CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Evaluation of Closed-Loop Resimulation Approach for Virtual Verification of Active Safety Functions

Master's thesis in Electronic Embedded System Design

Jingxiong Liu

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2018

Master's thesis 2018

Evaluation of Closed-Loop Resimulation Approach for Virtual Verification of Active Safety Functions

JINGXIONG LIU

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2018

Proof of Concept of Closed-Loop
Re-simulation Method in Verification
of Autonomous Vehicles
JINGXIONG LIU

© JINGXIONG LIU, 2018.

Supervisor: Siddhant Gupta, Volvo Car Corporation;
Lars Svensson, Chalmers University of Technology
Examiner: Per Larsson-Edefors, Chalmers University of Technology

Master's Thesis 2018
Department of Computer Science and Engineering
Electronic Embedded System Design
Chalmers University of Technology
SE-412 96 Gothenburg

Gothenburg, Sweden 2018

Jingxiong Liu
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

In this master thesis, the CLR (Closed-Loop Re-simulation) method with SPAS (Simulation Platform for Active Safety, a virtual testing platform built in Matlab Simulink) model and an Active Safety software: ASFG (Active Safety First Generation) in loop is evaluated.

The function tested in this thesis is brake function in ASFG. The brake model in the simulation is tuned according to the results obtained from field tests conducted on Volvo-manufactured vehicles. The testing of brake model is done in open-loop re-simulation, after which the SPAS model with refined brake model is put into CLR (Closed-Loop Re-simulation) to imitate real-life scenarios where the car will brake by itself in situations where human driver fails to react.

The scenarios tested first are simple CCR (Car-to-Car Rear) straight road scenarios, the simulation results are then compared with field test results. In the last part of the thesis, an AD (Autonomous Driving) route — a more complicated real-life road collected by Volvo is converted into virtual scenario for running the CLR method to test its feasibility in a more close-to-real-life application that is going to be implemented in the future.

The result is promising and possible improvements and add-ons are mentioned in future work part of the thesis.

Keywords: Embedded Software, Active Safety, Closed-Loop Re-simulation, Autonomous Vehicle, Virtual Simulation

Acknowledgements

I'd like to thank for the help and technical support from the Driver Assistance and Active Safety Group in Volvo Car Corporation, especially Melih Guldogus— my thesis partner from KTH, Siddhant Gupta— Volvo supervisor, Lars Svensson— Chalmers supervisor and Yury Tarakonov— the group manager who offered us the chance to do this master thesis.

Jingxiong Liu, Gothenburg, Jun 2018

List of Acronyms

AD SW—Automated Driving Software

AEB—Autonomous Emergency Braking

ASFG—Active Safety First Generation

CAE—Computer Aided Engineering

CCR—Car to Car Rear

CLR—Closed Loop Resimulation

CMbB—Collision Mitigation by Braking

ECU—Engine Control Unit

Euro NCAP—European New Car Assessment Programme

FCW—Front Collision Warning

OUT—Object Under Test

SPAS—Simulation Platform for Active Safety

SPI—Serial Peripheral Interface

SUT—System Under Test

Contents

1

Introduction

Volvo Car Corporation is one of the leaders in developing fully autonomous cars. The Active Safety Department in Volvo focuses on both development and proving of the safety of embedded software developed for autonomous driving.

Autonomous driving vehicles must handle all possible traffic situations, which requires large amounts of data. In the context of system testing, to save time and money and prevent long-field testing whenever there is a change in autonomous driving software, it is impractical to use traditional ways [1] like open-loop re-simulation with huge amounts of driving logs. In Volvo huge amount of the work was focused on open-loop testing, whereas the more practical closed-loop re-simulation (CLR) method [2] is under evaluation.

The idea of CLR is to let vehicle model, sensors and Active Safety/Autonomous Driving software (AD SW) run in a closed loop. This closed-loop model better represents future autonomous cars, with various sensors installed on the car detecting road condition and feeding back information for ECU (Engine Control Unit) embedded in cars to process at the same time, to indicate future behavior of cars. If the CLR method is proved trustworthy, then we can make the validation of Active Safety/AD SW much faster by replacing most open-loop and real field tests, and be one step closer to the realization of fully autonomous driving.

1.1 Aim

In this master thesis, the CLR method with Simulation Platform for Active Safety (SPAS) model and an Active Safety software in loop will be evaluated.

The purpose of testing this CLR method is to find out its feasibility in testing other Active Safety functions and AD software in the near future.

1.2 Scope

The scope of the thesis is vast considering the fact that the discrepancies between virtually simulated data and physical data from test track can arise due to various models used in the simulation environment.

This thesis project was divided into two parts, the model testing and validation part is the topic of my thesis partner Melih Guldogus in his thesis *Proof of Concept of Closed Loop Re-Simulation (CLR) Methods in Verification of Autonomous Vehicles*[18]. In the mean time, while I mainly focus on the realization of CLR tests for the evaluation of active safety functions.

Since simulation will not possibly cover every case in real life, the evaluation of CLR methods are based on certain user cases, i.e., specific simulation scenarios (e.g., braking with static target in front of the host car with initial speed ranged from 30 km/h to 80 km/h). After running a large amount of simulations with corresponding logs, we will conclude with certain confidence (e.g., how close are the braking distances in simulation compared with field test logs) that future tests on AD software regarding braking can be replaced with virtual simulation method like CLR instead of field tests.

In order to control the amount of variables so as to narrow down the scope of this master thesis (which is easier to pinpoint the causes of discrepancies between results), we will leave aside the sensor system and focus on investigating core vehicle models [1] and their braking performance when testing CLR.

Considering the time frame for the thesis work, the evaluation of the said CLR method will be focused on the user-cases arising out of EUNCAP Cases for testing of Car-to-car rear end (CCR)collisions. In figure 1.1 [22], one of the CCR test cases is demonstrated. There is a possibility that even though the model is perfect enough, the re-simulation result in closed-loop does not seem so, due to the software.

---

[1] core vehicle means car models with only basic mechanic models like braking and steering models, whereas sensor models and ECU are excluded

Figure 1.1: Euro NCAP CCR example.

1.3 Limitations

The work mainly focuses on the *evaluation* of CLR method. *Improvement* on CLR method (e.g. changing core vehicle models to increase correlation or modify functions in Active Safety software) will not be involved unless realization and evaluation of CLR are well completed beforehand.

2

Background

To help readers better understand the work conducted in this thesis project, this chapter aims to outline several fundamental concepts that are used in the thesis.

2.1 Simulation Platform of Active Safety (SPAS)

SPAS is the virtual simulation platform developed by the Active Safety CAE department at Volvo. SPAS, as of today is built for Matlab and Simulink environment and is executable on Windows and Linux environment. The simulation platform is used for the early verification of Active Safety functions both in terms of logic and performance testing. The environment is continuously developed in the agile framework for more efficient and reproducible virtual testing.

SPAS can be divided into mainly two parts: the core vehicle part and the software part. Core vehicle part models every mechanical component of a vehicle which are connected together to build a virtual vehicle depicting a physical vehicle in a virtual environment. The models can vary on their fidelity. For example, a radar model can be subjected to zero environmental noise or a brake model can be modeled without any effect of pressure rise, such as a radar model with no environmental noise or a brake pedal that reacts instantly. With continuous development and improvement, these models are improved on their fidelity with more mechanical and physical details to make the virtual tests comparable to real field test.

The second part is the software part, where all requests coming from SPAS core model (like braking request) are fed into Active Safety/AD SW software for processing. With sensor model add-ons that can detect barriers, the car can perform active response with little or no driver interventions, which is the idea of a future AD car.

The purpose of developing SPAS is to replace most field tests to save time and money, and also make the delivery of Active Safety/AD software much more faster.

2.2 Scenario

Scenario can mean two different things in this thesis: one is real-life road scenario and the other one is the virtual scenario built and loaded in SPAS.

2.2.1 Real-life scenario

In this thesis, the word “scenario” indicates traffic scenarios. When a car is driving on a road, the environmental surroundings around the host vehicle and the vehicle itself constitute the real-life traffic scenario. Major environmental elements like road condition, curvature of roads and objects around the host car are the main indicators for human drivers to decide what his/her next movement should be. Minor environmental elements include parameters like humidity, visibility and temperature, which can also be important to be taken care of in some certain situation.

In Volvo car corporation, huge amounts of field tests are conducted to test the performance of Volvo vehicles: some are collected on test tracks, e.g., car performance on slippery road and various active/passive functions; some are collected during road driving, e.g. driving a test vehicle around Spain to test sensor performance.

In order to run certain field test, the degree of restoration of real-life scenarios from traffic to test track will tell how much car manufacturers can trust the test results. For example, statistics show that car-to-car rear collision is one of the most frequent accidents in city driving [14]. If car manufacturers want to test their auto brake controller on test tracks, they have to make sure the test vehicle has the appropriate mass, tyre type, friction factor of roads so that the restored scenario can be related to the scenarios where accidents happened.

2.2.2 Virtual scenario

Although field tests are very important and necessary to test the performance and safety of new cars before they are delivered to the market, they are expensive and time-consuming. The test logs from test vehicles being collected from driving on public roads require a considerable effort and certain list of test parameters to be fulfilled. Moreover, vast amount of test data needs to be recorded to reach probabilistic trustworthiness. Virtual test can facilitate the testing of Active Safety/AD software as a good complement to field tests.

With the help of Computer Aided Engineering (CAE), virtual simulation is feasible and efficient if car components are well-modeled in the simulation platform. The scenario restored in the simulation platform is then called “virtual scenario”.

Virtual simulation will greatly facilitate the validation of both car components and AD software in various scenarios. Testers do not need to do large amounts of time-consuming and expensive field tests; all they need is a well-built virtual model and in a simulation platform where scenario can be restored in a good way.

However, there are also some disadvantages of virtual testing. Firstly, a well-built virtual model takes time to develop. For example, the brake controller in SPAS used today underwent heavy tuning to model physical behaviour. Secondly, it is still hard to determine how much percentage of field tests can be replaced with virtual tests: where should companies place the balance point between quality (field tests) and efficiency (virtual tests).

2.3 Validation

To make the virtual simulation of AD cars trustworthy, there are three categories of validation needed to be conducted — core-vehicle validation, sensor validation and AD software validation. In this thesis an ideal radar model is used, so more focus could be put on core-vehicle and AD software validation.

2.3.1 Core-vehicle validation

Core-vehicle validation refers to the validation of mechanical plant models and controller models in the virtual car, e.g., brake controller model and driver model.

Testers using only parts of field test logs (e.g., only extracting the braking request signal sent to ECU), use this as the input for simulation, and then compare the output between simulation and field test logs(e.g., comparing deceleration performance between simulation output and field test given the same input).

After testing several logs and tuning, the model under testing can be rated with a confidence level (e.g., more than 95% of closeness in braking distance between simulation and field tests). More details of core-vehicle validation will be explained in Chapter 4 — Vehicle model validation with SPAS.

2.3.2 Active Safety software validation

The validation of Active Safety software in the simulation platform is very similar to the validation of core-vehicle. The only difference is the input and its corresponding output. As shown in figure 2.1, when validating an Active Safety software, the input is still part of the field test logs. Instead of using the ECU input, sensor information is extracted and used as the input for Active Safety software. Then there will be comparison between the outputs of simulation and field test result.

Unexpected results (such as a collision when testing braking function software) will then be reported to the software team, the field testing will not commence until the

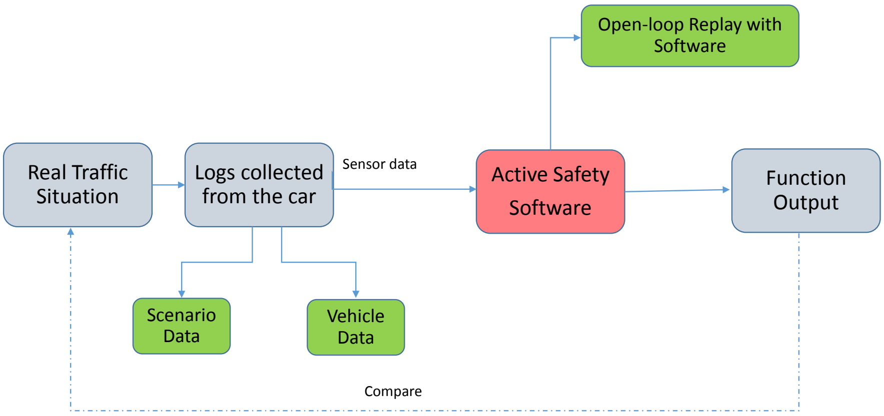


Figure 2.1: Validation of Active Safety software concept.

unexpected results caused by certain Active Safety software functions are fixed. In this way it will save a lot of time and trouble between the time-consuming back-and-forth procedure between software development team and field test team.

More detailed discussion about Active Safety software validation will be provided in Chapter 3 — Open-loop Replay with Active Safety software.

3

Open-loop Replay with Active Safety software

This chapter is a briefing of our evaluation of Active Safety software ASFG (Active Safety First Generation) v1 and v2. For more details please refer to the second chapter of my partner Melih Guldogus's master thesis [19].

3.1 Theory

In order to see the difference between different versions of Active Safety software, same field test logs are used as input into ASFG v1 and v2 respectively for simulation.

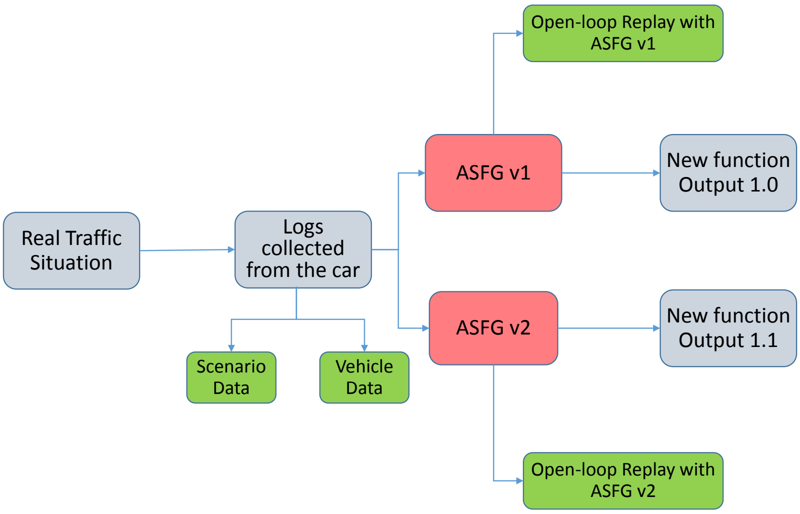


Figure 3.1: Comparison of ASFG software performance between different versions.

The purpose of this step is to see improvement, e.g., in braking functions. To better illustrate this, consider a scenario where there is a pedestrian standing 100 meters in front of a moving car. When subjecting the logs containing the sensor information into Active Safety software, the correct behavior of the Active Safety software should tell the car to start braking. Imperfect braking functions in Active Safety software may result in the car not braking enough or not braking at all.

3.1.1 Software development for active safety

Whenever there is a test failure (e.g., when testing braking functions in Active Safety software, there is a crash when there shouldn't be) reported using the current software version, the field-test team logs this failure and then report this to the software development team who is in charge of the specific function that may cause the test failure. After analyzing the test failure and debugging the function, the software team will deliver the updated version of software for field-testing again.

There is a possibility that even if the old function is fixed to avoid one specific test failure, new failures will arise using the updated version of software. The field-test team and software development team will hence go back and forth by exchanging their logs, analyzing problems and fixing functions for a more perfect Active Safety software.

The communication between field-testing team and software development team takes time, and it consumes even more time for field-testing of the updated software, let alone the debugging time for software development team.

Considering the above reasons, it is believed that virtual testing is an efficient way to facilitate the progress between field-testing team and software development team. For example, if the updated software should fail in one of the virtual tests, the malfunction may be diagnosed and fixed without resorting to expensive and time-consuming field tests.

3.1.2 Sensors on AD cars

Sensors are indispensable parts of an AD car; they perceive and reconstruct an environment that AD SW can comprehend and react upon. The AD software can do nothing without environmental information gathered by various sensors installed on an AD car. Some sensors like radars can detect the distance and speed of target, others like a camera-based sensor can tell whether the target detected is a human or an animal. In future AD cars, there will be more sensors installed, like sensors that can detect humidity and temperature in the air; with more sensors the reconstructed road scenario will be closer to real life.

The modelling of sensors is complicated when various kinds of noise sources are taken into consideration. Also, the shape, distance, surface texture and orientation of field of view can affect the detection of sensors in various ways. More sensors mean more information flow for the active safety software to process, for the vehicle under test in Volvo, there are already camera and radar-based sensors installed on cars.

However sensors are not the focus of this thesis, therefore they are not going to be discussed in detail.

3.2 Method

There are mainly two stages in re-simulation with ASFG as shown in figure 3.2. The first stage is composed with three phases, which contains camera detection information, radar detection information, and sensor-fusion information. At the end of stage 1 an Serial Peripheral Interface (SPI) interface is created so that the ASFG can recognize and process the information sent from the sensor fusion part. The second stage is the re-simulation part, which processes the fusion data sent from the first stage by ASFG. After processing the data, ASFG will send commands indicating the behavior of the vehicle in the very next moment — e.g., sending braking request to ECU through car bus.

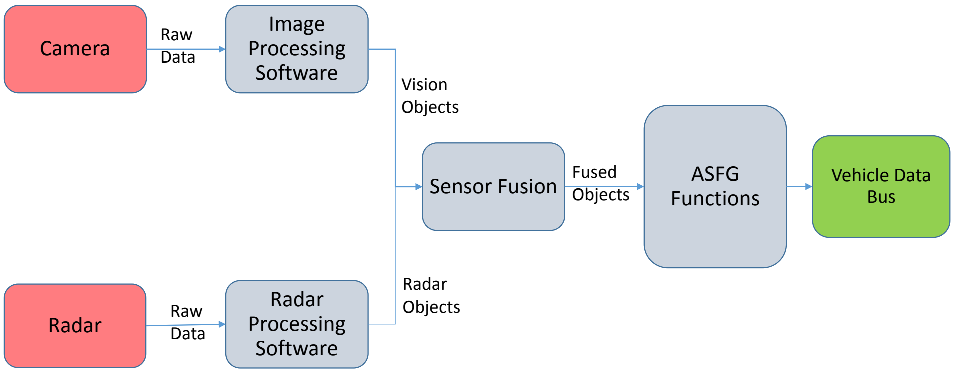


Figure 3.2: Re-simulation with ASFG and sensor input.

This process is called re-simulation because we are only taking image and sensor data from real logs, the output signals from ASFG are left out for comparison later. By comparing the output signals (like CMbB — Collision Mitigation by Braking enable signal or braking request) collected from data bus of real car and simulation outputs from virtual car bus, we can analyze the difference between virtual test and field test.

Also by changing different versions of Active Safety software (in our case they are ASFG v1 and v2) in the virtual simulation environment, we can see how much difference is among different versions of software in certain functions such as brake function.

By alternating between ASFG v1 and ASFG v2 in the software part while keeping sensor configuration the same in the sensor part, we put in the same field test logs (altogether with sensor data gathered with ASFG v1 built in cars) into these two versions of ASFG separately and see the output results. In doing so we can prevent difference in sensor configurations while focusing only on the difference in different versions of Active Safety software.

3.3 Result

Using existing logs at hand now, the outputs from these two versions of ASFG are exactly the same , which indicates there is not any updates in the braking function of the new software. With the conclusion kept in mind, ASFG v1 is used for all the CLR tests in the last two chapters of this thesis when testing CLR method.

4

Vehicle model validation with SPAS

This chapter explains how Simulation Platform for Active Safety (SPAS) model is validated: to make sure this virtual car is modelled close enough to a real Volvo car. This part is researched in more details by my parter Melih Guldogus in the third chapter of his thesis [20]. Various braking cases are being focused on for SPAS in open-loop without any sensor models or Active Safety software.

4.1 Theory

By subjecting open-loop field test logs with different initial speed and deceleration request into SPAS model, we evaluated the SPAS model by comparing the output (deceleration, speed over time and braking distance) from SPAS and driving logs from real cars.

The concept is shown below in figure 4.1. Field test logs are put into the SPAS model. Information like deceleration request and initial velocity is extracted from field test logs, and then put into SPAS — the virtual car. The output is the braking performance of the virtual car under the same scenario where testers tested real cars to collect the log.

By comparing output results like braking distance from virtual cars (SPAS model) and real cars, we can get ideas like how similar the real car and the virtual car model are.

This part of work is called core validation, which means the validation of virtual mechanical models of Volvo cars. The brake model (the controller and plant) is researched beforehand by my thesis partner M. Guldogus [20]; necessary changes are made to the brake controller to make it better for CLR tests. Once we are confident with core validation part of SPAS, then we can put the SPAS model in the loop in order to realize CLR.

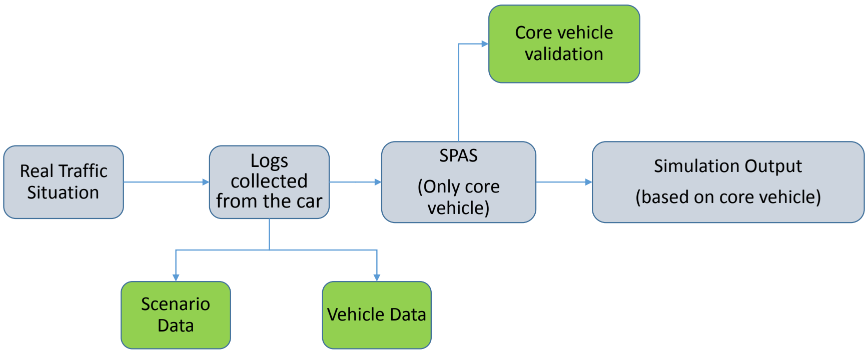


Figure 4.1: SPAS validation concept

4.2 Method

To generate convincing result, more than 100 braking logs are put into SPAS to run re-simulation. The input is braking request, which is either the output calculated from real traffic scenario by ASFG, or simply transformed from the force exerted on braking pedal by a human driver. Here it is used as a given input for open-loop test, the source where it comes from does not matter. The output is deceleration from SPAS model. Deceleration results are compared between real car logs and SPAS output. Also to make things more intuitive, velocity and braking distance are compared as well.

The scenarios used in CCR (Car to Car Rear) tests are Excel-sheet based. The road information in excel-based scenario is designed in a toolbox in Matlab named RoadMaker, in which users can define simple virtual roads like a 1000 m straight road or one section of a curvy road. Also, while making the scenario users can also define which kind of vehicle model (e.g., S90 or XC60) to be run on the virtual road.

The simulation output is then compared with the corresponding field log data (e.g., braking distances) to evaluate SPAS components concerned with braking. The input used here is braking request (or deceleration request), it is one of the outputs of braking functions in AD software. It is a non-negative integer ranging from 0 to 12, where 0 means no deceleration and 12 means the AD software expects the vehicle to have a deceleration of $-12\ \mathrm{m/s^2}$.

The simulation tests conducted [20] are using step requests as inputs. The simulation results are then compared with corresponding field test results.

4.3 Result

The results are shown in figures 4.2. These tests used field test logs from Volvo car S90, all with maximum deceleration request.

The braking performance of SPAS is found to be corresponding very well with the physical tests with a deceleration request of -12 m/s^2 (corresponds to maximum braking request value, which is 12). The SPAS Development team have been tweaking the SPAS braking model in open-loop tests with the maximum braking request. But when we tried to put in braking request lower than 12 the result is not satisfactory — no matter how much the deceleration request is, the deceleration always oscillates drastically between 12 and 0. Chassis model, driver model, brake model and their sub-components are investigated, and we found out that the modeling of the brake controller needs to be re-examined [20].

In order to solve this, the parameters of PI controller inside the braking model were tweaked along with the brake reaction time. We also tried to replace the SPAS chassis model with FMU (another thesis model), but FMU gave bad results so we stayed with SPAS chassis model.

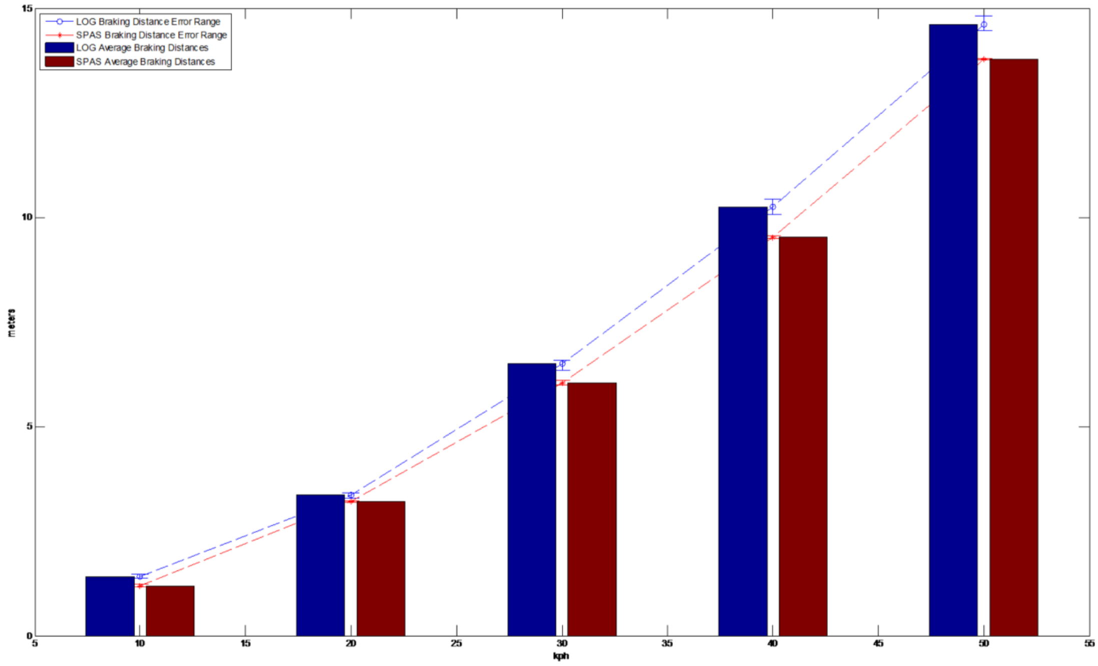


Figure 4.2: Average braking distance with maximum deceleration request[18]

The comparison of average braking distances is plotted in figure 4.2 with variance.

There are more than 150 logs tested, the blue bars indicate the average braking distance from field test, while the red bars indicate the braking distance from SPAS. The initial speed starts from 10, 20, 30, 40, 50 km/h, the average braking distance from SPAS simulation is smaller than the ones from field test logs and this is because the friction coefficient is a little bit higher than in real life. The more the initial speed, the larger the braking distance difference. The relative difference of average braking distance between SPAS simulation and field test data is within 5%.

After making sure the results are still good under maximum deceleration request, new field test logs from vehicle XC60 are tested as well, with more varied deceleration request [20].

As we are confident with the revised braking model now, next step is to see the braking performance in closed-loop re-simulation.

5

Closed-loop re-simulation for collision avoidance

5.1 Theory

After we are confident about the vehicle model and AD software, ASFG (Active Safety First Generation — the part containing sensor model and AD software in SPAS) and sensor models are activated in SPAS model to make it run in closed-loop mode. AD software and SPAS are connected to run in a closed loop re-simulation (CLR) environment . The purpose of testing CLR tells whether it is trustworthy to use CLR method as an indicator of car safety before field test. If the CLR is working very close to real field tests, we can replace most field tests with simulation tests on powerful computers to save time and money, also making the testing of AD software much faster.

The concept is shown in figure 5.1.

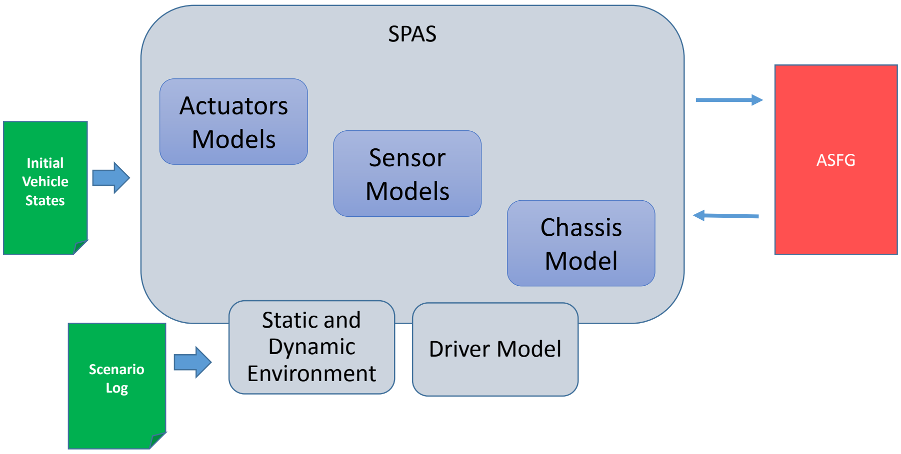


Figure 5.1: CLR method concept.

5.1.1 Euro NCAP

The test data in this chapter is obtained from Euro NCAP (European New Car Assessment Program) [17]. For twenty years, this program has accepted new cars from various car manufacturers in Europe and given safety ratings from test results.

There are some definitions to be clear with before the discussion of simulation results in this chapter [14]:

- Forward Collision Warning (FCW) – a signal sent automatically by the car alerting the driver about potential collision when the vehicle is detected too close to any target in front of it.

- Autonomous emergency braking (AEB) – when the braking force is not enough or none at all a certain time after the FCW signal, the car will brake automatically to decrease speed or brake to avoid potential collision.

5.1.2 Car-to-Car Rear collision tests

Car-to-car rear collision is one of the most frequent categories of car accidents. When drivers are not operating properly or distracted, this kind of accident is likely to happen.

In cities where vehicles tend to be crowded and often with relatively low speed, the CCR accidents consist over one quarter of all crashes [14]. While the injuries are not fatal most of the time, the whiplash injuries on drivers are very common.

The CCR accidents can also happen on open roads, when drivers are driving at relatively constant speed for a certain time and his attention is blunted over time. The driver may fail to notice the stopping or braking object in front of him sometimes.

For major car manufacturers, it is necessary for them to consider how to avoid nose-to-tail accidents. There is technology like collision warning system and auto-braking system. There are two protocols concerning the car safety stipulated by Euro NCAP. The AEB City system protocol [15] is for cars that drive at relatively low speed in cities, while the AEB Inter-Urban protocol [16] caters for higher speed when cars are not only driving inside cities. For AEB city systems, only CCRs scenario is tested [15]; while for AEB Inter-Urban systems, CCRs, CCRm and CCRb scenarios [16] are tested to score the safety of new cars. In this chapter there are simple closed-loop re-simulation tests first, i.e., CCRs, CCRb, CCRm scenarios, where a host car equipped with Active Safety software and sensors is driving towards a target car in front of it (s: static target, b: braking target, m: moving target).

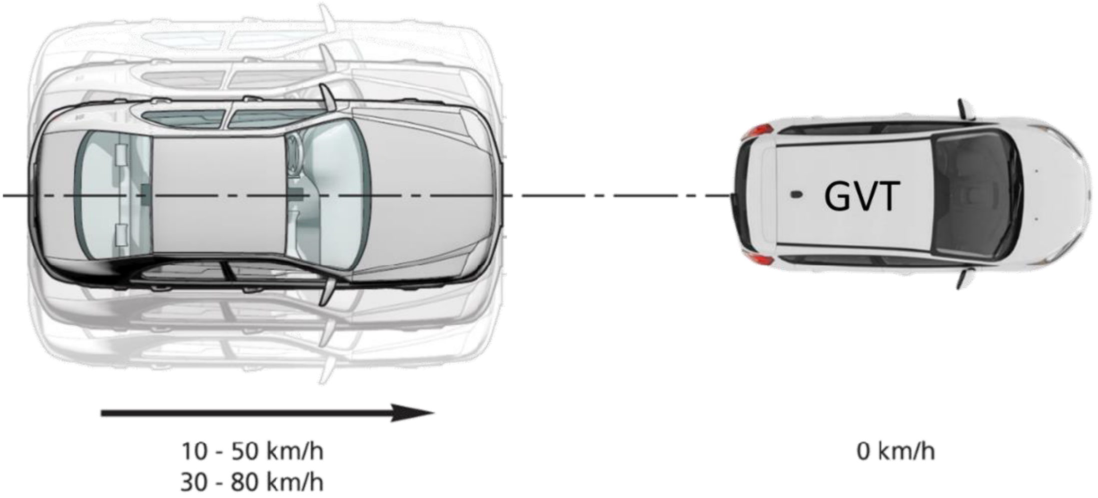


Figure 5.2: CCRs scenario Euro NCAP.

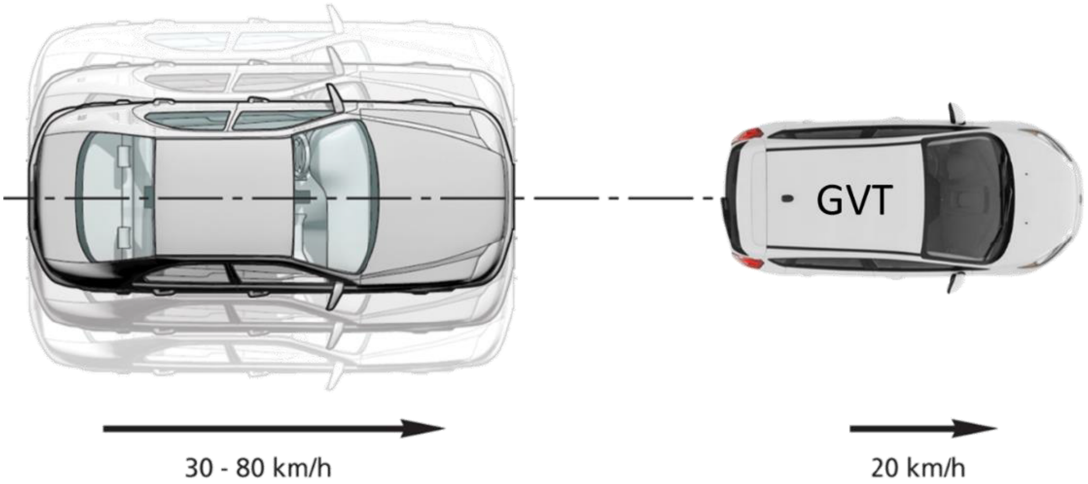


Figure 5.3: CCRm scenario in Euro NCAP.

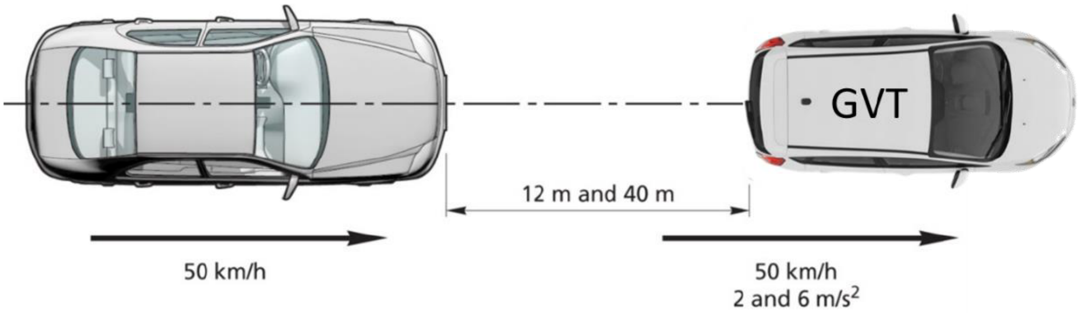


Figure 5.4: CCRb scenario in Euro NCAP.

5.1.3 Automatic Emergency Braking (AEB) Mechanism

In the CLR, the sensors installed on cars are able to detect the target within its effective range. There are two stages in braking with ASFG: First when the sensor detects something in front of the car, it sends a FCW (Forward Collision Warning) signal to the driver; the second stage is only activated when the driver does not give enough braking after the FCW flag, if that is the case, CMbB (Collition Mitigation by Braking) mechanism kicks in and ASFG will decide how the car should brake.

In SPAS model the sensor is modelled ideally and it can detect the object car in front of it in a certain range (e.g. 150 m). But in real cars the sensors will always experience noise caused by weather condition or simply different size and shape of the objects detected, therefore it is hard for the sensor to decide whether it detects a certain object around certain distance range. For example from the real logs we can see the relative distance (colored in blue) value is oscillating around 150 m (as shown in figure 5.11). The sensor detects something but since the reflecting signal is weak due to distance it confuses with noise until the car becomes closer and signals are strong.

5.2 Method

The scenarios used in CCR tests are Excel-sheet based with a set format so that SPAS can read it. The road information in excel-based scenario is designed manually according to the format mentioned above, in which users can define simple virtual roads like a 1000 m straight road or one section of a curvy road. Also, while making the scenario users can also define which kind of vehicle model (e.g., S90 or XC60) to be run on the virtual road.

For example, figure 5.5 shows one of the scenarios designed in BirdView (a SPAS tool to visualize simulation result in 2D). It is one section of a straight road, the red car is the host car and the blue car is acting as an either stationary or moving object for sensor detection. The x and y axles are coordinates of the map in meters. The radiating lines on the host car depicts the field angles of cameras and radars. The host car is driving to the left towards the target car to test AEB function. The red host car will receive the sensor signals reflected and Active Safety software will decide whether it needs to trigger any function based on the feedback signal.

After the road is defined properly according to the corresponding field-test logs, the road is loaded as a scenario into the simulation platform and ready for simulation. The output of the simulation will then be compared with the vehicle performance (in our case, braking performance evaluated by deceleration, velocity and braking distance) under specific scenario.

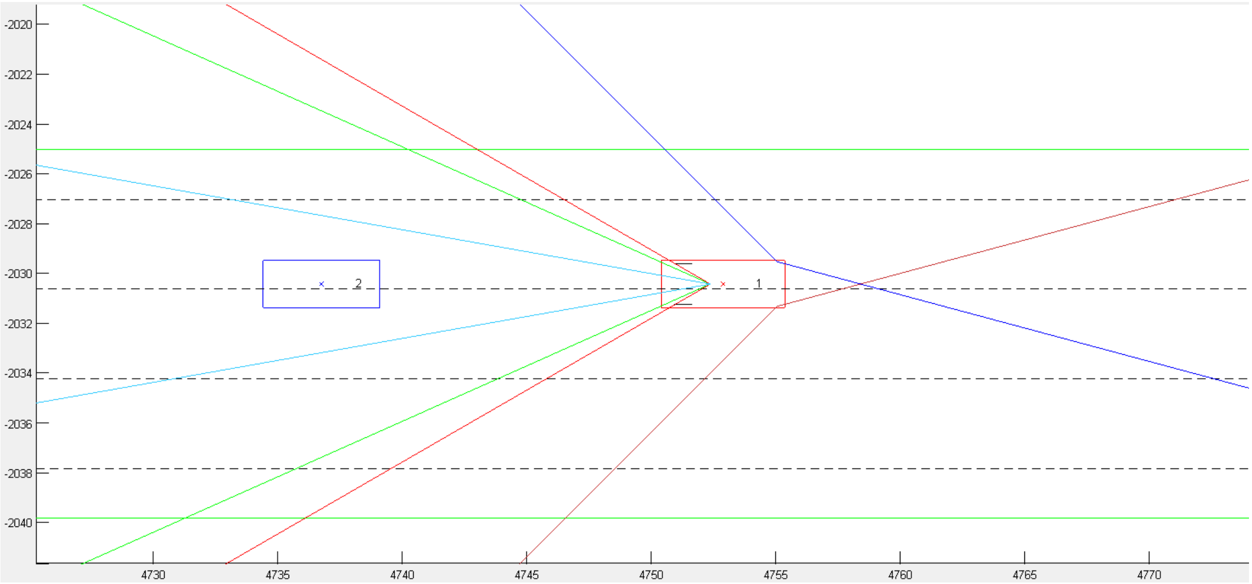


Figure 5.5: Virtual simulation shown in BirdView, car in red is the host car equipped with sensors and ASFG, car in blue is the target.

5.3 Results

To provide convincing results, more than 100 braking-with-ASFG logs were tested. There are 3 categories: CCRs (target car is static), CCRm (target car is moving in constant speed), CCRb (target car is decelerating with constant deceleration). Under each category there are logs with different initial host car speed. All the following figures plotted in this chapter are aligned by the FCW flag (red dot in each figure). The CMbB flag (green dot in each figure) comes a certain time after FCW flag.

There are four parameters to compare with between SPAS output and logs, i.e.: deceleration request, deceleration, velocity, braking distance (relative distance between host and target cars is used here). The situation is more complicated here because there are two sets of deceleration request — one from log and the other one decided by virtual sensor detection and ASFG. If this part generates good results, we will have a high confidence on real traffic scenario and CLR realization in the final step.

5.3.1 CCRs results

The example shown in this section are figures 5.6, 5.7, 5.8, representing the deceleration, velocity and relative distance respectively. In each figure the result from simulation is compared with field test logs. The figures are taken from one of the CCRs scenarios where the car type is S90, initial speed 50 km/h with a stationary target more than 200 meters away from it.

In figure 5.6 there are 4 curves plotted, deceleration and braking request from both simulation and field-test log. Also, the FCW flag is plotted as a red dot in the figure. After a certain time, the CMbB signal marked as a green is activated by AD software after a certain time because the driver model is removed from SPAS (this equals to the real-life scenario when driver is not braking the car) to test auto-braking. Right after the CMbB signal there is braking request sent by AD software and braking component of the car will act in time as well.

Something more is worth mentioning in figure 5.6: there is some acceleration when there is around 100 samples, that is due to the host car model in SPAS trying to stabilize at the set speed, just like real cars need some time to reach certain speed after it is started. At around 900 samples there is a small overshoot right after FCW signal, that is a small brake pulse to alert the driver that the host car is about to bump into target if there is no further intervention by the driver. If driver still does not brake after the FCW signal, CMbB function will then override the driver and brake by itself to avoid accident.

Deceleration request of field test data went to 12 (corresponding to -12 m/s^2) after full braking because the host car is still on with zero speed, while in SPAS it went to zero since the virtual vehicle is turned off after coming to a standstill. The oscillation after full stop in SPAS simulation is ignored by testers according to SPAS team, so no further time was spent on that part since it is not relevant to for the evaulation of braking distance.

Figure 5.7 shows the velocity results from both simulation and field test. The simulation result curve is colored in pink and the log data is plotted with light blue line. The slope of these two curves when the car decelerates is very close.

To be more intuitive, the relative distance between the host car and target car is shown in figure 5.8. The blue line represents the log data and the red line represents the simulation output. There are oscillations at the start of log data. This is because when the sensors are far away from the target the signals are more affected by noise. The default relative distance is set to 200 meters when there is no object detected. The signal gets stable when the relative distance is less than 100 meters and the two curves are almost the same.

The lowest point in figure 5.8 on its bottom right corner is the minimum relative distance between the target and host car. It is around 3 meters, which means that collision was successfully avoided.

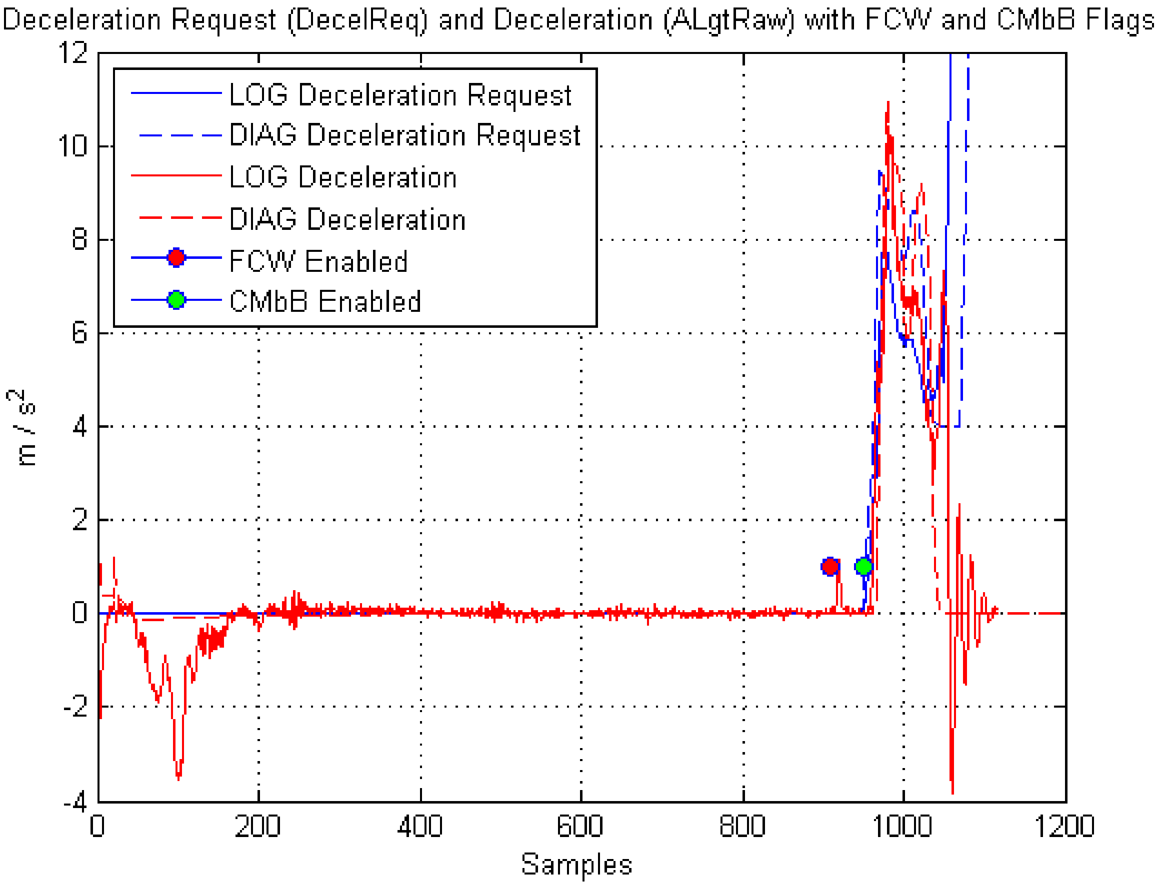


Figure 5.6: CCRs: Deceleration and Deceleration Request; time between two samples is 0.02 s; “LOG” is the data from field test, “DIAG” is the diagnostic output from simulation.

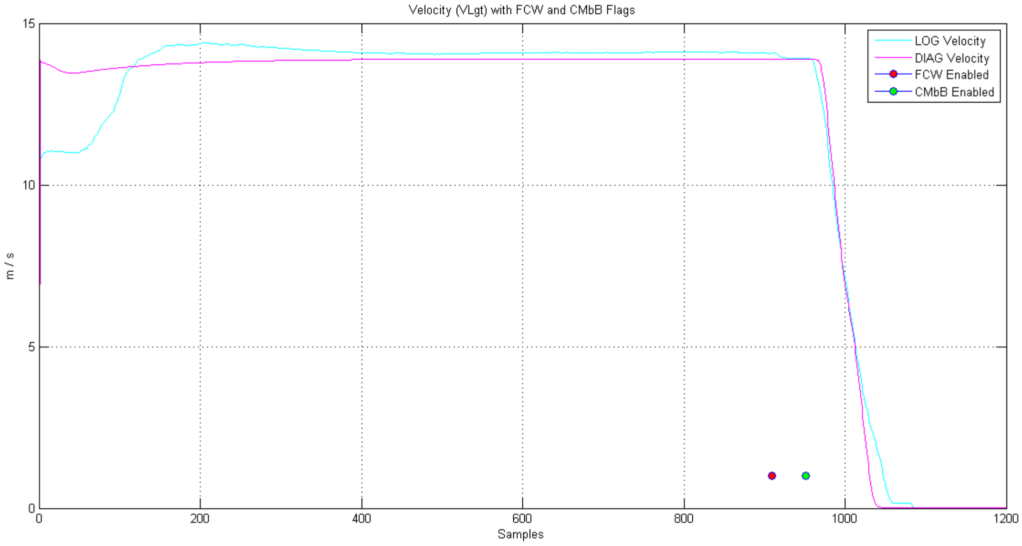


Figure 5.7: CCRs: Velocity; time between two samples is 0.02 s; “LOG” is the data from field test, “DIAG” is the diagnostic output from simulation.

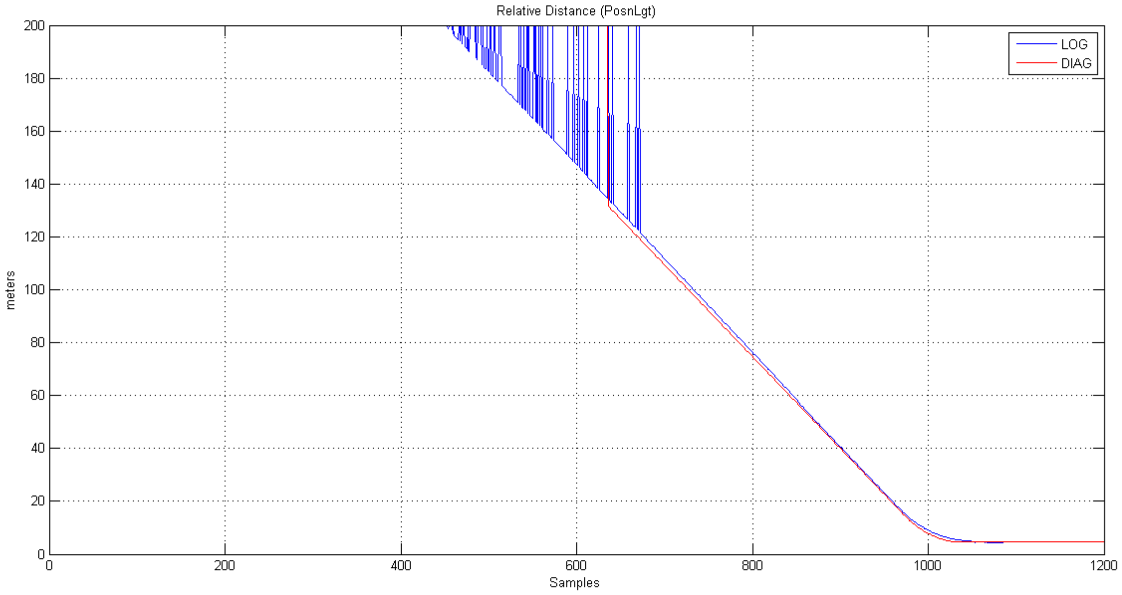


Figure 5.8: CCRs: Relative distance; time between two samples is 0.02 s; “LOG” is the data from field test, “DIAG” is the diagnostic output from simulation.

5.3.2 CCRm results

The following example is from a CCRm log with target car initial speed at 80 km/h.

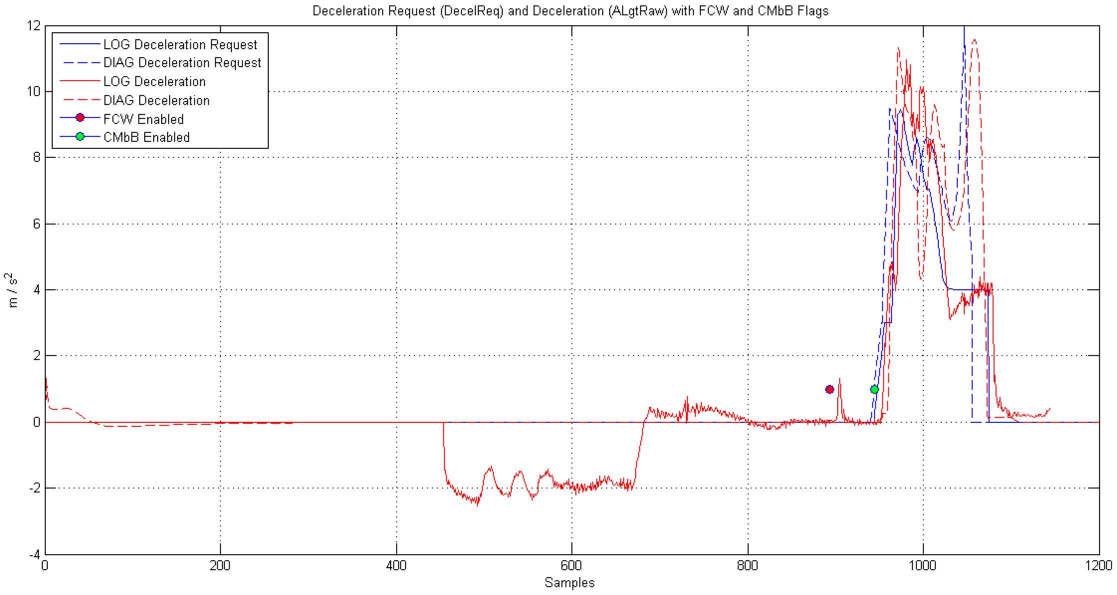


Figure 5.9: CCRm: Deceleration and Deceleration Request; time between two samples is 0.02 s; “LOG” is the data from field test, “DIAG” is the diagnostic output from ASFG.

The result shown in this section is tested with S90 model. The host car is driving with an initial speed of 80 km/h, the target car is moving at 20 km/h constant speed more than 200 meters away from the host car.

Similarly to the results shown in the CCRs section, figure 5.9 shows the braking request and deceleration information both from simulation and field test. The deviation around 600 samples is when the simulated car sped up to reach the same speed as the car in field test before deceleration. In figure 5.10 the velocity results are plotted, the slopes are pretty close. In figure 5.11 the relative distance between two vehicles are shown; again there is oscillation in field test logs due to environmental noise. The signal gets stable when the distance is less than 80 meters, and the two curves are almost the same.

The only significant difference in figure 5.11 is the diversion after the minimum distance point (bottom right on the figure). That is because in simulation there is no driver model, after the braking the car will stop while in field test the host car is maintaining a velocity less than 20 km/h following the target car. This can also be explained in the velocity figure.

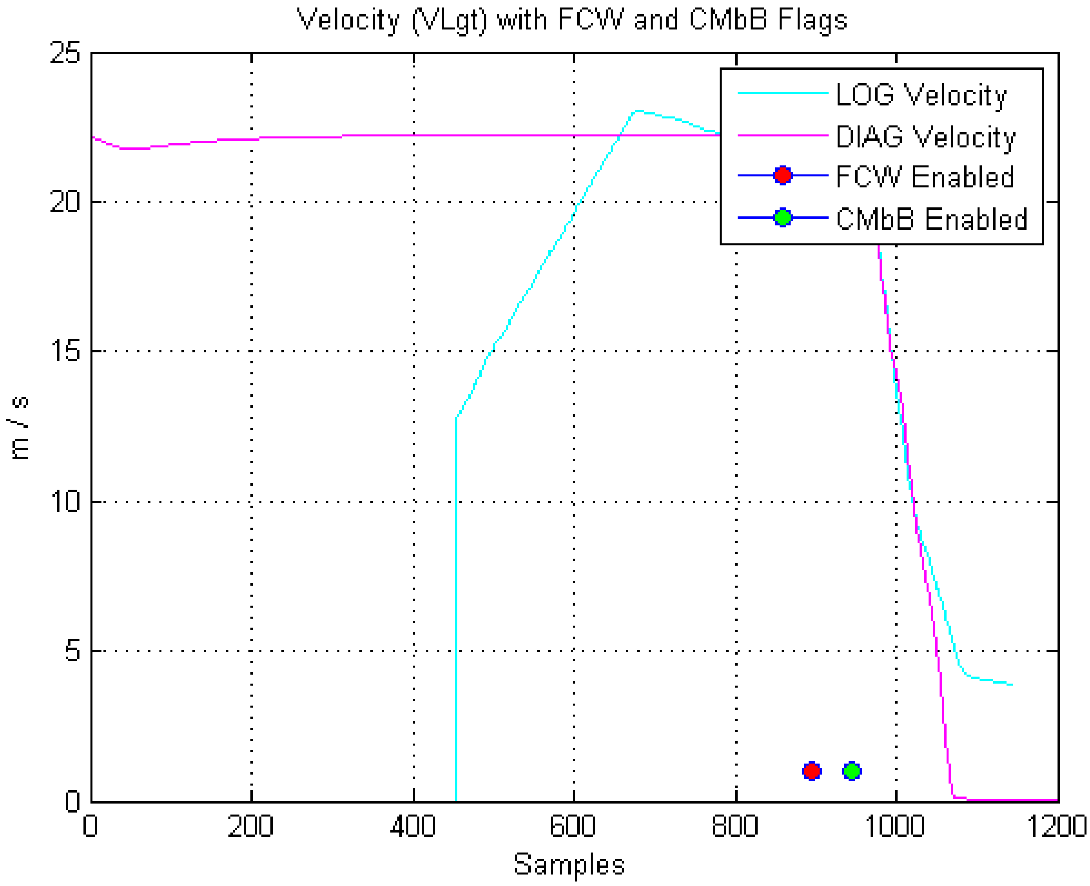


Figure 5.10: CCRm: Velocity; time between two samples is 0.02 s; *LOG* is the data from field test, *DIAG* is the diagnostic output from simulation.

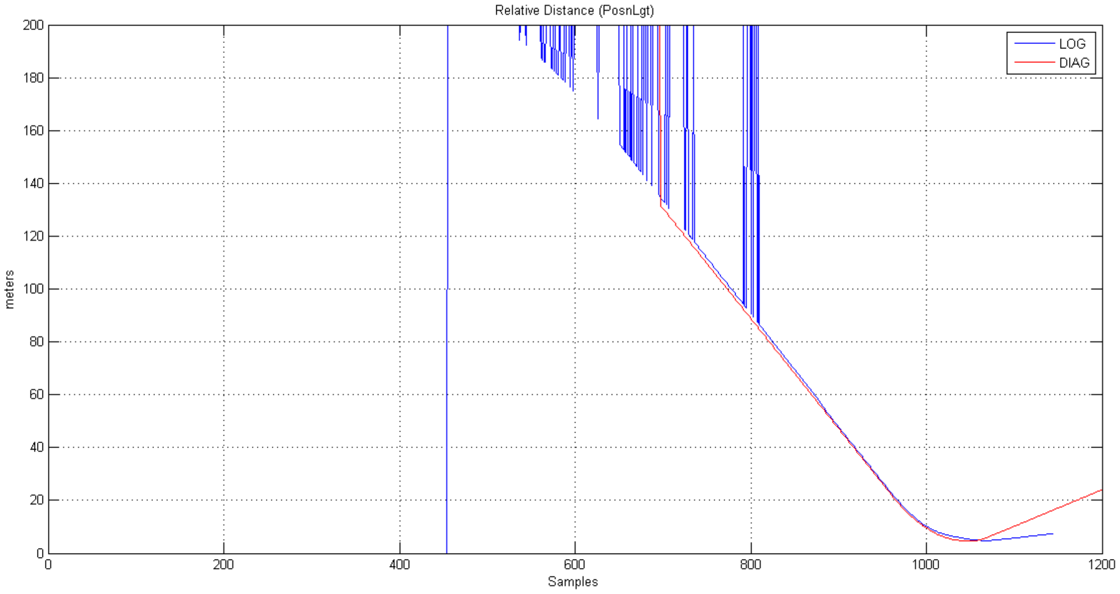


Figure 5.11: CCRm: Relative distance; time between two samples is 0.02 s; *LOG* is the data from field test, *DIAG* is the diagnostic output from simulation.

5.3.3 CCRb results

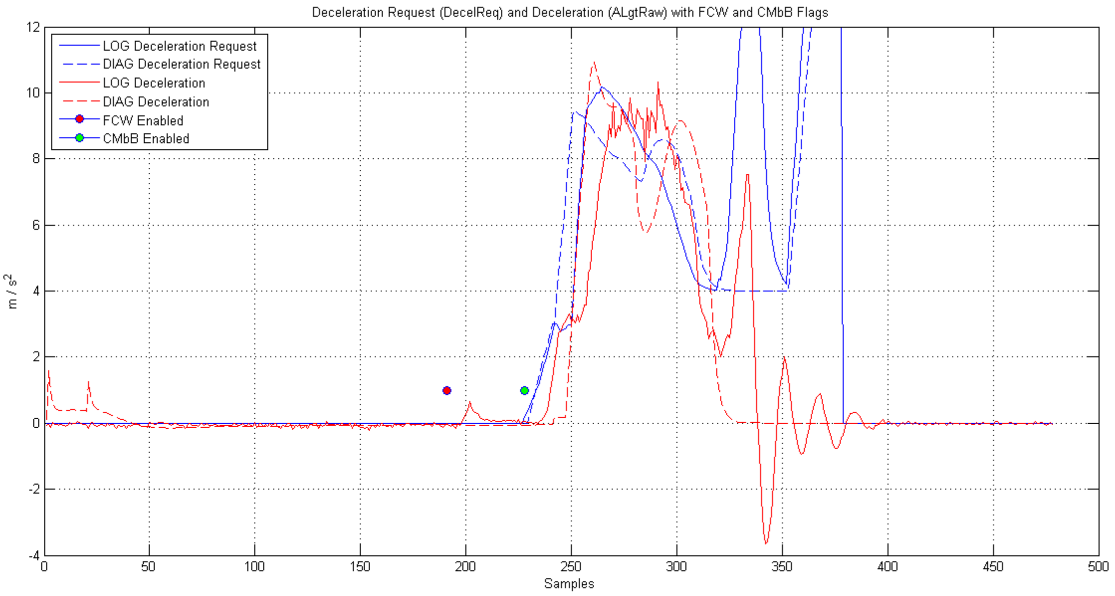


Figure 5.12: CCRb: Deceleration and Deceleration Requesty; time between two samples is 0.02 s; *LOG* is the data from field test, *DIAG* is the diagnostic output from simulation.

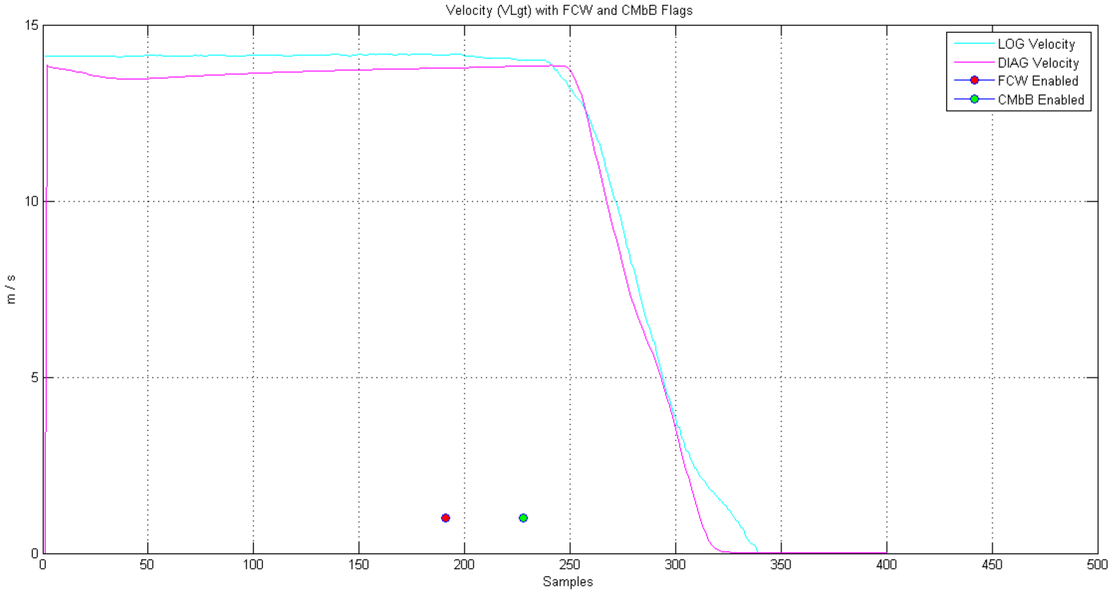


Figure 5.13: CCRb: Velocity; time between two samples is 0.02 s; *LOG* is the data from field test, *DIAG* is the diagnostic output from simulation.

The CCRb scenario is the most complicated one among all the CCR categories because the target is moving more unpredictably than either a static (CCRs) or

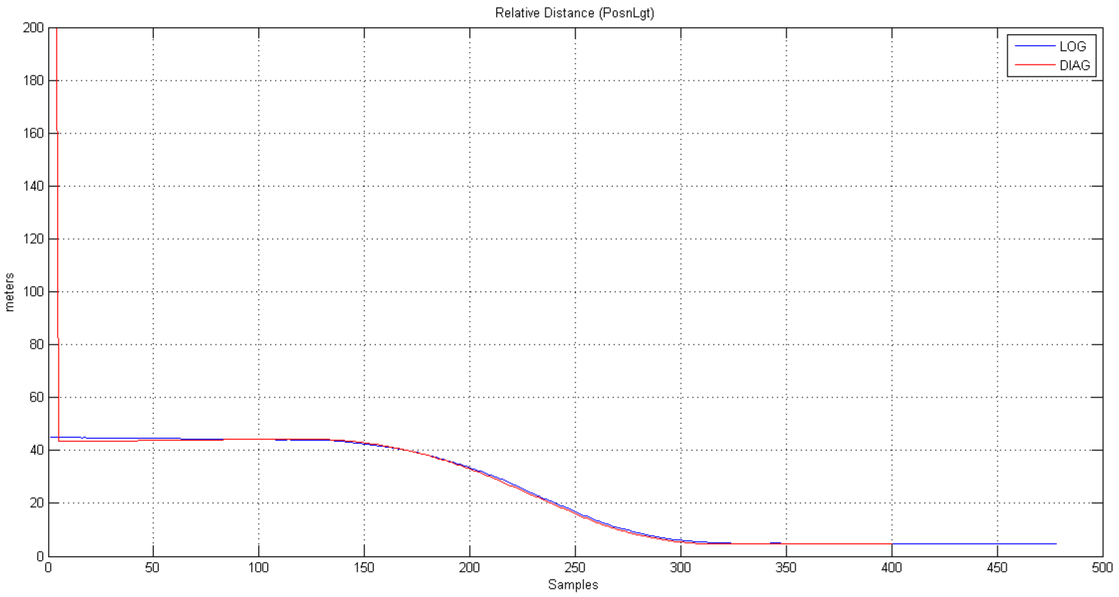


Figure 5.14: CCRb: Relative distance; time between two samples is 0.02 s; “LOG” is the data from field test, “DIAG” is the diagnostic output from simulation.

moving target at constant speed (CCRm). This is a good test for sensor algorithm and also AD braking functions because CCRb is closer to a real-life scenario.

The example shown in figure 5.12, 5.13, 5.14 is again tested with car model S90. In this case both the initial speed of host and target car have a speed of 50 km/h. The initial distance between two cars is 40 meters, while the target car is braking with 6 m/s^2.

The deceleration comparison and velocity comparison are shown in figure 5.12 and 5.13. One of the noticeable features is in figure 5.14, where there is no oscillation in the beginning part in the relative distance figure. That is because the starting distance is only 40 meters, far less than 100 meters, and the SNR (Signal-Noise-Ratio) is large enough for sensors to get stable signal. Again the simulation result is almost 100 percent simular to the field test regarding brake distance.

$$deviation = \frac{|real\,value - simulation\,value|}{real\,value} \tag{5.1}$$

The CLR model is constructed successfully in this section, and more than 100 CCR tests are simualated. Braking distance is the most representative output from simulations for evaluating braking function in CLR. In all simulations, the largest deviation of braking distance is within 5%, where CCRs has the best results, followed by CCRm and CCRb. The definition of deviation is listed on the above equation.

In the next chapter more complex scenario is going to be tested using CLR.

6

Closed-loop re-simulation for autonomous driving

In real life, the traffic situation is much more complicated than test-track scenarios: there will be more objects moving in more unpredictable ways. After the promising results we got from the CCR simulation, in this chapter more complicated scenarios obtained from Volvo road-driving logs will be tested using CLR method, so that we can have a more convincing and close-to-real-life result from CLR.

6.1 Theory

In this section, we looked into road driving logs collected by Volvo to build a more realistic scenario, where cars are running on real-world roads and information were collected as the car moves along. Previously the scenarios are easy to define, e.g., a straight road with only two cars with known initial speed in CCR tests; the scenario is simple and can be designed manually in Excel sheet for SPAS to load. This time, scenarios are not created by design engineers anymore; a Volvo-developed tool called ScenarioParser is used to extract map information from driving logs, and then to transform the extracted data into virtual scenarios. This tool can extract the information of all objects and roads information in Volvo expedition logs to create a realistic scenario. With some tweak in the extracted scenario manually, we can choose any of the targets moving on the map and designate the chosen one as an AD car by activating sensors and ASFG on it. Every car can be assigned to be the host car, objects and roads can be modified manually as well.

For example figure 6.1 shows one of the MATLAB-file-based scenarios extracted. There are as many as 38 objects running on 14 sections of roads. After the scenario is extracted successfully, we can run our modified AD car model on it and see if the target car runs safely on its own. In figure 6.2 car number one (colored in red) is activated as an AD car without a driver.

6.2 Method and Result

As mentioned above, we can manually delete any car objects or road sections to find only the interesting part (e.g., potential collision scenarios) during CLR.

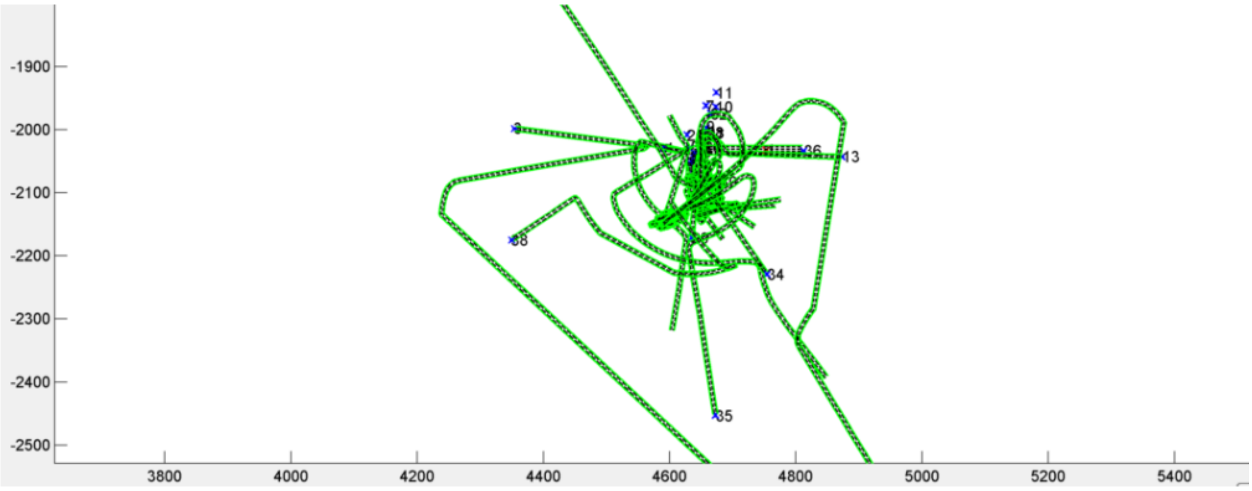


Figure 6.1: mat-file-based scenario, x and y axis are coordinates of map in meters, this is the visualization of a real world map rendered in BirdView. The numbers represents different vehicle objects on this map.

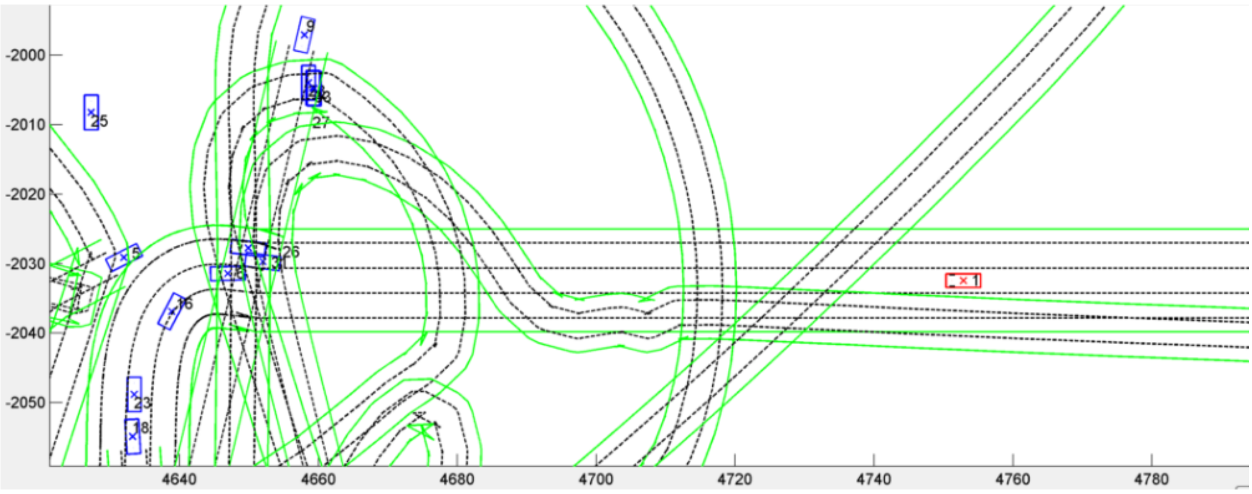


Figure 6.2: Replace one non-AD car with an AD car, the car in red is an AD car now with Active Safety functions. The x and y axis are coordinates of map in meters

Previously the user-defined scenarios are idealized with host cars and target car perfectly aligned and moving in the same direction; but active functions need to be tested in more close-to-real-life scenarios. In the extracted map , there is one critical scenario where we can test brake functions. For instance in figure 6.3, there are two cars (colored in blue) in the moving path of the red host car, THUS there is a potential collision in simulation. We can cut off rest of the roads and irrelevant targets to make simulation faster.

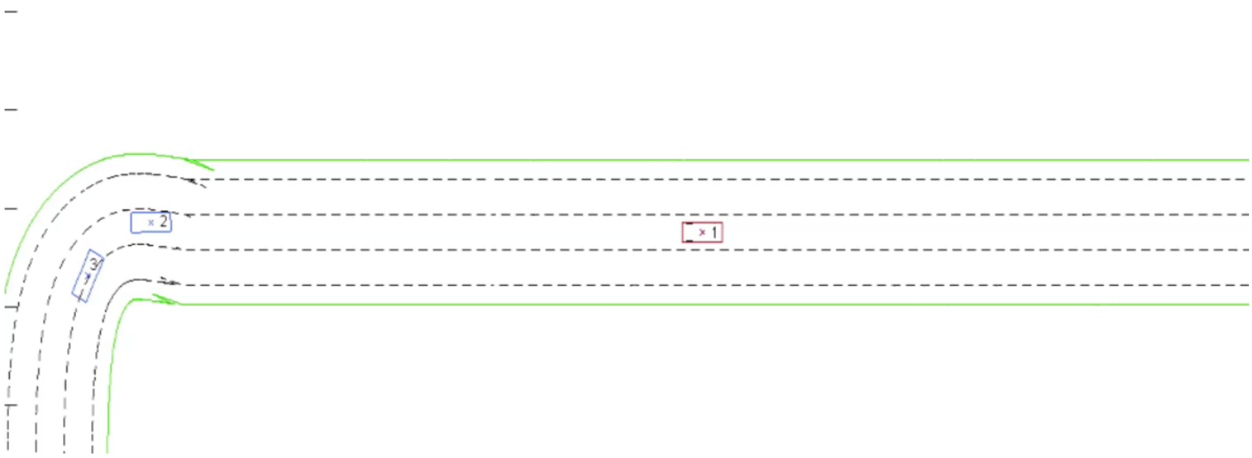


Figure 6.3: One interesting section of the whole scenario, the AD vehicle in red and traffic objects in blue extracted from the real log.

Figure 6.4: CLR test. In this figure one of the critical situation is found on the map, the curve right beneath the BirdView is the reaction of deceleration. The AD car is braking with the help of ASFG, there is deceleration on the host vehicle but it does not fully stop the car before the collision happens. This means the braking function, while working great under perfect CCR tests, is not performing as good in close-to-real-life scenarios. This is to be reported to the engineers developing ASFG and further tests will be done in the future to improve the braking function.

7

Conclusion

7.1 Conclusion from results

In Chapter 4 the brake model in SPAS is tuned to make SPAS brake controller act more like a brake controller in a real Volvo car. Based on the finely tuned brake controller in Chapter 5 CCR tests are run to verify CLR method. In Chapter 6 CLR method is further implemented on a virtual map. From the results we have a high confidence that most braking tests can be run with CLR method to give a convincing results. Based on the results, we have the highest confidence in braking distance of cars with ASFG installed, but less confident in velocity and least on deceleration.

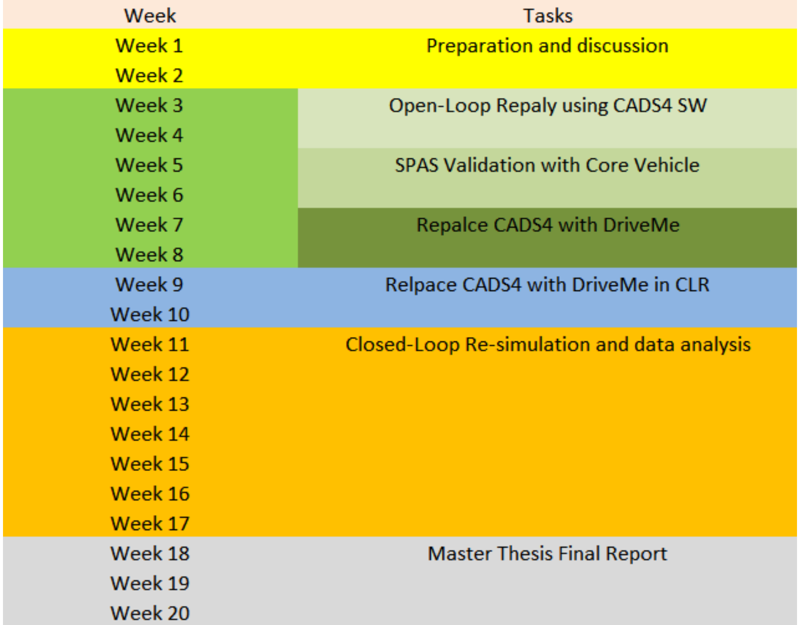
7.2 Limitations

The CLR tests are all based on braking tests. Steering and other car behaviors are not tested yet. Also, due to time limits there are not enough CLR tests on road-driving scenarios. As mentioned above, we have the highest confidence in braking distance of AD cars, but less confident in velocity and least on deceleration. But since only braking distance taken into account in safety evaluation, the result is acceptable.

Also, since simulation can not replicate the exact real-life scenario, it is never more convincing than a real road test. The purpose of CLR validation is to provide a faster and more economically efficient way to give a quick and preliminary feedback when there are updates in AD software on the presumption that the virtual car model (in this thesis it is SPAS) is of high fidelity compared to a real car. If there is a crash in real test but not in simulation, should we cast doubts upon car modelling or AD software, or both?

However this is the trend in car manufacturing industry [21], and it is believed that with more comparisons between real road tests and virtual simulation tests, the trustworthiness and fidelity of CLR method will be further improved.

The initial time plan is shown in Figure 7.1. Comparing this with the project progress, the time spent on the preparation of CLR is around 14 weeks actually, thus the time for CLR is not as much as planned. Most time is spent on the validation of SPAS core vehicle part since the braking model took more time to change



Week	Tasks
Week 1	Preparation and discussion
Week 2	
Week 3	Open-Loop Repaly using CADS4 SW
Week 4	
Week 5	SPAS Validation with Core Vehicle
Week 6	
Week 7	Repalce CADS4 with DriveMe
Week 8	
Week 9	Relpace CADS4 with DriveMe in CLR
Week 10	
Week 11	Closed-Loop Re-simulation and data analysis
Week 12	
Week 13	
Week 14	
Week 15	
Week 16	
Week 17	
Week 18	Master Thesis Final Report
Week 19	
Week 20	

Figure 7.1: Initial Time Plan. CADS4 (Collision Avoidance Driver Support) is the name Volvo used internally for ASFG

and test than expected. Also, DriveMe is not delivered to me during the project since it is not converted to a testable software on PC at that time, therefore all tests have to be replaced with ASFG. As a result, no comparison can be done between ASFG and DriveMe; all CLR tests have to be run with ASFG instead of DriveMe.

7.3 Ethics

The development of AD cars is to make human driving safer and more comfortable, but some people argue that AD cars are taking most of the driving pleasure of human drivers. It is very hard to answer questions like how smart our cars should be and should the future AD cars take over human driver completely. When all car companies are competing against each other in order to release fully-autonomous cars to attract media attention and make more profit, how much can we trust them? Even one single error in AD functions can cause accidents, let alone there are other non-AD cars driving on roads.

Also, if there is an accident caused by future AD cars, who is to blame? Should we blame AD software developers, car hardware manufacturers or potential human driver malfunctioning? When future AD cars are highly autonomous and networked, is the information of private car owners still private to themselves? What if the government or some organization secretly use the driving data to monitor on the general public for its own use.

When all cars are networked, if there is one hacker who can infiltrate into the network, tremendous impact can be cast upon the huge traffic flow and accidents can be caused easily.

7.4 Future work

More work and analysis will be dedicated to CLR tests with scenarios extracted from Volvo expedition logs after this thesis. If the braking validation in simulation is successful, more complicated scenarios will be included and AD functions like auto-steering and lane-following will be introduced as well to make CLR a more trustworthy method.

Bibliography

[1] Helmer, Thomas, et al. *"Safety Performance Assessment of Assisted and Automated Driving by Virtual Experiments: Stochastic Microscopic Traffic Simulation as Knowledge Synthesis."*. 2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2015.

[2] Stellet, Jan Erik, et al. *"Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions."*. 2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2015.

[3] T. Helmer, L. Wang, K. Kompass, R. Kates *"Safety Performance Assessment of Assisted and Automated Driving by Virtual Experiments: Stochastic Microscopic Traffic Simulation as Knowledge Synthesis."*. 2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2015. ISSN 2153-0017

[4] F. Fahrenkrog, A. Zlocki, and L. Eckstein, Bewertung Aktiver Sicherheit *"Vom Test zur Wirksamkeitsanalyse, ATZ, vol. 1, no. 116, pp. 3439, 2014."*.

[5] T. Helmer, M. Neubauer, S. Rauscher, C. Gruber, K. Kompass, and R. Kates *"International Symposium and Exhibition on Sophisticated Car Occupant Safety Sys- tems, ch. Requirements and methods to ensure a representative analysis of active safety systems"*. Fraunhofer-Institut für Chemische Technologie ICT, 2012. ISSN 0722-4087

[6] R. Kates, O. Jung, T. Helmer, A. Ebner, C. Gruber, and K. Kompass, *"Stochastic simulation of critical track situations for the evaluation of preventive pedestrian protection systems"*. Erprobung und Simulation in der Fahrzeugentwicklung, 2010.

[7] Stellet, Jan Erik, et al. *"Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions."*. Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on. IEEE, 2015.

[8] T. Helmer, *"Development of a Methodology for the Evaluation of Active Safety using the Example of Preventive Pedestrian Protection."*. No. ISBN 978-3-319-12888-7 in Springer Theses, Springer, 2015.

[9] *"EuroNCAP, BMW Pedestrian Warning with City Brake Activation."*. http://www.euroncap.com/ en/ratings-rewards/euro-ncap-advancedrewards/ 2014-bmw- pedestrian-warning-with-city-brake-activation/, as of June 29, 2015.

[10] J. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels *""Testing of advanced driver assistance towards automated driving: A survey and taxonomy*

on existing approaches and open questions". Intelligent Transportation Systems (ITSC) 2015 IEEE 18th International Conference on, pp. 1455-1462, Sept 2015.

[11] W. Wachenfeld, H. Winner *"Virtual Assessment of Automation in Field Operation A New Runtime Validation Method"*. Workshop Fahrerassistenzsysteme 2015, Walting, 2015

[12] H. Winner *"Einrichtung zum Bereitstellen von Signalen in einem Kraftfahrzeug"*. Patent DE 101(02), 771 (2001)

[13] A.Reschka., J.Rieken, M.Maurer: *"Entwicklungsprozess von Kollisionsschutzsyste- men fr Frontkollisionen: Systeme zur Warnung, zur Unfallschwereminderung und zur Verhinderung. In: Winner, H., Hakuli, S., Lotz, F., Singer, C. (Hrsg.) Handbuch Fahrerassistenzsysteme, 3rd edn., pp. 913935. Vieweg-Teubner-Verlag (2015)"*.

[14] *Euro NCAP AEB City Assessment Protocol v1.0, March 2013.*

[15] *Euro NCAP AEB Inter-Urban Assessment Protocol v1.0, March 2013.*

[16] *Euro NCAP AEB Test-protocol v1.1 June 2015.*

[17] *https://www.euroncap.com/en/about-euro-ncap/, June 27th 2017.*

[18] M.Guldogus: *"Proof of Concept of Closed-Loop Re-simulation Method in Verification of Autonomous Vehicles, June 2017"*.

[19] M.Guldogus: *"Chapter 2, Proof of Concept of Closed-Loop Re-simulation Method in Verification of Autonomous Vehicles, June 2017"*.

[20] M.Guldogus: *"Chapter 3, Proof of Concept of Closed-Loop Re-simulation Method in Verification of Autonomous Vehicles, June 2017"*.

[21] https://automotivemegatrends.com/automotive-megatrends-magazine-q3-2015 *"From physical to virtual: will technology see the end of the traditional crash test?, Oct 2015"*.

[22] https://www.euroncap.com/en/vehicle-safety/the-ratings-explained/safety-assist/aeb-interurban/ *May 28th 2018*