding system parts has been made, which was vital to create the system model in TASS PreScan. There were three possible kinds of sensors considered: photoelectric sensors, Lidar, and depth camera. These kinds of sensors are available in predefined form in TASS PreScan, with a possibility of their configuration in very wide range. The decision was made to use Hokuyo Lidar (Figure 3(c)).

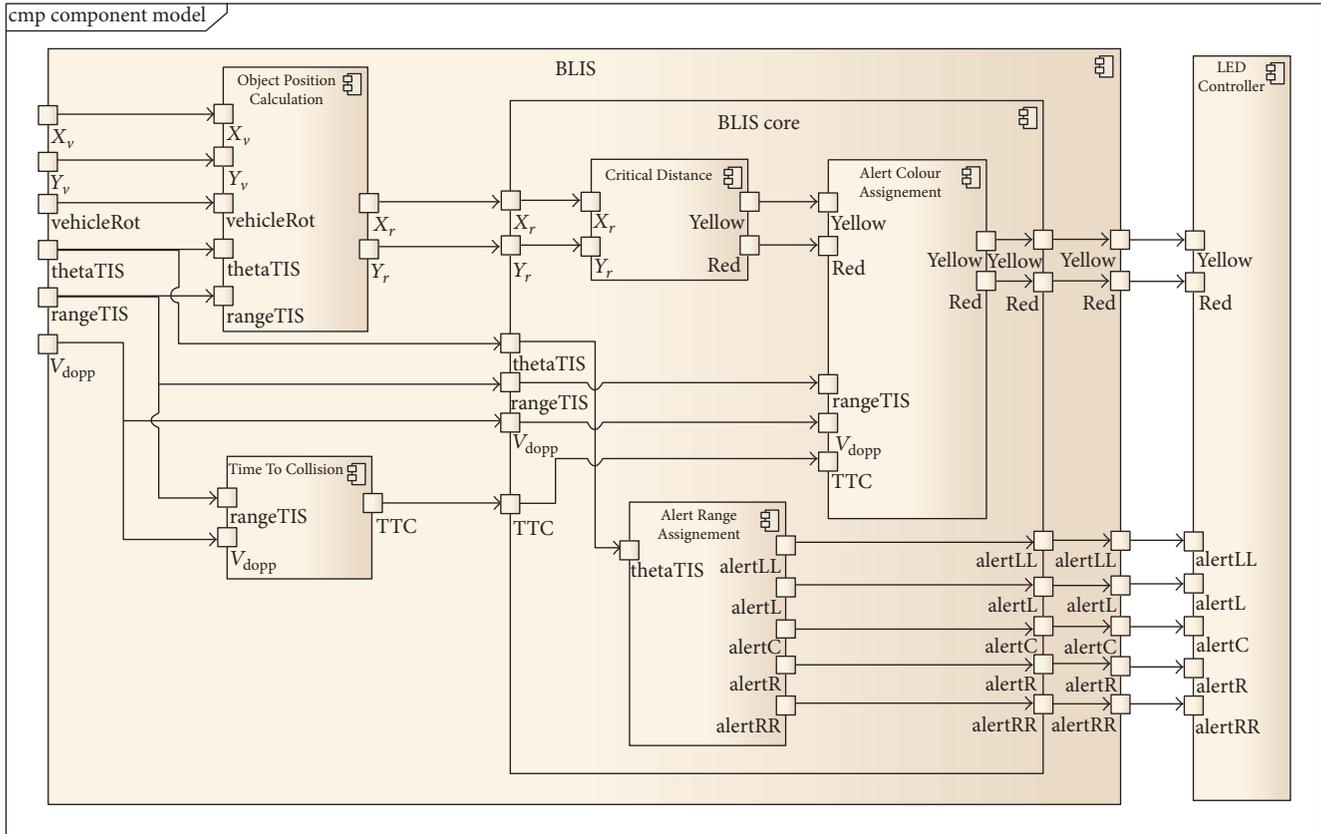


FIGURE 4: Flowchart of BLIS for Bytel vehicle.

What was crucial when deciding about rejecting the photoelectric sensors option was their low resistance to environmental conditions; a high risk of inaccurate results was identified. Also, in case of photoelectric sensors, there would be a need for using at least 9 sensors in order to cover the area invisible to the driver sufficiently, whereas there was only one Lidar needed to enable scanning of this range.

However, the difficulties in configuring and designing a system based on depth camera and the size of the system that interfere with the aerodynamic requirements have resulted in rejection of this solution.

4.3.3. Splitting System into Subsystems. Defining the subsystems and splitting the system into them when developing a virtual prototype not only made it easier to follow the design process step by step but also created good ground for tests analysis.

The main determinants of dividing line which would be used for splitting the system were assumed real-life subsystems. This means that it was needed to consider the separate parts of BLIS system, as they were meant to be implemented in real life. This is why the decision was made to consider BLIS in two subsystems: data processing subsystem and control subsystem.

The data processing system included also a verification subsystem, which uses the data generated in PreScan in order to compare the values between ideal ones and the ones that were received via data processing based on assumed

input values. An example of such comparison can be seen in Figure 9.

4.4. Operational Model of the System. Operational model of the system can be seen in Figure 4. The base of the system is the use of one laser sensor with a large angular range and coverage.

Input variables of the system are as follows:

- (i) X_v : X coordinate of the Bytel vehicle in the absolute coordinate system
- (ii) Y_v : Y coordinate of the Bytel vehicle in the absolute coordinate system
- (iii) vehicleRot: rotation angle of the vehicle in the z-axis
- (iv) thetaTIS: the angle at which the vehicle was detected
- (v) rangeTIS: the distance of the detected vehicle to the sensor
- (vi) V_{dopp} : vehicles relative velocity

Outputs of the system are as follows:

- (i) yellowAlert: low (yellow) risk information
- (ii) redAlert: high (red) risk information
- (iii) alertLL: information about the appearance of an object in the extreme left area

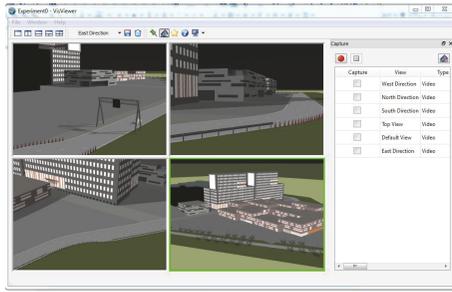


FIGURE 5: Model of the environment made in TASS PreScan.

- (iv) alertL: information about the appearance of an object in the left area
- (v) alertC: information about the appearance of an object in the center area
- (vi) alertR: information about the appearance of an object in the right area
- (vii) alertRR: information about the appearance of an object in the extreme right

4.5. Building Scenario: Building Environment

4.5.1. Simulation Environment Modelling. In order to carry out the relevant simulations to verify the correctness of operation of the designed system, it was necessary to design a simulation environment, vehicles with the dynamics, and trajectory, as well as modeling the sensor which is the basis of the system. Arrangement of roads and buildings is based on the imported Street Map model of the Ahoy Arena area in Rotterdam, where the Shell Eco-marathon 2014 competition took place. The designed model of the environment (Figure 5) was supposed to reflect the actual conditions as faithfully as possible.

4.5.2. Vehicle Movement Parameters Modelling. After building the environment, we have to upload the vehicle model for which Advanced Driver Assistance System is designed. It is necessary to assign a trajectory and to define the parameters of vehicle movement.

TASS PreScan software allows implementation of any velocity profile and trajectory. First trajectories have to be defined and then velocity profile for the two vehicles has to be determined. As a result, speeds and acceleration and paths of the vehicles during the simulation are described in detail. Figure 6 shows the speed profile for a vehicle overtaking a leading Bytel vehicle.

4.5.3. Sensor Parameters Modelling. An important point of the simulation design process is accurate mapping of parameters of used sensor. The PreScan allows the use of the Lidar model; however, due to the large angular range and high frequency sampling, a better solution was to use a sensor of TIS type (Technology Independent Sensor) (Figure 7). TIS, thanks to the wide possibilities of modification of parameters, has enabled a very accurate representation of the Hokuyo sensor to be used in the BLIS assistance system.

PreScan software enables visual verification of the range of the proposed sensor (Figures 13, 14, and 15), which helps in controlling the distance and angular range. With this visualization, the right area of operation for the designed sensor can be easily chosen. By controlling the moment of appearance of another vehicle in sensor range, it is possible to verify operation during simulation of the system as early as at design stage (Figures 13, 14, and 15).

4.6. System Model: Data Processing System. After developing all the details of the simulation, it is allowed to start modelling of the system. TASS PreScan software cooperates with the MATLAB/Simulink software. When installing TASS PreScan the reference to MATLAB/Simulink is established. After starting PreScan Sim module user has ready-made models of sensors, vehicles, and other control elements visible as Simulink models. These models are generated by the TASS PreScan individually for each of the simulations taking into account the modifications described earlier.

4.6.1. System Main Window. The main window of system visible after opening the PreScan Sim module contains two models of vehicles and components fulfilling control warning LEDs functions in the simulation.

The main block model represented in the main window of PreScan Sim module contains modifications resulting in BLIS system design. The output from the model of the first Bytel vehicle that is a vehicle equipped with a sensor provides brief information on occurrence of alarms yellow and red and the area of detection of the object. At the input of each diode is given the information about activity in its assigned area and information about yellow or red alarm occurrence.

4.6.2. Vehicle Model. GUI window of vehicle model includes four groups of blocks:

(i) After entering the vehicle model (Figure 8), in the left side of the window, you can see models generated by the PreScan software describing it. In the case of a Bytel vehicle, it is a trajectory model, parameters motion model, and model of sensors assigned to the vehicle, TIS. Based on previously developed block diagram showing the planned method of the system operation, you can immediately pull out of these variables which are necessary when developing a BLIS model: x and y coordinates of the absolute position of the Bytel vehicle, rotation of the vehicle in the absolute coordinate system (vehicleRot), speed of the vehicle (V_v), the value of distance of detected obstacle (rangeTIS), and the relative speed of vehicles were determined using the TIS sensor (V_{dopp}) and the angle in which the obstacle appears (thetaTIS).

(ii) Blocks containing programs designed to perform the calculations of parameters used in the BLIS system: Object Position Calculation, Time To Collision, and Object Velocity Calculation. The blocks are linked to necessary input variables and output variables are derived.

(iii) Block of BLIS system together with its inputs and outputs

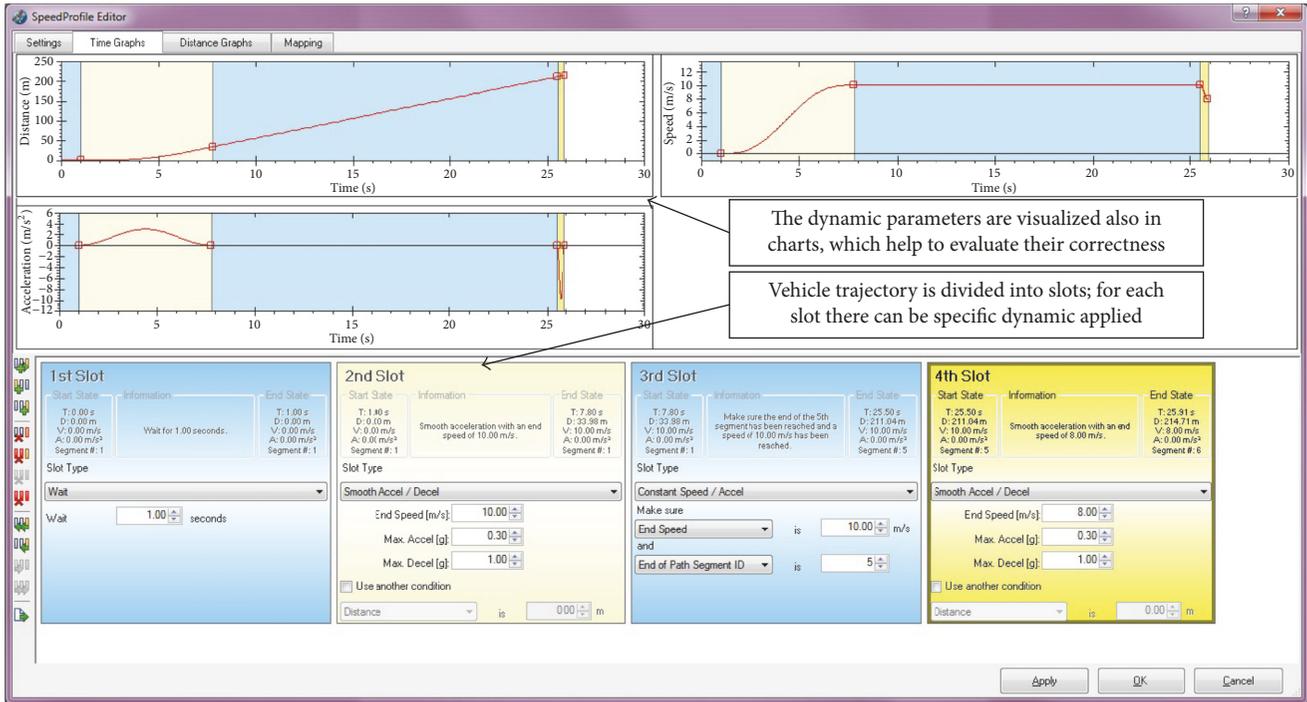


FIGURE 6: Speed profile for overtaking vehicle.

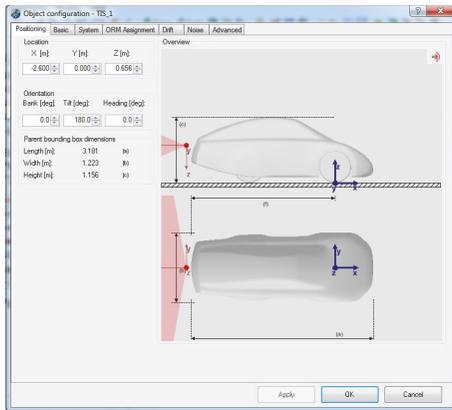


FIGURE 7: Modelling Hokuyo Lidar in BLIS.

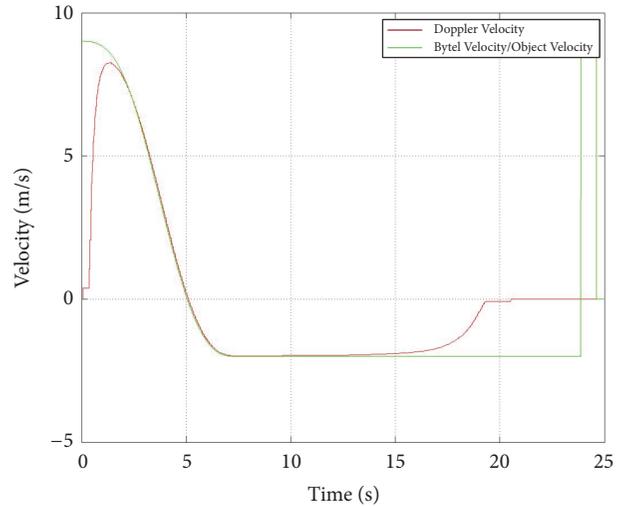


FIGURE 9: Comparison of two values: ideal one (Doppler Velocity) and the one achieved in data processing (Bytel Velocity, Object Velocity).

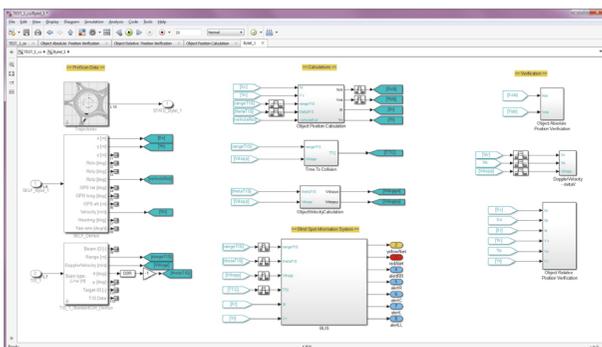


FIGURE 8: General models and blocks of the BLIS system.

(iv) Blocks including programs used to verification calculations

4.6.3. Part I: Calculations. Based on the block diagram of the BLIS system, it had to be determined which calculations should be performed to obtain the necessary to define the logic of the system variables. Directly from the PreScan software, the following variables were extracted:

- (i) X and Y coordinates of Bytel vehicle in the absolute coordinate system
- (ii) Rotation angle of the vehicle in absolute coordinate system (vehicleRot)
- (iii) Vehicle velocity (V_v),
- (iv) The distance of the detected obstacle (rangeTIS)
- (v) Relative velocity determined by TIS sensor (V_{dopp})
- (vi) The angle at which obstacle was detected (thetaTIS)

These variables do not provide direct way to determine the logic of the system, which was based on three basic pieces of information:

- (i) Comparison of the relative position of the object with the value of YCD (Yellow Critical Distance) or RCD (Red Critical Distance), which are functions defining the threshold value of position of the object in the x axis, for which there is raised up risk of collision
- (ii) Value of relative velocity of vehicles V_{dopp}
- (iii) TTC value (Time To Collision)

On the basis of these variables, the following result variables were obtained in the way of calculations:

- (i) X_r : x coordinate of the object relative position relative to Bytel vehicle determined based on the value rangeTIS and thetaTIS
- (ii) Y_r : Y coordinate of the object relative position relative to the Bytel vehicle determined based on the value rangeTIS and thetaTIS
- (iii) Yellow Critical Distance: function determined based on the value Y_r
- (iv) Red Critical Distance: function determined based on the value Y_r
- (v) Time To Collision: time to collision, determined based on the value rangeTIS and V_{dopp}

4.6.4. Object Position Calculation. The activities carried out in Object Position Calculation block are divided into two sub-systems: the calculation of the value of X and Y coordinates relative position of the detected object and calculation of the coordinate values of X and Y absolute position of the object.

Program used to calculate the values of the X and Y coordinates relative position of the detected object accepts as input values thetaTIS and rangeTIS. Values X_r and Y_r determining the relative position of the vehicle in the x and y axis are obtained by transforming the coordinate system of polar to Cartesian coordinate system.

The second part of the Object Position Calculation block represents calculations regarding the position of the object in the absolute coordinate system. The program uses the input values: thetaTIS, vehicleRot, rangeTIS, coordinates X_v and Y_v , point rotation matrix in the two-dimensional space, and value of coordinates and the value o displacement vector of the center of the local coordinate system of the Bytel vehicle from the edge detected in the x and y axis.

An important assumption was rangeTIS $= 0$, because the value 0 is assigned to variable rangeTIS at a time when there are any object detected in sensor range. In this case, variables X_{ob} and Y_{ob} are assigned the value 0.

4.6.5. Time To Collision. Time To Collision block task is to calculate the value of time, from the temporary situation of the temporary traffic parameters to the potential collision. This is an important element of calculations, because the value of TTC is directly influenced by the decision to activate the yellow or red alarm.

Also, here switch block was used making passed signal dependent on the value of the variable V_{dopp} . The variable takes the value 0 when there are no objects detected in the sensor range sensor. In this case, the variable TTC is assigned the value 20 as the maximum time for which the threat of a collision is practically nonexistent.

4.6.6. Object Velocity Calculation. Then, the third block of the calculation is an Object Velocity Calculation. It contains code to calculate the coordinates relative speed values V_{dopp} . Therefore, a distribution of the vector velocity V_{dopp} on the components of X and Y depends on the angle thetaTIS. Output values are V_{dopp_x} and V_{dopp_y} .

4.6.7. Part II: Verification. The second part of the prepared program was verification part. Before modelling BLIS system, it had to be checked whether calculations delivering the required input values to the system are correct. For this purpose, the values were compared with values generated by the simulation environment (PreScan).

Verification of the relative speed values V_{dopp} : relative speed V_{dopp} is called the Doppler Velocity, which results in the fact that its value is not always unequivocal from the expected one. In order to check the difference between the variable V_{dopp} and the actual vehicle speeds difference verification program was elaborated. The example results can be seen in Figure 9.

Verification of X and Y coordinates relative position of the object is as follows.

To verify the calculations of the value of X and Y coordinates relative position of the object (relative to the Bytel vehicle), the computed variables were compared with variables X_r and Y_r with the difference of the values of the coordinates of the vehicle Bytel (X_v, Y_v) and the object (X_o, Y_o) obtained artificially by PreScan software, taking into account the offset point to be the center of the coordinate system of the vehicle, to whom they are given the coordinates of absolute position.

Verification of coordinates X and Y absolute position of the object is as follows. In order to verify the calculation of the X and Y coordinates of the relative position of the object (relative to the Bytel vehicle), the computed variables X_{ob} and Y_{ob} were compared with the object position coordinates (X_o, Y_o) obtained artificially from the PreScan software.

4.6.8. Part III: BLIS System. The last step was to build the logics of the BLIS system. This part of the system consists of

three parts: Data, Alert Range Assignment, and Alert Colour Assignment.

Data. This section of the BLIS window contains ordered model input values, blocks of recording of signals to the workspace.

Alert Range Assignment. The part called Alert Range Assignment is responsible for assigning to the appropriate areas of the alarm (LL, L, C, R, RR) angular ranges in which the object appears.

- (i) Range LL is assigned for $\theta_{TIS} \langle 48; 87.5 \rangle$ deg.
- (ii) Range L is assigned for $\theta_{TIS} \langle 7; 49 \rangle$ deg.
- (iii) Range C is assigned for $\theta_{TIS} \langle -8; 8 \rangle$ deg.
- (iv) Range R is assigned for $\theta_{TIS} \langle -49; -7 \rangle$ deg.
- (v) Range RR is assigned for $\theta_{TIS} \langle -87.5; -48 \rangle$ deg.

Alert Assignment Colour. Code fragment entitled Alert Colour Assignment is responsible for assigning the alarm corresponding colour in accordance with the degree of threat, yellow or red.

Decisions on the colour of the alarm are taken on the basis of three pieces of information:

- (i) Comparison of the position of the object relative to the value of YCD (Yellow Critical Distance) or RCD (Red Critical Distance), which are functions defining the threshold value of position of the object in the x axis, for which there is an increased risk of collision
- (ii) Values V_{dopp}
- (iii) The value of TTC (Time To Collision)

The YCD function (Yellow Critical Distance) is given by

$$YCD = -4 * |Y_r| + 12. \quad (1)$$

The RCD (Red Critical Distance) is given by

$$RCD = -\frac{4}{3} * |Y_r| + 4. \quad (2)$$

Yellow alarm occurs when the following conditions are met:

$$(X_r \leq YCD \text{ (yellowFun)} \cap X_r \neq 0 \cap V_{dopp} \leq 0) \cup 4 < TTC \leq 10. \quad (3)$$

Red alarm occurs when the following conditions are met:

$$(((X_r \leq RCD \text{ (redFun)} \cap X_r \neq 0) \cup 0 < X_r \leq 1) \cap V_{dopp} \leq 0) \cup 0, 1 < TTC \leq 4. \quad (4)$$

Figure 10 visualizes the conditions for activating the alarms.

- X_r : X coordinate of relative position of the object
- YCD: Yellow Critical Distance
- RCD: Red Critical Distance
- V_{dopp} : relative velocity of the object
- TTC: Time To Collision

4.7. System Model: Control System. Having data processing system, we can begin to design the control system of warning LEDs. In the previous section, signals have been mentioned that we receive at the output of the processing unit. The system of transmission between the BLIS system and the LED is shown in Figure 11 and is the same for each of the diodes.

On the output from the Bytel vehicle block, we have seven binary signals:

- (i) yellowAlert: the state of yellow alarm
- (ii) redAlert: the state of red alarm

- (iii) alertRR: the state of alarm for the RR range
- (iv) alertR: the state of alarm for the R range
- (v) alertC: the state of alarm for the C range
- (vi) alertL: the state of alarm for the L range
- (vii) alertLL: the state of alarm for the LL range

At the input of each of the LEDs blocks, the following signals are given:

- (i) yellowAlert: state of yellow alarm
- (ii) redAlert: state of red alarm
- (iii) alertX: state of alarm for the corresponding diode range (RR, R, C, L, and LL)

For each of the five LEDs used to display warning information, the same control was developed.

4.7.1. Part I: System Activation. The logics of the system is based on a switch block which, depending on the condition of the simultaneous occurrence of any of the alarms and



FIGURE 10: Logic of the red and yellow alarm.

the activation of the range corresponding to LED (in this case the LL range), transmits a signal specifying the colour of the alarm (equivalent to the activation of the alarm) or transmits the signal from the default, LEDs gray colour. The same condition checked in switch block acts as a trigger of alertColour block.

4.7.2. Part II: Determination of Alarm Colour. The condition mentioned above checked in switch block acts simultaneously as a trigger of alertColour block.

In the case where you get any of the alarms (yellowAlarm or redAlarm) and at the same time the object was detected in the corresponding range, the block alertColour is executed. This block is to assign a diode suitable for the degree of danger colour: the colour red for a higher degree of threat and the yellow colour for a lower degree of threat. In the case where there is information about the appearance of red alert and activating the appropriate range, signal carrying information about the red alarm is passed, if not, yellow. This logic makes the red alarm the priority.

5. Testing

The test phase of the design process is often treated in a disrespectful way, but properly planned and carried out tests allow you to determine the imperfections and errors of the project very accurately and at the same time to facilitate their improvement greatly. During test planning, the first step is to verify the correctness of the calculations made in the model. In order to determine what we should investigate properly, it is necessary to answer the following questions:

- (i) What are the inputs to the model?
- (ii) What are the outputs from the model?
- (iii) How outputs were from the model obtained?
- (iv) Which of these activities relate to the independent variables of the operation of the system, which are the physical representation of unknown values?

Answering these questions will allow the identification of transformation, which should be explored in terms of their compliance with the physical values that do not belong to a set of values of the input model. PreScan software provides many of these values, which makes the comparison of the model with the actual data possible without an actual field testing.

The next step in the test planning is to return to the design assumptions and answer the questions:

- (i) What is the designed system supposed to do? What are the requirements?

- (ii) What information does the system provide the user with?
- (iii) What should not the system do?

The above questions relate to the direct effects of the system, its functionality, and purpose but they do not specify requirements regarding the project completely. The tests are considered for remaining elements, which are invisible to the user and dependent largely on the designer of the system. These elements are the basis of the system. In order to determine them, it is necessary to answer the following questions:

- (i) How should the designed system operate?
- (ii) What inputs and outputs should it generate/set?
- (iii) What functions are its main characteristics?

The answers to these questions should be clarified on the basis of the design phase, the software built, and the expected behaviour of a system. Having already collected information about the system sorted out according to the above scheme, it is possible to begin to plan test case studies. What is most important in the operation of the safety system is its correct operation and efficiency. If the system is to comply with the quite simple functions (e.g., warn the driver about a negative temperature), it is not necessary to develop complex test scenarios.

The case is complicated, however, when the system has a more extensive set of functionalities, when it has to react differently in different situations, when the output value depends on the value of larger quantities of inputs, and where there is the possibility of false alarms. In this case, it is good to previously develop scenarios to test and test cases for each scenario. After tests, a summary of the verification of the correctness of the system should be made. In the case where the system does not meet one (or several) of the designated test case studies, it is necessary to carry out repair procedures. This phase resembles the design phase, but instead of building the system from the beginning, repairs are carried out and changes to the existing system on the basis of data obtained in the test phase are introduced. Well-conducted tests allow us to identify errors in the system quickly and often arise the way for solutions. In this chapter, the test phase for BLIS system design for the Bytel vehicle will be described.

5.1. Tests Plan

5.1.1. Calculations Verification. Based on the previously elaborated flowchart, the first step of the procedure of Blind Spot Information System test planning for Bytel vehicle is the answer to the question about the correctness of the calculations implemented in the model.

- (i) What are inputs to the model?

In the model, the following inputs are used:

- (a) X_v , [m]: X coordinate of Bytel vehicle in absolute coordinate system

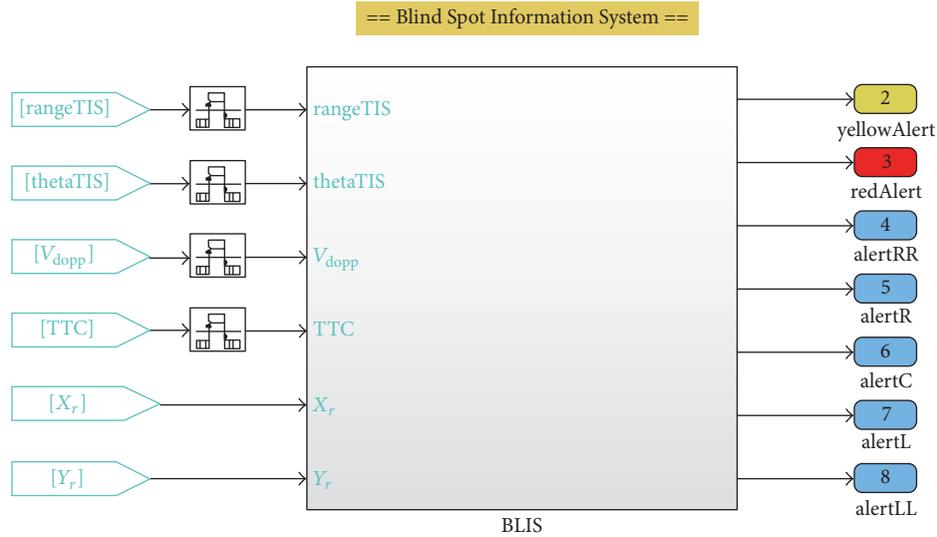


FIGURE 11: BLIS block.

- (b) Y_v [m]: Y coordinate of Bytel vehicle in absolute coordinate system
- (c) vehicleRot [$^\circ$]: rotation angle related to z axis of Bytel vehicle
- (d) V_v [m/s]: Bytel vehicle velocity
- (e) rangeTIS [m]: distance from the sensor to detected object
- (f) V_{dopp} [m/s]: relative vehicles velocity
- (g) thetaTIS [$^\circ$]: angle at which obstacle was detected

(ii) What are the outputs of model?

In the model, the following outputs are generated:

- (a) yellowAlert (0/1): yellow alarm status
- (b) redAlert (0/1): red alarm status
- (c) alertLL (0/1): LL alarm status
- (d) alertL (0/1): L alarm status
- (e) alertC (0/1): C alarm status
- (f) alertR (0/1): R alarm status
- (g) alertRR (0/1): RR alarm status

(iii) How are the above outputs generated?

Decisions on the classification of the alarm due to the level of risk (yellow or red) system are taken on the basis of three pieces of information:

- (a) Comparison of relative position of object with value of YCD (yellow Critical Distance) or RCD (Red Critical Distance), which are functions determining threshold values of object along x axis for which increased risk of collision exists
- (b) V_{dopp} value
- (c) TTC value (Time To Collision)

Decision on selection of the area, which is assigned to an alarm, depends only on the value of the angle thetaTIS.

- (iv) Which of these activities relate to the independent variables from the operation of the system and which are the physical representation of unknown values?

Variable determined by calculation are as follows:

- (a) X_r : X coordinate of object position related to Bytel vehicle determined on the basis of values of rangeTIS and thetaTIS
- (b) Y_r : Y coordinate of object position related to Bytel vehicle determined on the basis of values of rangeTIS and thetaTIS
- (c) Yellow Critical Distance: function determined on the basis of Y_r value
- (d) Red Critical Distance: function determined on the basis of Y_r value
- (e) Time To Collision: time to collision determined on the basis of rangeTIS and V_{dopp} values

Based on this information, it was decided to examine the above-mentioned dependence by comparing them with the values of the corresponding variables artificially generated by the PreScan software. An example of such a comparison can be seen in Figure 12.

5.1.2. Verification from the User Perspective: Visual Verification. The next step is to answer the questions about system importance from the perspective of the user (Figures 13 and 14).

- (i) What should the designed system do?

The purpose of the system is to inform the user of the location of another vehicle behind the driven one, which is not visible to the driver, or its notification

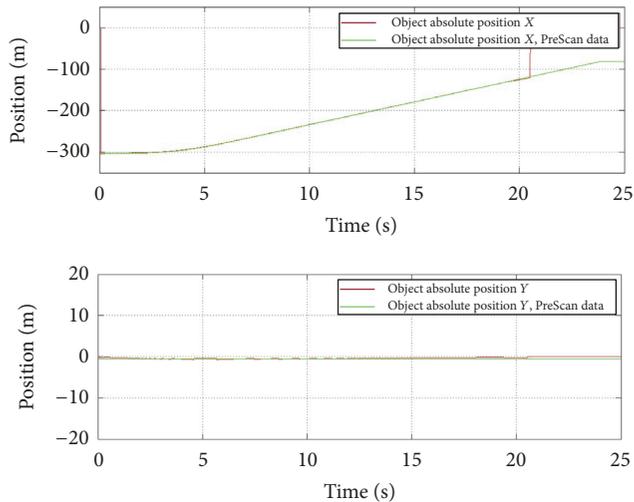


FIGURE 12: Comparison of corresponding variables determined by virtual BLIS system and directly by PreScan.

requires too much effort of the driver. The information should be communicated in a visual way, using LEDs.

- (ii) What information should the system provide the user with?

System is to inform the user of the observation of the vehicle in one (or more) of the five ranges designated for the driven vehicle and the degree of risk associated with the event. For low-level threats, the alarm should accept yellow colour and for high threat red.

- (iii) What should not the system do?

The system should not react to vehicles in the area of the sensor, but in a place visible to the driver of the vehicle. The system also should not generate false positives: that is, the alarm is not caused by the appearance of the object in the field of sight of the driver.

5.1.3. Numerical Verification. The next step is to analyse important questions from the perspective of the creators of the system (Figure 16):

- (i) How the designed system should operate?

The system is intended to operate on the basis of information from the vehicle (vehicle location, vehicle speed), and the information from the sensor (the distance of the detected object from the sensor, the angle at which the object appears, Doppler speed). The system should set, on the basis of the data, relative location information of vehicles to each other and the speed difference between vehicles. The system should assign one of the five predesignated areas to a given position. Based on the information about the speed difference between vehicles and their relative position system, it should set TTC (Time To Collision). Depending on the value of TTC and the distance between the vehicles, the system should assign areas

of the degree of risk: no risk, the LED is not lit; low threat, LED yellow; high risks, the LED lights red.

- (ii) What input and output values should it take?

The system should set the following inputs and outputs:

- Distance of detected object from driven vehicle: 0–40 [m]
- Velocity of driven vehicle: 0–15 [m/s]
- Relative velocity of vehicles: 0–15 [m/s]
- Position of the vehicle on x , y axis [m]
- Angle of appearance of object: $-87,5^\circ$ – $87,5^\circ$

The system should generate the following output values:

- Value 0/1 on output corresponding to LL, L, C, R, and RR LEDs
- Value 0/1 on output corresponding to yellow alarm
- Value 0/1 on output corresponding to red alarm

- (iii) What functions characterize it?

- TTC (Time To Collision) for every moment when object is within the sensor range
- RangeTIS for every moment when object is in the sensor range
- Relative object and Bytel vehicle position: X_r , Y_r
- ThetaTIS angle, in which the object detected remains for each moment of time in which the object is located within the sensor range
- Relative object and Bytel vehicle velocity

5.1.4. The Development of a Set of Test Cases. Based on the information collected, Table 1 was prepared containing test cases, which were used to verify the operation of the system BLIS.

5.2. Test Case Scenario: Vehicle Overtaken from the Left Side

5.2.1. Description of the Scenario. The object comes from the left side of the vehicle Bytel. The ratio of speed of an object to Bytel vehicle speed is in the range of 0 to 1.5 for each moment of virtual experiment. The speed difference between the object and the Bytel vehicle is not greater than 10 m/s. The speed of the object and the vehicle does not exceed 12 m/s (Figures 13, 14, and 15).

5.2.2. Results Verification. In order to investigate the above-mentioned cases, we have to use different methods. For testing cases (1) (Table 1) and (3) (Table 1), it was necessary to analyse the relevant variables for cases (2) (Table 1) and visual inspection simulation is sufficient.

Calculation Verification. In order to check the correctness of the algorithms implemented in the BLIS system model comparative charts were used comparing variables with variables generated by the PreScan software.

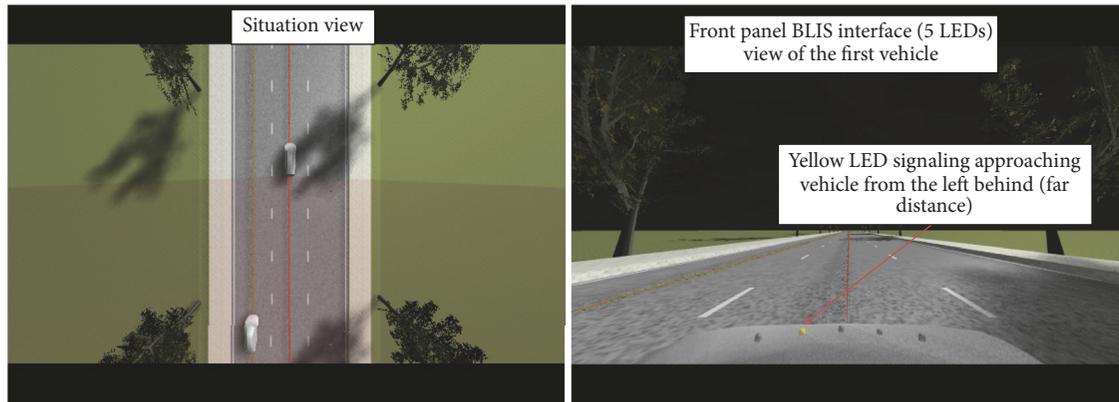


FIGURE 13: Example of visual verification, scenario 1, overtaking phase 1.

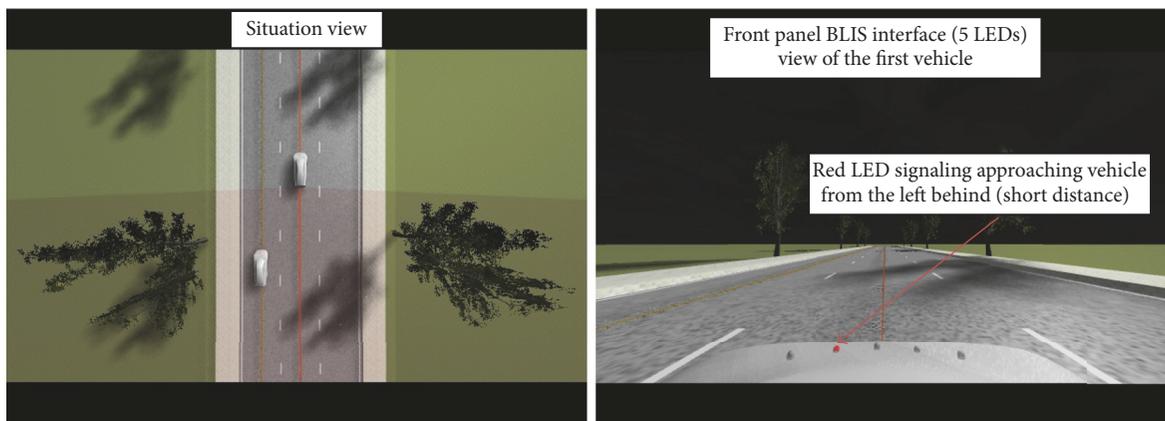


FIGURE 14: Example of visual verification, scenario 1, overtaking phase 2.

- (i) Comparison of V_{dopp} speed with the difference in velocity of the Bytel vehicle and the object overtaking Bytel; both speeds are available in the PreScan software.

Based on a comparison graph of the Doppler Velocity and difference of vehicles velocities, it can be concluded that the mapping of Doppler Velocity does not fully coincide with the actual vehicles speeds difference. The biggest differences can be seen for the time intervals: 0–2 s and 16–19 s. This is connected with the moment of occurrence of the object in a short distance from the sensor. Despite these cases, it is concluded that the proposed function of the velocity can be used in the design of system.

- (ii) Comparison of the relative position in the x and y axis of overtaking object with respect to the Bytel vehicle, calculated using information about the distances of overtaking object and the angle in which it is located, with a difference of location coordinates of the vehicle and the object with a leading Bytel; coordinates are available in the PreScan software.

Based on the comparison charts X and Y calculated using data on the distance and angle of the object

overtaking with the actual difference in the position of the Bytel vehicle and the overtaking object, it was found that the adopted model calculation of the relative position of the object is correct. In addition, the graphs present compliance of variables only until reaching 20th second. It is associated with the period when the object is within the range of the sensor, which enabled the calculation of its relative position.

- (iii) Comparison of the absolute position of the x and y axis overtaking object calculated using information about the distances of overtaking object, the angle at which it is located and the coordinates of the Bytel vehicle, the absolute position of the object, which is made available in the PreScan software.

Based on the comparison charts X and Y calculated using data on the distance, angle and coordinates of the object overtaking the Bytel vehicle with the actual position of the absolute object, it was found that the adopted computational model of the absolute position of the object is correct. The graphs show compliance only until reaching 20th second. It is associated with the period when the object is within the range of the sensor, which enabled the calculation of its relative position.

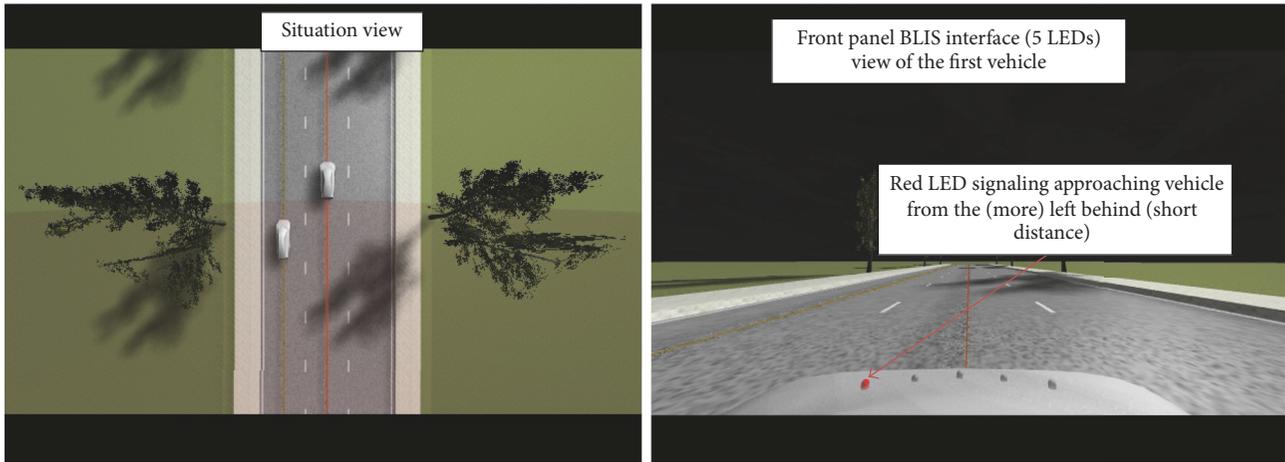


FIGURE 15: Example of visual verification, scenario 1, overtaking phase 3.

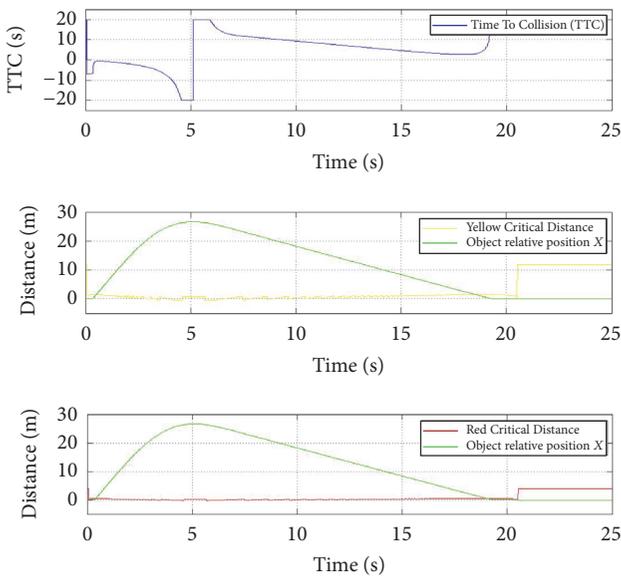


FIGURE 16: TTC (Time To Collision), comparison of YCD (Yellow Critical Distance), and RCD (Red Critical Distance) with object relative position X.

Visual Verification (Figures 13, 14, and 15)

- (i) The object is coming from the direction established in the description of the scenario: object is coming from the left.

Based on the simulation results, the system has been found to meet this condition.

- (ii) The object is located within the sensor range: it was possible to carry out tests of the BLIS system and the object must be in range of the sensor.

Based on the simulation results, the system has been found meeting the above condition.

- (iii) Operation of warning LL LEDs, since the object is overtaking on the left side of the Bytel vehicle after fulfilment of additional conditions LED should light

up depending on the object distances in red and/or yellow.

Based on the simulation results, the system has been found meeting the above condition.

- (iv) Operation of warning L LEDs, since the object is overtaking on the left side of the Bytel vehicle after fulfilment of additional conditions LED should light up depending on the object distances in red and/or yellow.

Based on the simulation results, the system has been found to meet the above condition.

- (v) Operation of warning C LED, since the overtaking object at any moment in time is located centrally behind the Bytel vehicle the LED should not light up.

Based on the simulation results, the system has been found to meet the above condition.

- (vi) Operation of warning R LED, since the overtaking object at any moment in time is located from the right side of Bytel vehicle the LED should not light up.

Based on the simulation results, the system has been found to meet the above condition.

- (vii) Operation of warning RR LED, since the overtaking object at any moment in time is located from the right side of Bytel vehicle, the LED should not light up.

Based on the simulation results, the system has been found to meet the above condition.

Numerical Verification

- (i) Inspection of Bytel vehicle velocity:

According to the assumptions of this scenario, Bytel vehicle velocity at any moment in time should not be greater than 12 m/s. Based on the velocity chart of the Bytel vehicle the system has been found to meet this assumption.

- (ii) Inspection of overtaking object velocity:

TABLE 1: Test cases used to verify the operation of the BLIS system.

Number	Test description	Initial conditions	Expected value	Outcome
(1)	Calculation verification			
(1.1)	X_r position	$\text{rangeTIS} \neq 0$	$X_r \cong X_v - X_o$	
(1.2)	Y_r position	$\text{rangeTIS} \neq 0$	$Y_r \cong Y_v - Y_o$	
(1.3)	Yellow Critical Distance	$\text{rangeTIS} \neq 0$	$\text{YCD} = -4 * \text{abs}(Y_r) + 12$	
(1.4)	Red Critical Distance	$\text{rangeTIS} \neq 0$	$\text{RCD} = -4/3 * \text{abs}(Y_r) + 4$	
(1.5)	Time To Collision	$V_{\text{dopp}} \neq 0$	$\text{TTC} = -\text{rangeTIS}/V_{\text{dopp}}$	
(2)	Visual tests			
(2.1)	Object approaches from direction assumed in the scenario description	-	Left/right side, behind the object	
(2.2)	Object is in the sensor range	-	The object is in the sensor range	
(2.3)	Correct functioning of the LL warning LED	Object in the LL area, danger	LED lights up Y/R	
(2.4)	Correct functioning of the L warning LED	Object in the L area, danger	LED lights up Y/R	
(2.5)	Correct functioning of the C warning LED	Object in the C area, danger	LED lights up Y/R	
(2.6)	Correct functioning of the R warning LED	Object in the R area, danger	LED lights up Y/R	
(2.7)	Correct functioning of the RR warning LED	Object in the RR area, danger	LED lights up Y/R	
(3)	Computational tests			
(3.1)	Object velocity/Bytel vehicle velocity	$V_v \neq 0$	≥ 1	
(3.2)	Object velocity/Bytel vehicle velocity	$V_v \neq 1$	$\leq 10 \text{ m/s}$	
(3.3)	Object velocity	-	$\leq 12 \text{ m/s}$	
(3.4)	Bytel vehicle velocity	-	$\leq 10 \text{ m/s}$	
(3.5)	Turning on yellow alert	$4 \leq \text{TTC} \leq 10 \cup (X_r \leq \text{YCD} \cap V_{\text{dopp}} < 0 \cap X_r \neq 0)$	$\text{yellowAlert} = 1$	
(3.6)	Turning on red alert	$0.1 \leq \text{TTC} \leq 4 \cup (X_r \leq \text{RCD} \cap V_{\text{dopp}} < 0 \cap X_r \neq 0)$	$\text{redAlert} = 1$	
(3.7)	LL area control	$48 \leq \text{thetaTIS} \leq 87.5$	$\text{alertLL} = 1$	
(3.8)	L area control	$7 \leq \text{thetaTIS} \leq 49$	$\text{alertL} = 1$	
(3.9)	C area control	$8 \leq \text{thetaTIS} \leq -8$	$\text{alertC} = 1$	
(3.10)	R area control	$-49 \leq \text{thetaTIS} \leq -7$	$\text{alertR} = 1$	
(3.11)	RR area control	$-87.5 \leq \text{thetaTIS} \leq -48.5$	$\text{alertRR} = 1$	

According to the assumptions of this scenario overtaking vehicle velocity at any moment in time should not be greater than 12 m/s. Based on the velocity chart of the overtaking vehicle the system has been found to meet this assumption.

(iii) Inspection of object and Bytel vehicle velocities ratio:

According to the assumptions of this scenario, ratio of velocities of object to Bytel vehicle velocity at any moment in time should not be less than 1. Based on the velocities ratio chart, the system has been found to meet this assumption.

(iv) Inspection of object and Bytel vehicle velocities difference:

According to the assumptions of this scenario, the difference of overtaking vehicle velocity and Bytel vehicle velocity at any moment in time should not be greater than 10 m/s. Based on the vehicles velocities difference chart system has been found to meet this assumption.

(v) Inspection of alarm activation:

In order to inspect correctness of activation of yellow and red alarm, the following charts were used: TTC (Time To Collision), YCD (Yellow Critical Distance), and RCD (Red Critical Distance) compared to X_r (relative position of object in x axis). Charts are shown in Figure 16.

(vi) Inspection of yellow alarm activation:

According to the assumptions of the design, the yellow alarm should be activated under the following conditions:

- (a) TTC (Time To Collision) ranges from 4 to 10 s or
- (b) the value of the relative position of the object in the x -axis representing the longitudinal axis of the Bytel vehicle is less than the value of the function YCD (Yellow Critical Distance).

Based on the charts of relative position of the object in the x -axis and functions of Yellow Critical Distance, charts of TTC and yellow alarm activity system have been found to meet this assumption.

- (vii) Inspection of red alarm activation:

According to the assumptions of the design, the red alarm should be activated under the following conditions:

- (a) TTC (Time To Collision) ranges from 0,1 to 4 s or
- (b) the value of the relative position of the object in the x -axis representing the longitudinal axis of the Bytel vehicle is less than the value of the function RCD (Red Critical Distance)

Based on the charts of relative position of the object in the x -axis and functions of Red Critical Distance, the charts of TTC and red alarm activity system have been found to meet this assumption.

- (viii) Inspection of operation of LL alarm corresponding to extreme left scanning area:

As designed LL alarm should be activated when the vehicle is detected in an angular range from 48° to 87.5° . Based on the charts, TIS sensor range, the angle of appearance of the object, and graph LL alarm activity system have been found to meet this assumption.

- (ix) Inspection of operation of L alarm corresponding to left scanning area:

As designed L alarm should be activated when the vehicle is detected in an angular range from 7° to 49° . Based on the charts, TIS sensor range, the angle of appearance of the object, and graph L alarm activity system have been found to meet this assumption.

- (x) Inspection of operation of C alarm corresponding to central scanning area:

As designed, C alarm should be activated when the vehicle is detected in an angular range from -8° to 8° . Based on the charts, TIS sensor range, the angle of appearance of the object, and graph C alarm activity system have been found to meet this assumption.

- (xi) Inspection of operation of R alarm corresponding to right scanning area:

As designed R alarm should be activated when the vehicle is detected in an angular range from -49° to -7° . Based on the charts, TIS sensor range, the angle of appearance of the object, and graph R alarm activity system have been found to meet this assumption.

- (xii) Inspection of operation of RR alarm corresponding to extreme right scanning area:

As designed RR alarm should be activated when the vehicle is detected in an angular range from -87.5° to -48° . Based on the charts, TIS sensor range, the angle of appearance of the object, and graph RR alarm activity system have been found to meet this assumption.

5.2.3. *Final Results of Virtual Verification.* Based on the above results, Table 1 of test cases was completed, and it was found that tests for scenario I were completed successfully as can be seen in Table 2.

6. Consideration of Possible Improvements

Although the final result of applying the described method was satisfying, there were still areas for the improvement. First of all there should be stronger emphasis put on the iterative approach to design, combining design phase with testing phase.

In this context, it is important to choose the scenarios of simulation tests carefully and, on the basis of subsequent experience, expand the list of these scenarios to the new situations observed in the race conditions.

What also would be of a great value is real-life implementation system and comparison of real-life values with outputs received from simulation. This could help to identify the areas where the described method does not prove correct or needs improvement.

Similarly, as TASS, PreScan enables also creating experiments for various weather conditions and it would be valuable to test the designed system for different atmospheric conditions. It is especially true for systems that assume using sensors which output values are weather-dependent (e.g., sonars, Lidars).

There is considerable potential for the transfer of work results related to the BLIS system as well as its methodology for commercial development. The particular advantage of the developed system is its modularity and the ability to adapt to other types of vehicle and other operating conditions.

7. Conclusions

The described method of Advanced Driver Assistance System design, with the use of virtual prototyping tool, has been applied in real-life project. The outcome of taking this approach can be evaluated as successful, since the reasons for using this kind of method have been proven and the final results have been considered satisfying.

Using simulation method in ADAS design shortens work time, makes the iterative approach to design easier and

TABLE 2: Approved results of test cases.

Number	Test description	Initial conditions	Expected value	Outcome
(1)	Calculation verification			Approved
(1.1)	X_r position	rangeTIS $\neq 0$	$X_r \cong X_v - X_o$	Approved
(1.2)	Y_r position	rangeTIS $\neq 0$	$Y_r \cong Y_v - Y_o$	Approved
(1.3)	Yellow Critical Distance	rangeTIS $\neq 0$	$YCD = -4 * \text{abs}(Y_r) + 12$	Approved
(1.4)	Red Critical Distance	rangeTIS $\neq 0$	$RCD = -4/3 * \text{abs}(Y_r) + 4$	Approved
(1.5)	Time To Collision	$V_{\text{dopp}} \neq 0$	$TTC = -\text{rangeTIS}/V_{\text{dopp}}$	Approved
(2)	Visual tests			Approved
(2.1)	Object approaches from direction assumed in the scenario description	-	Left/right side, behind the object	Approved
(2.2)	Object is in the sensor range	-	The object is in the sensor range	Approved
(2.3)	Correct functioning of the LL warning LED	Object in the LL area, danger	LED lights up Y/R	Approved
(2.4)	Correct functioning of the L warning LED	Object in the L area, danger	LED lights up Y/R	Approved
(2.5)	Correct functioning of the C warning LED	Object in the C area, danger	LED lights up Y/R	Approved
(2.6)	Correct functioning of the R warning LED	Object in the R area, danger	LED lights up Y/R	Approved
(2.7)	Correct functioning of the RR warning LED	Object in the RR area, danger	LED lights up Y/R	Approved
(3)	Computational tests			Approved
(3.1)	Object velocity/Bytel vehicle velocity	$V_v \neq 0$	≥ 1	Approved
(3.2)	Object velocity/Bytel vehicle velocity	$V_v \neq 1$	≤ 10 m/s	Approved
(3.3)	Object velocity	-	≤ 12 m/s	Approved
(3.4)	Bytel vehicle velocity	-	≤ 10 m/s	Approved
(3.5)	Turning on yellow alert	$4 \leq TTC \leq 10 \cup (X_r \leq YCD \cap V_{\text{dopp}} < 0 \cap X_r \neq 0)$	yellowAlert = 1	Approved
(3.6)	Turning on red alert	$0.1 \leq TTC \leq 4 \cup (X_r \leq RCD \cap V_{\text{dopp}} < 0 \cap X_r \neq 0)$	redAlert = 1	Approved
(3.7)	LL area control	$48 \leq \text{thetaTIS} \leq 87.5$	alertLL = 1	Approved
(3.8)	L area control	$7 \leq \text{thetaTIS} \leq 49$	alertL = 1	Approved
(3.9)	C area control	$8 \leq \text{thetaTIS} \leq -8$	alertC = 1	Approved
(3.10)	R area control	$-49 \leq \text{thetaTIS} \leq -7$	alertR = 1	Approved
(3.11)	RR area control	$-87.5 \leq \text{thetaTIS} \leq -48.5$	alertRR = 1	Approved

faster, enables early error detection and identification, and encourages parallel work. Therefore, it can be stated that using this method results also in costs reduction and system quality improvement.

The elimination of the need to verify the system in practice is due to the use of algorithms that verify the simulation results without allowing uncritical reliance on simulation results solely.

The high reliability of the BLIS system itself was achieved by taking into account not only the position of the adjacent vehicles, but also their direction of travel as well as the relative speed of the vehicles. This made it possible to calculate the potential collisions and predict the time it could take place (TTC). This approach eliminates false alarms of

most nonroad traffic situations, despite the relatively close proximity of the vehicle to other objects on the track or in the immediate vicinity.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] D. Ammon and W. Schiehlen, "Advanced road vehicles: Control technologies, driver assistance," in *Dynamical Analysis of Vehicle Systems. Theoretical Foundations and Advanced Applications*, W. Schiehlen, Ed., pp. 283–304, Springer, New York, NY, USA, 2009.

- [2] F. Coşkun, Ö. Tuncer, E. Karşlıgil, and L. Güvenç, "Vision system based lane keeping assistance evaluated in a hardware-in-the-loop simulator," in *Proceedings of the ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis, ESDA2010*, pp. 103–114, Istanbul, Turkey, July 2010.
- [3] O. J. Gietelink, J. Ploeg, B. De Schutter, and M. Verhaegen, "Development of a driver information and warning system with vehicle hardware-in-the-loop simulations," *Mechatronics*, vol. 19, no. 7, pp. 1091–1104, 2009.
- [4] A. D. Hall, *A methodology for Systems Engineering*, Van Nostrand, 1962.
- [5] A. Belbachir, J. Smal, J. Blossville, and D. Gruyer, "Simulation-driven validation of advanced driving-assistance systems," *Transport Research Arena – Europe*, vol. 48, pp. 1205–1214, 2012.
- [6] K. Jezierska-Krupa and W. Skarka, "Using Simulation Method for Designing ADAS Systems for Electric Vehicle. Proceedings of the 23rd ISPE Inc. International Conference on Transdisciplinary Engineering Location: Fed Univ Technol, Curitiba, Brazil, October, 2016," in *Transdisciplinary Engineering: Crossing Boundaries*, M. Borsato, N. Wognum, and M. Peruzzini, Eds., vol. 4 of *Advances in Transdisciplinary Engineering*, pp. 595–604, 2016.
- [7] E. Hull, K. Jackson, and J. Dick, *Requirements Engineering*, Springer Science & Business Media, 2005.
- [8] G. Pahl and W. Beitz, *Engineering Design: A Systematic Approach*, Design Council/Springer, 1988.
- [9] W. Schiehlen and D. Ammon, *Advanced Road Vehicles: Control Technologies, Driver Assistance*, Institute of Engineering and Computational Mechanics, University of Stuttgart, Stuttgart, Germany.
- [10] B.-F. Wu, H.-Y. Huang, C.-J. Chen, Y.-H. Chen, C.-W. Chang, and Y.-L. Chen, "A vision-based blind spot warning system for daytime and nighttime driver assistance," *Computers and Electrical Engineering*, vol. 39, no. 3, pp. 846–862, 2013.
- [11] A. Cholewa, W. Skarka, K. Sternal, and M. Targosz, "Electric vehicle for the students' shell eco-marathon competition. design of the car and telemetry system," in *Telematics in the Transport Environment, Proceedings of the 12th International Conference on Transport Systems Telematics, TST 2012, Katowice-Ustroń, Poland, October, 2012*, J. Mikulski, Ed., Communications in Computer and Information Science, pp. 26–33, Springer, Berlin, Germany.
- [12] W. Skarka, "Reducing the Energy Consumption of Electric Vehicles," in *Transdisciplinary Lifecycle Analysis of Systems, Proceedings of the 22nd ISPE-Inc International Conference on Concurrent Engineering Location: Delft Univ Technol, Delft, Netherlands, July, 2015*, R. Curran, N. Wognum, M. Borsato et al., Eds., vol. 2 of *Advances in Transdisciplinary Engineering*, pp. 500–509, IOS Press, Amsterdam, Netherlands, 2015.
- [13] Shell, Shell Eco-marathon, referenced at: <https://www.shell.com/energy-and-innovation/shell-ecomarathon.html>.
- [14] TASS International, *PreScan: Simulation of ADAS and Active Safety*, 2015, <https://www.tassinternational.com/prescan>.
- [15] A. Jałowicki and W. Skarka, "Modeling in Ultra-Efficient Vehicle Design. Proceedings of the 23rd ISPE Inc. International Conference on Transdisciplinary Engineering Location: Fed Univ Technol, Curitiba, Brazil October, 2016," in *Transdisciplinary Engineering: Crossing Boundaries*, M. Borsato, N. Wognum, M. Peruzzini et al., Eds., vol. 4 of *Advances in Transdisciplinary Engineering*, pp. 999–1008, 2016.
- [16] W. Skarka and M. Wąsik, "Aerodynamic Features Optimization of Front Wheels Surroundings for Energy Efficient Car. Proceedings of the 23rd ISPE Inc. International Conference on Transdisciplinary Engineering, Fed Univ Technol, Curitiba, Brazil, October, 2016," in *Transdisciplinary Engineering: Crossing Boundaries*, M. Borsato, N. Wognum, M. Peruzzini et al., Eds., vol. 4 of *Advances in Transdisciplinary Engineering*, pp. 483–492, 2016.
- [17] LTD Hokuyo Automatic Co. Scanning laser range finder utm-30lx-ew specification. 2012.
- [18] K. Cichoński and W. Skarka, "Innovative Control System for High Efficiency Electric Urban Vehicle. Proceedings of the 15th International Conference on Transport Systems Telematics (TST): Wrocław University of Science and Technology, Wrocław, Poland, April, 2015," in *Mikulski, J.*, OF. TOOLS, Ed., vol. 531 of *Communications in Computer and Information Science*, pp. 121–130, 2015.
- [19] S. A. Rodríguez Flórez, V. Frémont, P. Bonnifait, and V. Cherfaoui, "Multi-modal object detection and localization for high integrity driving assistance," *Machine Vision and Applications*, vol. 25, no. 3, pp. 583–598, 2014.

