INTRODUCTION

Autonomous functionalities of cars would be more common in future. Advance driver assistance systems (ADAS), like Advanced Emergency Braking System (AEBS), has become a standard in some vehicles. This new solution needs to be tested and validated on roads, but before that would happen, some tests should be provided to check if such a system may be used on the roads. Computer simulation allows to identify some conditions in early planning of algorithm. Simulations can be also used to verify and validate the system in constant conditions.

Computer simulations can be used thru full development of the ADAS system. The problem is – how to test a system? Environment allows to use specify road conditions and repeat them every time, but this conditions need to be identified earlier.

The authors propose to use some tests from real competition which took a place in 2007 – 3rd edition of DARPA Grand Challenge known as DARPA Urban Challenge (DUC) [1].

1. OBJECTIVE

Most of the ADAS systems are checked basing on the ISO regulations [2-5] and other test protocols like eVALUE [6], Euro NCAP [7], NHTSA [8]. Most of them are a simple test of one case of system behaviour. Such scenarios are not usually found on the roads. Real life scenarios are more complex. All the ADA Systems needs more tests made by the developers.

Objective of this paper is to propose tests based on the real life situations that had place during DUC with autonomous vehicles. Three main functionalities are presented with scenario for left turn, right turn (merge), and vehicle separation.

After tests for each of functionality alone, the combined scenario has been created. Scenario takes place on the map from DUC represented in PreScan environment [9].

2. REQUIREMENTS

Scenarios should cover the situations described in technical documentation of DUC requirements. Information about environment would be gather by a generalized sensor model – Actor Information Receiver (AIR). This type of sensor delivers information about distance, angle and velocity of the object.

During simulation the model of Audi A8 with simply dynamics and PathFollower [9] as the driver has been used. This combination allows to steer the velocity with throttle and brakes. The controller takes care about steering angle of steering wheel as well. Other Actors during simulation are using only kinematics and moves with defined velocity all the time.

3. SCENARIOS

Scenarios are based on the technical documentation of the DUC [10]. Three scenarios for main functionalities have been chosen. The set of tests is composed from manoeuvres: vehicle separation left turn, merge to traffic (right turn).

Vehicle separation requires keeping safe distance between host car and target car ahead. It is sum of the minimal safe distance and additional distance related to the host car velocity.

$$\text{minimal safety distance} = 2[m] + \frac{\text{vehicle length}[m]}{10[m]} \times \text{velocity}[\frac{\text{m}}{s}]$$

This functionality is also combined with Adaptive Cruise Control (ACC), because separation also needs to control the velocity.

Left Turn on the crossings needs to monitor environment and trigger safe manoeuvre. Such a ride is composed of stop before turn, estimation of distances and times to approaching vehicles, start moving when it is possible. The possible turn is, when time gap between cars on opposite lane is bigger than 10s.

Right Turn (also known as the merge to traffic) is combined of stopping the vehicle, monitoring environment and checking distances and times to objects, starting the manoeuvre. In this scenario, also time gap of minimum 10s is required.
4. CONTROLLER

Main controller is composed of 3 main functionalities: Adaptive Cruise Control, Left Turn and Merge. ACC controls Host car velocity all the time, when trajectory turns on the crossing, algorithm for turning left or right is activated.

4.1. Adaptive Cruise Control

First controller needed to adapt velocity of host car to the traffic condition is Adaptive Cruise Control (ACC). Such a system adjusts speed of the car to keep the safe distance between following vehicle and accelerated/decelerated if it is needed. The “safe distance” is the minimal safety distance described in DUC criteria. But for purpose of the aDRIVE project it was modified to minimal value of 1m (Figure 1).

The algorithm (Figure 4) takes as the inputs information about target and Host vehicle state. Based on the inputs values Queuing vehicle is checked if it is on the same lane as the host car. Two states can be defined here, the vehicle is on the straight road or on the bend.

If the Vehicle is on the straight road, the target car needs to be near distance Y =0 (+/- 1.5m).

If the host car is on the bend, radiuses of the curvatures of host and target need to be the same (+/- 1.5m).

If cars are on the same lane, Time To Collision (TTC) is calculated:

\[ \text{TTC} = \frac{\text{Range}}{V_t - V_s} \]

Range – Range to the target

V_t – Target Velocity

V_s – Host Velocity

Also the “Safe distance” is calculated

\[ \text{Safe distance} = 1 + 0.5 \times V_s \]

Next the desired velocity is set by controller to the minimal value of velocity set by user (can be connected with GPS map and Speed limit on the road) and traffic condition – velocity of the car ahead.

4.2. Left Turn

During travel it is often need to make the left turn manoeuvre. Vehicle would cross the opposite lane and needs enough space to make this move safely. During DUC such a safety distance has been described as 10s time gap between cars on the opposite lane (Figure 2).

The host vehicle should stop before crossing and start moving when the time gap will be big enough for the DUC criteria.

4.3. Merge

Merge manoeuvre also known as turn right is also needed like left turn to travel through the defined path. Car is moving forward, and it is stopped before road crossing. On the crossing vehicle would turn right, so it needs space defined as a 10s time gap and free lane (Figure 3).

The algorithm (Figure 5) is based on inputs from sensor: Target’s Range Theta Phi, Velocity and from state, velocity and Yaw rate. Also, to start the controller, information about turn is needed, it is predicted based on the defined path that vehicle would follow. The left Lane is checked if there is enough space to make the manoeuvre based on distances in X and Y direction and TTC in X direction to calculate time gap. If the manoeuvre can be done safety, the car accelerates to desired velocity.

5. SIMULATIONS

Every algorithm has been tested during simulation of a simple case before it has been used in complicated scenario. Simulations have been prepared in PreScan environment connected with Simulink [11] for controllers.

5.1. ACC

Scenario for ACC testing is a straight road with two cars. Host car with AC should follow the target car keeping the safety distance while target changes velocity. Target vehicle moves only with kinematics speed controller, while host car has a simple dynamics model. Start distance between cars CoG is 40m.
Bezpieczeństwo i ekologia

Fig. 4 ACC algorithm

Fig. 5 Left Turn Algorithm

Fig. 1 Merge algorithm
5.2. Left Turn

Left turn experiment has a 3-way crossing. Host car needs to turn left while from opposing side other cars came. There are different time gaps between approaching cars. Host car should select the one bigger than 10s.

5.3. Merge

Merge scenario has a 3-way crossing. Host car approaches to it and wait till time gap between cars on the target lane will be bigger than 10s. When there will be such a space car starts moving along path.

6. RESULTS

The following results are effect of the computer simulation of selected scenarios with implemented algorithms in Simulink model.

6.1. ACC

Host car has a speed limit set to 15 m/s while car ahead moving with variable velocity with range 0 – 13.8 m/s. Host car first drives with maximal allowed velocity, when it approaches to the target car it starts to adjust velocity to follow car ahead with keeping the safe distance. As the beginning distance between cars (40m between CoGs and 35m for sensor) is bigger than safety distance 15m/s (8.5m) car may accelerate. The entire time controller works with throttle to change velocity or stop when target vehicle brakes, and wait. When target starts moving again, the host car also starts following it until the full stop at the end.

In scenario of left turn car first ride 56m straight road to stop before crossing. It waits until the time gap will be big enough to make a safe manoeuvre. Such a gap is at 17s of simulation. Before this there was shorter gaps of 5s, 6s during them, car was waiting.

---

**Fig. 2 ACC Scenario**

**Fig. 4 Velocity of followed vehicle (green) and host vehicle (blue)**

**Fig. 3 Merge Scenario**

**Fig. 5 Distance to the followed vehicle (blue) and safety distance (green)**

**Fig. 6 Time gap between approaching vehicles**
7. COMPLEX SCENARIO

7.1. Scenario description

Complex scenario uses all described functionalities. Car has defined trajectory from point A to B. On path from A to 1, Host drives with ACC functionality behind target car. In point 1, the merge functionality works to detect tame gap between cars approaching from the left. Next turn right is free to go, so there was no need to work with merge functionality, and it doesn’t cause false alarms or unnecessary braking. In point 2 the left turn functionality looks for a space. At the end in point B is a parked vehicle, so the ACC causes braking and full stop in safe distance.

7.2. Simulation

Target vehicles approaches on the curved road segments, what can cause later detection of the objects. In the purpose of turning the indicator an additional controller has been used. It detects turns in defined trajectory and applies braking to velocity of 2.5m/s.

7.3. Results

Host car adjusts velocity to the car ahead from 0 to 32 second. From 32 to 53 second car is in merge mode. 53 to 65 Car drives again with ACC with maximal allowed velocity, 65 to 70 car turns right, 70 to 88 drives straight. From 88 to 93 the left turn functionality works. After this to 100 second car accelerates, after this it starts decelerating and fully stops at the end from 104s.

SUMMARY

Three scenarios based on the DUC technical requirements and one complex scenario have been created. Presented algorithms have been tested with simulations of the selected scenarios. Combination of functionalities: ACC, Left Turn and Merge give a simply
autonomous vehicle that can drive along defined path. All separated systems have pass the test. Also the combined scenario has been finished successfully.

In the future more systems required by DUC would be created including car behaviour on the 4-way stop crossing, U-turn and passing manoeuvre.

ACKNOWLEDGEMENTS

Presented work has been financed and supported by the project aDrive, a PBS3/B6/28/2015 project within Applied Research Program of the National Centre for Research and Development.

REFERENCES

2. ISO 15622:2002 – for Adaptive Cruise Control system (ACC)
3. ISO 22178:2009 – for Low Speed Following (LSF)
4. ISO/NP 22179 – for Full Speed Range Adaptive Cruise Control
5. ISO 17361:2007 for Lane Keep Assistance system
9. PreScan Manual

Funkcjonalności ADAS bazujące na zawodach DARPA Urban Challenge

Zaawansowane systemy wspomagające kierowcę (Advance Driver Assistance Systems - ADAS) odnajdują powszechne zastosowanie we współczesnych samochodach. Aktualne techniki komputerowe pozwalają na wykonywanie testów i weryfikacji nawet skomplikowanych algorytmów, użytych w złożonych systemach ADAS, w symulacjach czasu rzeczywistego. Systemy takie mogą być testowane zgodnie z istniejącymi normami i standardami (na przykład: ISO) oraz protokolami (na przykład zaproponowane przez NCAP lub NHTSA). Każdy z tych testów ma jednak pewne ograniczenia, które nie są możliwe do spełnienia w przypadku fizycznych testów, jak chociażby tylko jeden dodatkowy samochód jako monitorowany obiekt.

W 2007r. odbyła się trzecia edycja DARPA Grand Challenge, znana także jako DARPA Urban Challenge (DUC). Podczas zawodów, w pełni autonomiczne samochody miały za zadanie bezpiecznie ukończyć przejazd po realistycznym terenie miejskim z uwzględnieniem zarówno wszystkich przepisów ruchu drogowego (USA) jak i innych (fizycznie występujących w teście) uczestników ruchu. Opierając się o wymagania DUC wykonana została zaproponowana seria symulacji komputerowych. W pełni autonomiczny, wirtualny pojazd, kontrolowany był przez serię rozdzielonych, niezależnych i jednozadaniowych systemów ADAS dla łatwiejszej analizy zachowania samochodu we wczesnej fazie testów. Pomimo udanego ukończenia przejazdu testowego, algorytmy sterujące samochodem wymagają dalszej optymalizacji.

Authors:

Marcin Mirosław, Karol Golon, Tomasz Dziewoński, Dominik Jastrzębski – Warsaw University of Technology, Faculty of Power and Aeronautical Engineering.