CHALMERS

Development of a Parameterized Passenger Vehicle Model for Longitudinal Dynamics for a Desktop Driving Simulator

Master's Thesis

MATTEO SANTORO

Department of Signals and Systems
Division of Automatic Control, Automation and Mechatronics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2014
Master's thesis 2014:08

MASTER'S THESIS

Development of a Parameterized Passenger Vehicle Model for Longitudinal Dynamics for a Desktop Driving Simulator

MATTEO SANTORO

Department of Signals and Systems
Division of Automatic Control, Automation and Mechatronics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2014

Development of a Parameterized Passenger Vehicle Model for Longitudinal Dynamics for a Desktop Driving Simulator
MATTEO SANTORO

Master's Thesis 2014:08
ISSN 1652-8557
Department of Signals and Systems
Division of Automatic Control, Automation and Mechatronics
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Cover:
General view of Desktop Driving Simulator, located at Chalmers, University of Technology, Göteborg, Sweden.

Development of a Parameterized Passenger Vehicle Model for Longitudinal Dynamics for a Desktop Driving Simulator
Master's Thesis
MATTEO SANTORO
Department of Signals and Systems
Division of Automatic Control, Automation and Mechatronics
Chalmers University of Technology

ABSTRACT

Driving Simulator are an important research tool in the automotive field. They are used to simulate and verify several vehicles and driving behaviour in a realistic as well as conditioned environment. In essence, driver simulators are especially valuable in gauging drivers action and perception, which cannot be adequately simulated or are less suitable in testing in real vehicles.

Adequate and accurate vehicle model, representative of the real vehicle behaviour in different driving conditions, is needed. However, to model a certain vehicle is a non-trivial, expensive and time consuming task. Advanced driver simulators are available in Sweden, especially kept by VTI in Göteborg and Linköping, which provide a very realistic response and simulate, very accurately, a real-time scenario.

This master thesis report describes a Desktop Driving Simulator, located at Chalmers University of Technology, which can simulate an experiment at an office level before moving to advanced simulators or moving to test track.

Moreover, the thesis describes the hardware of the Desktop Driving Simulator and its software system in a very detailed.

Several simulations were also performed with an Anti-lock braking system (ABS)-equipped vehicle. It is used in the modern cars to prevent the wheels from locking after brakes are applied. So, a Vehicle Dynamics Model with an ABS subsystem was therefore developed, using Simulink. The vehicle model was parameterized for a passenger vehicle, such as a Volvo S40 (1.6 ton).

Different cases were analized, such as Straight-Line Braking and Double-Lane Change in order to compare offline and online simulations and make some considerations on driver behaviour and influence.

Key words: Driving Simulator, software, ABS, simulations.

The Master's Thesis has been performed during the Erasmus exchange period at the *Chalmers University of Technology* (Göteborg – Sweden). The agreement has been with the *Università degli Studi di Salerno* (Salerno – Italy) and it has lasted 6 months, from March 2014 to August 2014.
This Thesis has been developed in collaboration with Arpit Karsolia, who will publish his Master's Thesis in September 2014.

ACKNOWLEDGEMENT

This Master´s thesis has been performed from March 2014 to August 2014 at Chalmers University of Technology, Göteborg, Sweden, in close collaboration with VTI and SP.
The thesis is a part of Chalmers work in the research project called ASTAZero SIM, founded by Vinnova, Reference (Diarenummer) 2013-04715, based on the ASTAZero Track, in Göteborg.
Offline and online simulations of the passenger vehicle model for the Desktop Driving Simulator were developed. The software used were MATLAB and Simulink. The Desktop Driving Simulator is located at the Department of Signals and Systems, at Chalmers University of Technology, Göteborg.
A part of the thesis has been carried out with Arpit Karsolia. I would like to thank him and my supervisors, Prof. Jonas Sjöberg and Prof. Bengt Jacobson. They helped me a lot during these months and they supported and endured me continuously, especially during some difficult times.
I would also like to thank many other people, such as Eleni Kalpaxidou, Jonas Andersson Hultgren, Sogol Kharrazi & Ingvar Nasman from VTI.

Göteborg, August 2014

Matteo Santoro

CONTENTS

1 Introduction

Driving Simulators represent an important tool to simulate different driving vehicles in a realistic way.
Essentially, they are used to verify drivers behavior and perception, which cannot be simulated in testing in real vehicles.
Data collection from real vehicle testing is a not trivial, expensive and time-consuming task. There are some advanced simulators in Sweden, especially kept by VTI in Göteborg and Linköping, that allow to simulate an experiment in real-time.

Nowadays, driving real vehicle on the road in a safe way has become complicated for many reasons, so using a driving simulator is becoming really significant. Every test needs a long and complex process of simulation and validation. Indeed, a big amount of time in Driving Simulators is often used for debugging/changing the experiment. The solution adopted in this project is a Desktop Driving Simulator, which can simulate an experiment/scenario at an office level before moving to more advanced simulators or moving to test track.

However, results obtained cannot be exactly compared with those of advanced simulators, especially regarding drivers behaviour, but at least they could provide some indications.

1.1 Driving Simulator and its role

One of the most common and universal tasks people perform every day is driving. It seems a really ordinary action, but it actually needs all of cognitive, perceptual, sensory and motor functions.

Different experimental studies can always be performed with real vehicle on-road tests, but using a Driving Simulator is better for several reasons, such as safety and cheapness. Moreover, it gives measures of driver behavior that are repeatable and objective, it allows for complete control of the driving environment and finally it can be managed in a laboratory setting.

Driving simulation is a really powerful tool in the automotive field, but it also other areas of research, such as vehicle dynamics, motion cueing, sound and graphics rendering and much more are involved. Moreover, the studies about the Driving Simulators apply to many areas of transportation research, like road, infrastructure and vehicle design, but also human factors, driver interaction with the vehicle (for instance steering or braking) and the traffic environment.

The main advantages of Driving Simulator are listed below:

- repeatability in the experiments: this factor gives the possibility to collect data that may take much longer time in a real life driving
- objective performance scoring: this because the drivers face the same driving conditions

- control of the driving environment: this is very important, because removing unnecessary elements from the scenario, the driver can drive carefully and can change driving tasks in an easy way; indeed, events, traffic or environmental conditions can be set and rapidly controlled
- minimal risk to drivers

Using of simulations is growing up, especially because the cost of the hardware is decreasing. For this reason, many of the national road safety research institutes, but also some of universities now consider Driving Simulators as a main part of their research.

1.2 Goals and objectives

The Desktop Driving Simulator allows to test vehicle before moving to test track. Different aspects are concerned by this master project.

Firstly, this thesis is a part of Chalmers work in the research project called ASTAZero SIM, founded by Vinnova, Reference (Diarenummer) 2013-04715, based on the ASTAZero Track, in Göteborg. ASTAZero (the Active Safety Test Area) is the world's first full-scale test facility for future traffic safety solutions. It opened on August 21st, giving the possibility to all vehicles manufacturers and safety developers to test all modes of road transportation in different types of scenario, such as rural road, highway and city area. So the main goal of this project is to perform some pretests before driving in ASTAZero Track.

Other aspects are faced in this thesis: in Chapter 2 can be found a very detailed description on the Desktop Driving Simulator and its hardware system and its software. But the thesis also provides a model of the Anti-lock Braking System (ABS) to the vehicle model for the driving simulator and a parameterized passenger vehicle model for the simulator. The ABS system has been fitted to the original Vehicle Dynamics Model (VDM) developed and provided by VTI and it has been used in both offline and online simulations, running simple case/scenarios for basic verification of the model

A great help in this thesis was offered by VTI. Swedish Road and Traffic Research Institute, also known as VTI, is an important research institute in the transport sector; it is specialized in the automotive field, but also other areas, such as traffic and driving simulation, passive safety, tire testing. In particular, VTI uses some simulators to perform and run some simulator experiments.

1.3 Why the Desktop Driving Simulator?

The Desktop Driving Simulator can be considered a step between offline simulations and tests in advanced simulators or in real vehicles. It allows to perform tests in a safe and cost effective way.

For this reason, some specific requirements on the Desktop Driving Simulator are needed. It should be:

- portable, in order to be used in several working environments, e.g. different researchers at different offices

- easy to change the Vehicle Dynamics Model
- easy to change the scenario, which includes the road or test track plus the surrounding traffic

For more details about the original VDM and the further modified model that incorporates the ABS system used in the Desktop Driving Simulator and implemented during this thesis, see Chapters 3 and 4. Anyway, this model satisfies some characteristics: it is flexible, so that it can be used to represent different passenger vehicles; it is quite simple, in order to be modified in an easy way and to avoid some numerical problems that could happen during testing, causing instability; it is able to be run in real time.

As already stated in the previous section, the overall motive of this thesis is the preparation for tests in more advanced simulators or in real vehicles. Anyhow, some limitations can be listed regarding using of the Desktop Driving Simulator: it is not important with realistic driver experience; only an automatic transmission has been used (no gearstick) and no traction control (it is only by brake interventions, no clutch/differential control).

1.4 Thesis outline

Chapter 2 includes a comprehensive and exhaustive description about the Desktop Driving Simulator used in this thesis, its hardware and software, how it works and how to run a simulation. How to create and run a project can be found in Appendix B.

Chapter 3 contains a brief description and analysis of the original Vehicle Dynamics Model provided by VTI.

Chapter 4 describes, after a short introduction on the Braking system, the Anti-lock Braking system (ABS) and its importance, giving greater importance to the wheel slip ratio, which mostly influences the ABS. This chapter also includes the ABS Simulink model and the parameterized passenger vehicle model for the simulator.

Chapter 5 explains the offline and online simulations performed in this report for basic verification of the model.

A brief introduction to the vehicle dynamics and some basic concepts are included in Appendix A.

Appendix C contains some tables that represent the input parameters data for every subsystem of the Vehicle Dynamics Model used in this thesis.

Appendix D includes some information about the logged data.

In Appendix E are showed the results of the offline and online simulations for basic tests, such as Straight Ahead Acceleration, Straight-Line Braking and Double-Lane Change.

1.5 Splitting of the work

A part of this thesis has been carried out with Arpit Karsolia. The main goals of his thesis have been: establish modularity of the vehicle model –

parameterization to multiple vehicles; establish ESC/ABS functionality with moderate levels of tuning; study scenario III and motivate driver behavior with simulations. He carried out some offline and online tests of the Vehicle Dynamics Model with 3 different sets of ambulance vehicle data, that is Volvo S40, Mercedes Sprinter (2.8ton) and Mercedes VitoXL (3.5 ton), developing the following manoeuvers: Straight-Line Braking, Sine wave with Dwell, Double-Lane Change and Scenario III.

The main objectives of my project, as has been stated in Section 1.3, have been: a complete description on the Desktop Driving Simulator; the modeling of the Anti-lock Braking System (ABS) to the vehicle model for the driving simulator; the running of some offline and online simulations, as will be shown in the thesis.

2 Desktop Driving Simulator environment

This chapter gives a short overview of the Desktop Driving Simulator (DDS). During this thesis, it was located at the Department of Signals and Systems, at Chalmers University of Technology, Göteborg, starting from the end of April.

2.1 Desktop Driving Simulator Hardware system

The Desktop Driving Simulator system, shown in Figure 2.1, consists of the following main parts: three screens, steering wheel and pedals, computer simulation (Simulator PC) with four cores and powerful graphics board and the computer (xPC target PC) running the external vehicle model.

In *Figure 2.1* are shown three screens: the High Definition (HD) screen is in the middle, where the simulations run, on the left there is another one used to run xPC target model (more details in Section 2.2.2.1) and finally on the right a third screen, only employed graphically.

Figure 2.1: Desktop Driving Simulator located at Chalmers University of Technology

2.1.1 High Definition screen

The main screen used for the Desktop Driving Simulator is a BenQ 27 \ "LED GL2750HM. Its dimensions are the following: HxWxD = 488.5 x 654.4 x 191.2 mm.

2.1.2 Steering wheel and pedals

The Logitech G27 is an electronic steering wheel designed for Sim racing video games on the Personal Computer (PC) and Play Station (PS). It uses a Universal Serial Bus (USB) interface.

A small picture (*Figure 2.2*) of it is shown below.

Figure 2.2: Leather-wrapped steering wheel

Figure 2.3: Stainless steel pedals

Some specifications are listed below.
It features:

- A steering wheel:
  - 270 mm, leather-wrapped steering wheel
  - Range of rotation adjustable up to 900 degrees
  - 2 force feedback motors
  - One set of gears between motors and wheel, including an anti-backlash design
  - 6 buttons
  - 2 paddleshifters
  - Dual-motor force feedback, that provides excellent steering response
- A set of stainless steel pedals (*Figure 2.3*), including:
  - Accelerator (light spring)
  - Brake (heavy spring)
  - Clutch (medium spring)
  - A carpet grip, which keeps the pedals in position while playing
- A shifter unit:
  - 8 buttons
  - 1 D-pad

- A gear stick with a six-speed 'H' pattern gearbox; reverse is selected by pressing down and changing to sixth

In this thesis, the clutch and gear stick were not used: the Vehicle Dynamics Model adopted an automatic transmission.

2.1.3 Central Processing Unit (CPU) simulation

As regard the computer simulation, it has a Webhallen Configuration D13-0405 - i7-4770K / 16GB / 120GB SSD + 2TB / DVD / Win8.1.

2.1.4 xPC target computer

This computer belongs to Chalmers University of Technology (*Figure 2.4*).

2.2 The computers network

The Desktop Driving Simulator uses two computers to run, which are connected each other and exchange data between them and a small chart of the communication is shown in *Figure 2.4*.

2.2.1 Simulator PC

It manages the entire simulation. It is responsible to run the virtual environment model and the graphical rendering, to handle the input/output transfer from/to the other parts of the Desktop Driving Simulator, to control the errors, to exchange signals with the Vehicle Dynamics Model in the xPC target PC, to generate the scenario, etc..

The operating system is Windows 7. The software of the Desktop Driving Simulator, instead, is developed at VTI; it is basically a C++ code, which includes several classes, among which: data storage, User Datagram Protocol (UDP) routing, eXtensible Markup Language (XML) reader, etc.. Moreover, its functionalities can be extended developing customised plugins to meet the customer's requirements (for example to control the scenario or to manage the events, etc..).

2.2.2 xPC target PC

It runs the external Vehicle Model; all the inputs from and to the Desktop Driving Simulator go via the xPC Target computer.

Both the computers communicate via Ethernet (LAN, that is Local Area Network).

How the Vehicle Model communicates with the software is explained in Section 2.4.2.

2.2.2.1 xPC Target

xPC target is a host-target solution for testing real-time systems using standard PC hardware.
It is an environment that uses a target PC, separate from the Simulator PC, for running real-time processes.

Now, in MATLAB R2014-a, xPC Target™ has been updated and renamed to Simulink® Real-Time™.

The Vehicle Dynamics Model, developed in Simulink, is executed on the computer running the xPC Target operative system.
Besides, xPC Target allows to add I/O blocks to the model.

Real-time systems are able to run programs with very precise timing requirements. A Real-Time Operating System (RTOS) is the main component of the real-time systems. A general-purpose OS (such as Windows) usually runs many processes and applications at once and also provide other features, like user interface graphics; this could cause a delay when an user program is executed. Instead, a Real-Time Operating System has the main purpose to execute a single program with very precise timing.

2.2.3 Communication between the computers

The two computers are interconnected in a wired LAN network and the exchange of heterogeneous data between them happens through the UDP protocol (User Datagram Protocol). Further information about that can be found in Section 2.4.1.

The communication speed between the xPC Target PC and the Simulator PC is set to 60 Hz. If data is sent at a different speed, the Desktop Driving Simulator may lose the communication.

2.3 Desktop Driving Simulator Software

The simulation software consists of Core, VISIR, SIREN and Vehicle Dynamics Model of course. SIREN is used to compute the audio, VISIR is the VTI in-house developed graphical image generator, responsible to compute the virtual environment. Regarding the Core, one of the tasks is to run the scenarios and to send data to VISIR, SIREN and VDM (*Figure 2.4*).

The tools used for the software are: xPC target model, MATLAB R2014-a, Simulink and Qt Creator version 5.2.1.

Qt Creator is a software environment for configuring real-time testing processes. It helps the Simulator PC in several tasks, such as running simulations from Simulink, sending and receiving data. Moreover, it allows to interact with these tasks using a powerful and editable user interface.

In Appendix B of this report is described how to create and run a project using Qt Creator environment. For further information consult the reference manual of the software.

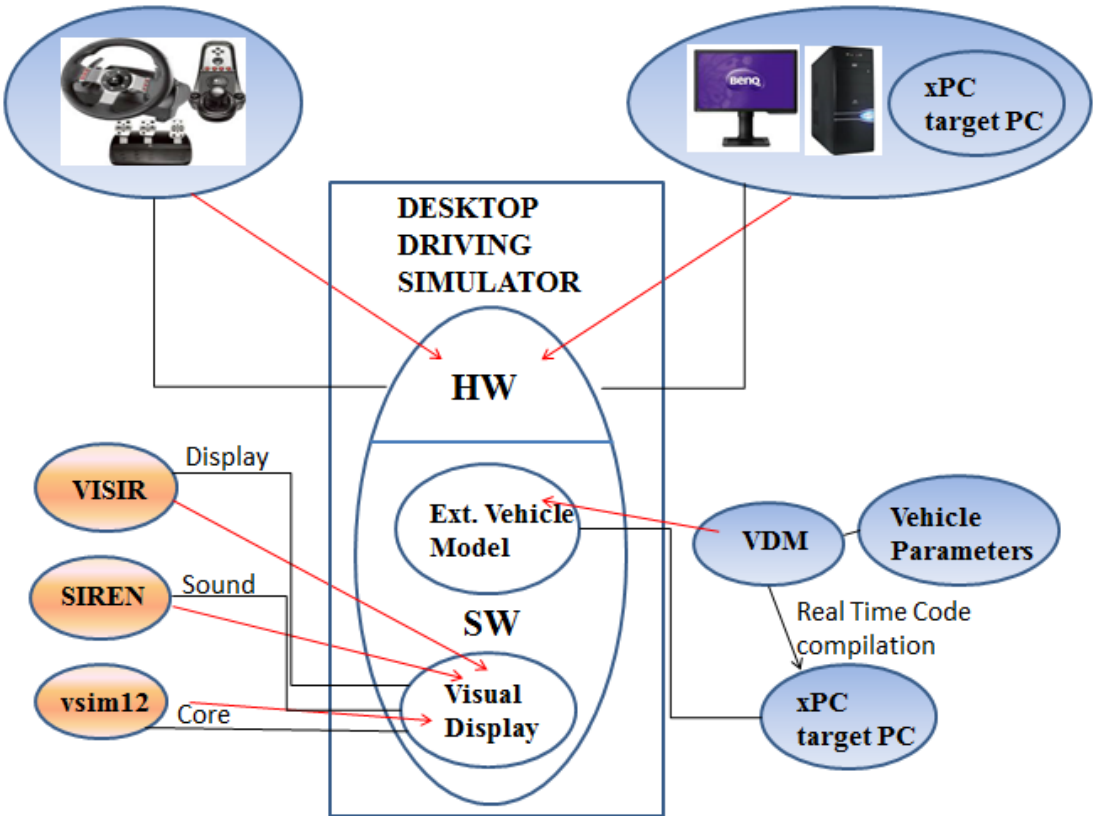


Figure 2.4: Desktop Driving Simulator design

2.4 Working with VTI's code

In this section an introduction on the simulator code, developed at VTI, is given.
The purpose of this primer is to facilitate the other users to make changes in the code quickly, in order to meet their requirements.

As has been said in Section 2.2, the Desktop Driving Simulator consists of two main blocks (Simulator PC and xPC target PC), which are interconnected and they exchange data with different modalities.
Indeed, the computers are connected each other in a wired LAN network. In order to interact with the simulation parameters, is necessary to slightly modify the VTI's simulator code, implementing new functionalities.

The VTI code is located in the Simulator PC in the directory:

Computer/Local Disk(C:)/astazerosimchalmers/vsim12.pro

2.4.1 UDP Internet Protocol

The exchange of data between the computers of the Desktop Driving Simulator happened through the User Datagram Protocol (UDP). This protocol provides a a direct method to send and receive messages over an IP (Internet protocol) network.
The protocol is with no connection: the main handicap is that the delivery and the duplicate protection are not secured.

The UDP protocol needs the definition of the IP address of the Simulator PC and the port numbers of both the PCs (Simulator PC and xPC target PC), as shown in *Figure 2.5*. The two ports serve to identify the end-points within the two computers. The port of the Simulator PC needs when a reply has to be sent back to the other computer.

IPAddressxPC: 127.0.0.1 IPAddressSimulatorPC: 169.254.247.23

PortAddressxPC: 56072 PortAddressSimulatorPC: 5154

Figure 2.5: The UDP layout

The source code can be found in:

Computer/LocalDisk(C:)/astazerosimchalmers/project/astazerosimchalmers/udpservice.hpp

and

Computer/LocalDisk(C:)/astazerosimchalmers/project/astazerosimchalmers/udpservice.cpp

(.hpp means headers plus plus and .cpp means C++).

For the IP addresses and the port numbers of the computers, see also the file:

Computer/Local Disk(C:)/astazerosimchalmers/conf/desktop.xml

2.4.2 How to exchange data between the simulator and the model

In order to interact directly with the Vehicle Dynamics Model (developed in Simulink), it needs to exploit the *PacketFromSimulinkCar*, which is defined in:

Computer/LocalDisk(C:)/astazerosimchalmers/dynamics/simulinkcar/packetsimulinkcar.hpp

and it contains interface classes for the *simulink_car* model.
In addition to this, other changes are required:

in the file: **Computer/Local Disk(C:)/astazerosimchalmers/conf**

change vehicle model syntax to "*simulink_car*" (initially set to "*joystickdynamics*").

The "*joystickdynamics*" model is a different vehicle model, which is defined in *JoystickDynamics* class in the C++ code, and it has mostly a graphics function. When this one is used, the "*simulink_car"* model is not used at all (no data is sent to or received from the VDM).

Moreover, in:

Computer/Local Disk(C:)/astazerosimchalmers/conf/desktop.xml

set the IP address of the xPC target computer.

Instead, as regard the Simulink model, the following changes are needed:

- set xPC target PC IP address in *UDP Send block* (see Section 2.4.1); receive block can read 0.0.0.0 as accepting all IP

- run *vti_def.m* file (further details in Section 5.4), as it contains input UDP parameters.

2.4.2.1 Interaction requirements

- Use *Wireshark* (network protocol analyzer) to establish whether packets are transmitted and received between the two

- Switch off firewall when running simulation

- Check MATLAB compiler version, "*R2014-a*" doesn't have an internal compiler;
    - setup compiler using mex setup (MATLAB function)

2.4.3 Adding a simulation event in the code

The regular execution of the driving scenario may be changed with an event, for example presenting to the driver activities.
An event is triggered when the established condition is met.

```
#include "scenario.hpp"
#include "../projectcommon.hpp"

namespace astazerosimchalmers {
	ImplScenarioFactory scenario_factory;

	ImplScenarioFactory::ImplScenarioFactory() }
		core::ScenarioHandler::instance().registerFactory("astazerosimchalmers", this);
	}

	void ImplScenarioFactory::produce(core::ScenarioHandler* scenario) {
		core::TriggerPtr init = core::TriggerPtr(new InitTrigger());

		scenario		visadd...("_distance", "distance", 100000.0, init, true);
		scenario		visadd...("_snow", "enabled", false, init, true);

		scenario		datadg...("enable", true, init, true);
```

Figure 2.6: Example of a trigger scenario.hpp

In order to define an event, open the file:

Computer/LocalDisk(C:)/astazerosimchalmers/project/astazerosimchalmers/events.hpp

and in the module *produce* of the class *ImplEventFactory,* is necessary initialize the event (*Figure 2.7*).

```
namespace newname {
	struct ImplEventFactory : public core::EventFactory {
		ImplEventFactory();
		void produce(core::EventHandler* handler);
	};
```

Figure 2.7: Initialization of the event in event.hpp

Make an example (*Figure 2.8*):

```
namespace conetrack {
	class ConeTrackBase : public core::Event {
	public:
		ConeTrackBase(int id, const std::string& tag, int nof_cones);
		void init();
		void interalReset();
		void step(const long iteration, const double timestep);


	protected:
		std::vector<StaticRoadActorPtr> m_cones;
		OpenDrivePosition m_end_pos;
	};
```

Figure 2.8: Example of a Cone Track event in event.hpp

Afterwards, it is possible to create a new class for the event in the file:

…project/astazerosimchalmers/events.hpp

Then in the file:

…project/astazerosimchalmers/events.cpp

one has to define the event and how it evolves. As reference, it is advised to read how other events have been previously implemented.

An event is identified by an ID that has to be unique: one has to increment the ID base (equal to 2400) by a user-defined number (e.g. if one adds 11, the event's ID is 2400 + 11 = 2411, as shown in *Figure 2.9*).

```
Event(id_base + 11, "name_of_the_event")


namespace astazerosimchalmers {
	const int id_base = 2400;
	ImplEventFactory event_factory;

	ImplEventFactory: :ImplEventFactory() : EventFactory("astazerosimchalmers_event_factory") {
		core::EventHandler::instance().registerFactory("astazerosimchalmers", this);
	}

	Void ImplEventFactory::produce(core::EventHandler* handler) {
	//Cone track slalom
	{
	core::EventPtr event(new conetrack::Slalom(id_base + 11, "come_track_slalom"));
	event→setHandler(handler);
	event→enableLog(true);
	event→setThread(0);
	event→setBlocking(true);
	handler→addEvent(event);
	}
```

Figure 2.9: Example of an event in event.cpp

The event is controlled using the variables and the methods defined in:

…project/astazerosimchalmers/scenario.hpp

and

…project/astazerosimchalmers/scenario.cpp

Once the event is created, one must to define which is the condition that triggers it (*Figure 2.10*). For further information about that, consult the reference manual of the software and the file:

Computer/Local Disk(C:)/astazerosimchalmers/core/scenario

```
scenario→setTimer(20,scenario→getTimeoutTrigger());
scenario→startEvent(2411,scenario→getTimeoutTrigger());

scenario→setTimer(40,scenario→getTimeoutTrigger());
scenario→startEvent(2401,scenario→getTimeoutTrigger());
```

Figure 2.10: Example of a trigger scenario.cpp

Moreover, another interesting aspect regards the possibility to generate some traffic. Considering always the file:

…project/astazerosimchalmers/scenario.hpp

and

…project/astazerosimchalmers/scenario.cpp

it is possible to enable it, typing true or false in the string of the code.

As shown in *Figure 2.11*, there are more options, such as generate oncoming, generate ahead, allow overtaking.

```
scenario→addAction("traffic:generate_oncoming", "enabled",true,init,true);
scenario→addAction("traffic:generate_ahead", "enabled",true,init,true);
scenario→addAction("traffic:allow_overtaking", "enabled",true,init,true);
```

Figure 2.11: Example on adding traffic in scenario.cpp

2.4.4 Changing the vehicle's shape in the code

It is possible to change the model of the vehicle (shape and dimensions) and it does not affect the dynamics, because it is computed within the code on the Simulator PC.

So, open the file:

Computer/LocalDisk(C:)/astazerosimchalmers/project/astazerosimchalmers/astazerosimchalmers.conf

and change the value of shape, with a different vehicle's name, as shown below in *Figure 2.12*:

```
…
<vehiclemodel type="simulink_car">
   <shapes>
    <shape type="volvo_s40_green"/>
   </shapes>
…
```

Figure 2.12: Example on changing vehicle's shape

The vehicle shapes' database (*Figure 2.13*) is in:

Computer/Local Disk(C:)/astazerosimchalmers/conf/shapes.xml

```
…
<!-- CARS -->
      <volvo_s40_red type = "ActorTypeCar"
                                          boundary = "rectangle"
                                          front = "1.00"
                                          rear = "3.47"
                                          width = "1.77"
                                          wheel_base = "2.60"
                                          track_width_front = "1.54"
                                          center_of_gravity = "-1.056, 0, 0"
                                          driver_position = "-1.2, 0.36, 0.0"
                                          dynamic_state_origo = "-1.056, 0, 0.0" >

      <wheels>
               <wheel id="11" tag="front_left_wheel" v="0, 0.752"/>
               <wheel id="11" tag="front_right_wheel" v="0, -0.752"/>
               <wheel id="21" tag="rear_left_wheel" v="-2.6, 0.752"/>
               <wheel id="-21" tag="rear_left_wheel" v="-2.6, -0.752"/>
      </wheels>
      </volvo_s40_red>

      <volvo_s40_green copy_of = "volvo_s40_red" />
      <volvo_s40_grey copy_of = "volvo_s40_red" />
      <volvo_s40_blue copy_of = "volvo_s40_red" />
      <volvo_s40_blue_foerst copy_of = "volvo_s40_blue" />
…
```

Figure 2.13: Example of a car in the vehicle shapes' database in conf/shapes.xml

2.4.5 Changing the road's scenario in the code

VTI adheres to the *OpenDRIVE* standard for the description of the road network. The data format is *Ascii* readable XML-structure (eXtensible Markup Language) and can be found in *xodrfiles*.
The road scenario is obtained putting together several small pieces of road. The whole database (*Figure 2.14*) is in:

Computer/LocalDisk(C:)/astazerosimchalmers/project/astazerosimchalmers/astazerosimchalmers.xodr

Each road is identified by a name, an unique ID, the length and the type of connection with the other different roads.

```
<road name="rural_1" id="1" length="1080" sOffset="0" junction="-1">
<road name="rural_2" id="2" length="2250" sOffset="1080" junction="-1">
```

```
<road name="rural_3" id="3" length="835" sOffset="3330" junction="-1">
<road name="rural_4" id="4" length="8.70" sOffset="4165" junction="-1">
<road name="memos_1" id="1001" length="10000" sOffset="0" junction="-1">
<road name="M_3_1" id="2001" length="12400" sOffset="0" junction="-1">
<road name="Herrbeta_1" id="3001" length="100" sOffset="0" junction="-1">
<road name="Herrbeta_2" id="3002" length="400" sOffset="0" junction="1">
<road name="Herrbeta_3" id="3003" length="700" sOffset="0" junction="2">
<road name="Herrbeta_4" id="3004" length="200" sOffset="0" junction="-1">
<road name="Herrbeta_5" id="3005" length="500" sOffset="0" junction="1">
<road name="Herrbeta_6" id="3006" length="2.12" sOffset="0" junction="-1">
<road name="Herrbeta_7" id="3007" length="400" sOffset="0" junction="2">
<road name="Norsholm_1" id="5001" length="1300" sOffset="0" junction="-1">
```

Figure 2.14: Roads' database in astazerosimchalmers.xodr

Moreover, it is also possible to turn the driving direction changing the value of *start_road_direction* with *+1* or *-1*.

Finally, it is possible to set urban and rural roads or a road with tunnels.

2.4.6 Compiling the code

When all the changes of the code have been done, one has to compile it. So, in Qt Creator click on *File* > *Save* (or Ctrl + S), then click *Build* (or Ctrl + B) and finally *Run* (or Ctrl + R).

If there are some errors, the debugger will help to locate and fix them.

2.4.7 Running the simulation

In order to run the simulation, follow the steps below:

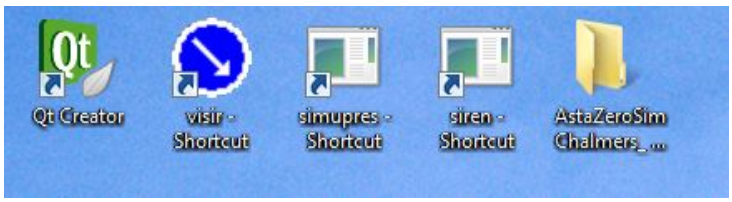


Figure 2.15: Shortcuts on the Simulator PC's desktop

1. Switch on the Simulator PC: it does not always work at the first attempt. Type "*dds2014*" as password.

2. Run VISIR in the Simulator PC: click on the shortcut located on the Simulator PC's desktop (*Figure 2.15*). The program will start to generate the scenario. When the procedure is completed, on the screen, it will appear the string *Scenario done* (*Figure 2.16*) and, on the second screen, will appear the generated scenario (*Figure 2.17*). As default, the road's scenario that will appear is called *malilla_w_1*.

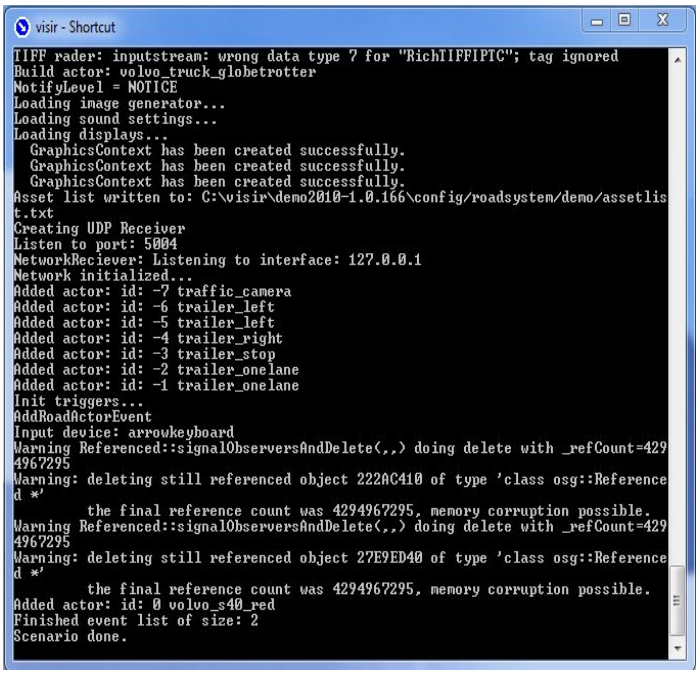


Figure 2.16: Running of VISIR

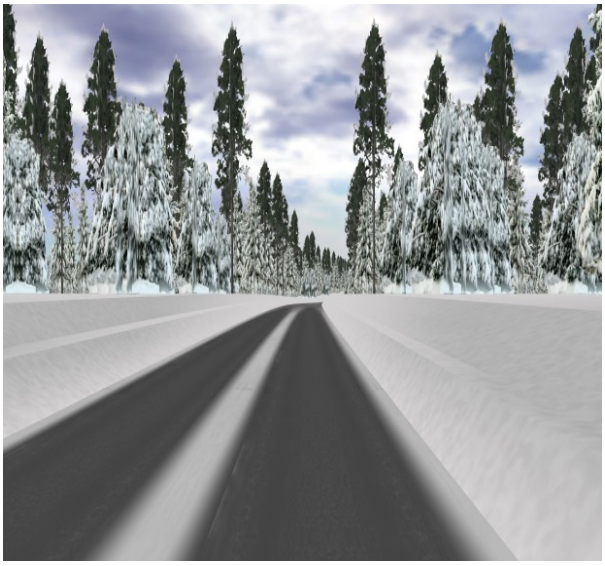


Figure 2.17: Scenario generated by VISIR

It has also been used a third screen just to have another view of the road, located on the right of the Simulator PC.

3. Run SIREN in the Simulator PC: click on the shortcut located on the Simulator PC's desktop (*Figure 2.15*). That is for the sound.

4. Run COMMAND PROMPT in the Simulator PC: click on the shortcut located in the lower left corner of the Simulator PC's desktop. Then, go to *cd C:\siren\soundfiles\csound*, press enter and type *"csound soundgen_xc60 _udp.csd"*.

5. Run SIMUPRES in the Simulator PC : click on the shortcut located on the Simulator PC's desktop (*Figure 2.15*). Then click on *Visa* and *Instrumentbrada*.

It will appear a dashboard (*Figure 2.18*). It is also possible set some parameters, such as *engine_revolution*, *ove_odometer*, etc..

Figure 2.18:Dashboard that appears running SIMUPRES

6. Run QT CREATOR in the Simulator PC: click on the shortcut located on the Simulator PC's desktop (*Figure 2.15*) and then open vsim12 project, as showed in Appendix B. On the left of the Qt Creator User Interface, go to *Projects* and then select *Run.*

In the Executable, there is the string *C:\astazerosimchalmers\debug\vtisim.exe*; so, in *Arguments* type: *"-p astazerosimchalmers –d j –S –v 127.0.0.1 5004 –a"* (*Figure 2.19*).

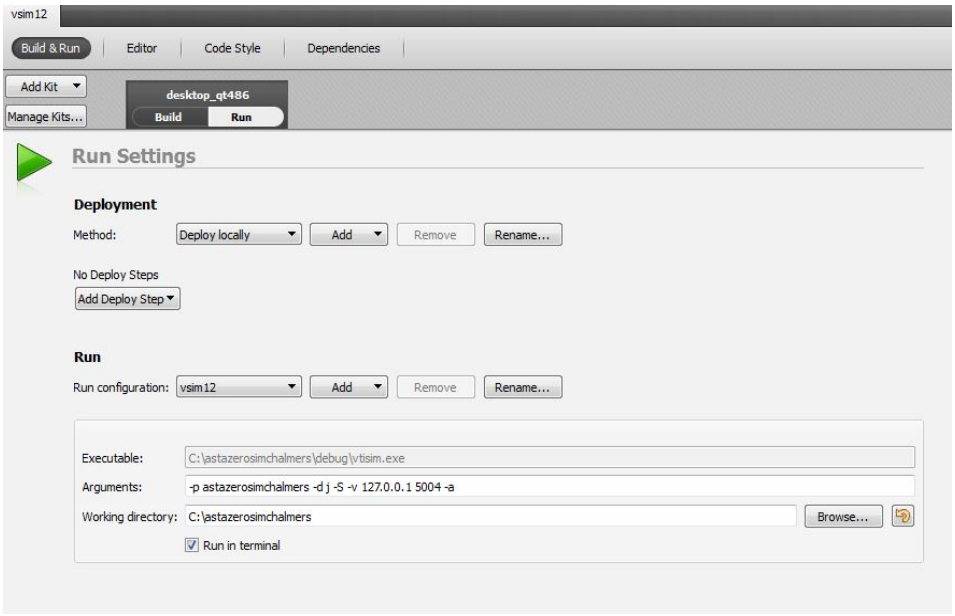


Figure 2.19: Run settings in Qt Creator

7. Insert USB pen drive in the Chalmers PC: put into the USB pen drive two files: *BOOTSECT.RTT* and *xpctgb.RTA*; they need to boot up Simulink Real-Time. Then, insert carefully USB pen drive in the xPC target PC.

8. Run MATLAB R2014-a in the Simulator PC: click on the shortcut located on the Simulator PC's desktop (*Figure 2.15*) and in the *Command Window* of MATLAB type *"slrtexplr"*. A screen, like the one depicted in *Figure 2.20*, will appear. It is *Simulink Real-Time Explorer* environment. In the first window on the left (*Targets*), do right click on *TargetPC1* and select *Properties.* At this point, click on:

- *Host-to-Target communication* and inside the *Target Network Settings* make the following changes: as *IP address* put *169.254.247.23* (IP Address of the Simulator PC), as *Port* insert 22222, as *Gateway* type *129.16.176.1*

- *Target settings* and tick *USB support*

- *Boot configuration* and select *Removable Disk* as *Boot mode*

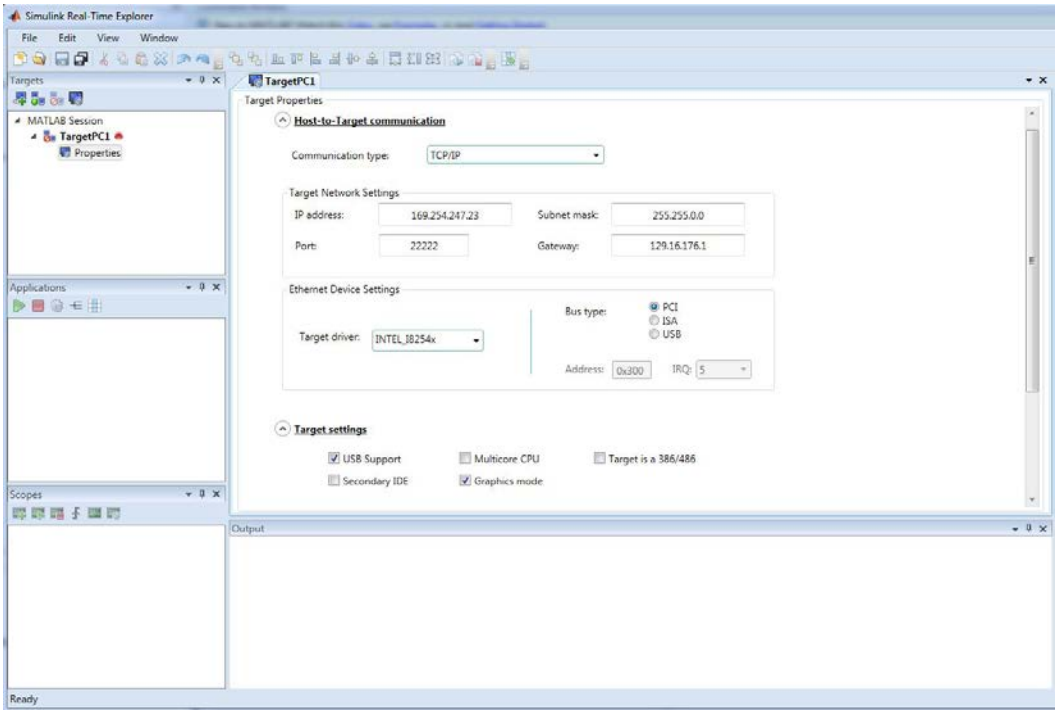


Figure 2.20: Simulink Real-Time Explorer interface

Run the .m files in MATLAB and the .mdl files in Simulink and then in the same environment click on *Build Model*. So, return again in Simulink Real-Time Explorer and in the first window on the left (*Targets*), do right click on *TargetPC1* and select *Connect*, while in the second window below (*Applications*), will appear *TargetPC1/name of the Simulink file*, do right click on it and press *Start*.

Finally, to run the simulation, select *Run* in Qt Creator.
To stop it, press Ctrl + C.

2.5 Steering wheel force feedback

The steering wheel force feedback is a very important aspect in the Desktop Driving Simulator, even if it represents a limitation of this project. Through the steering wheel, the driver gets some information on the driving conditions and that influences the perception of realism driving the simulator. This would imply a careful study on the steering wheel force feedback, but due to time restrictions, it was not possible to do that in detail during this thesis.

It is "*Built-in*" torque plus the requested torque from the Vehicle Dynamics Model. The "*Built-in*" torque can be tuned thanks to the Logitech Gaming Software settings. To do that, click on the shortcut located on the Simulator PC's desktop. A screen, like the one depicted in *Figure 2.21*, will appear.

Figure 2.21: Logitech profiler environment

Go to *Options* and then select *Global Device Setting*s. Here is possible to tune the "*Built-in*" torque via *Spring Effect Strength* and *Damper Effect Strength* in a range between 0 – 150 % (*Figure 2.22*).

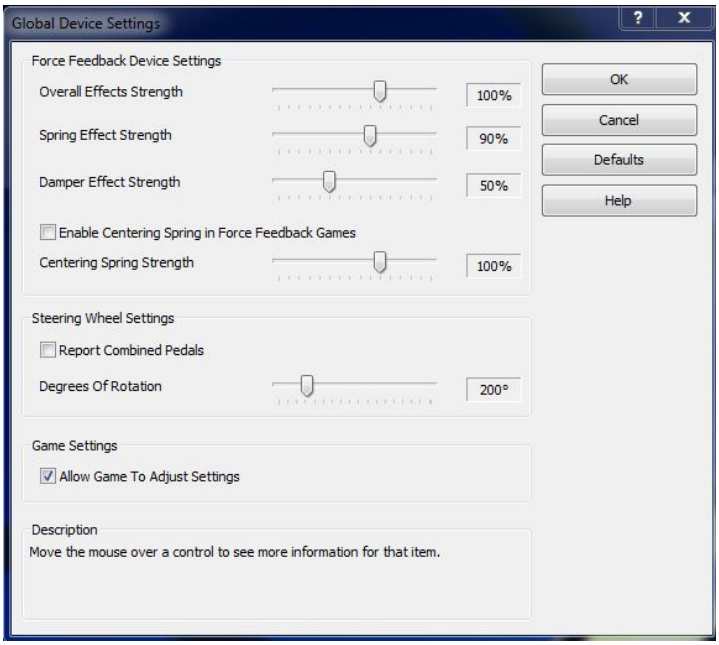


Figure 2.22: Global Device Settings in Logitech profiler

The steering wheel angle in vsim12 is possible to set up in the file:

Computer/Local Disk(C:)/astazerosimchalmers/conf/joystick.xml (*Figure 2.23*)

This value is the same also in the Logitech Gaming Software (*Options* > *Global Device Settings* > *Steering Wheel Settings* > *Degrees of Rotation* (maximum value is 900 degrees)).

To check the steering wheel angle sent to the Vehicle Dynamics Model from vsim12, it could bring up the "*steering_wheel_angle*" parameter in SIMUPRES.

```
<joystick>
    <settings name="g27" active="true">
        <throttle min="32500" max="-32500" axis="1"/>
        <brake min="32500" max="-32500" axis="2"/>
        <steering right="32767" left="-32768" deadzone="2" angle="900" axis="0"/>

    <haptic>
        <spring enabled="true"/>
        <damper enabled="true"/>
        <inertia enabled="false"/>
        <friction enabled="false"/>
    </haptic>
    </settings>
</joystick>
```

Figure 2.23: Settings of the joystick in joystick.xml

Going to the file:

Computer/LocalDisk(C:)/astazerosimchalmers/project/astazerosimchalmers/joystickdriver.cpp

there are the updates for spring, damper, friction and inertia effects (*Figure 2.24*)

```
if (iteration % 4 == 4) {
        Effect::Update update;
        update.left_coefficient  = p_stw_torque;
        update.right_coefficient  = p_stw_torque;
        update.left_saturation  = p_stw_torque;
        update.right_saturation  = p_stw_torque;
        m_joystick_handler.updateEffect(Effect::SpringEffect, update);

        update.left_coefficient  = p_stw_torque;
        update.right_coefficient  = p_stw_torque;
        update.left_saturation  = p_stw_torque;
        update.right_saturation  = p_stw_torque;
        m_joystick_handler.updateEffect(Effect::DamperEffect, update);

        update.left_coefficient  = p_stw_torque;
        update.right_coefficient  = p_stw_torque;
        update.left_saturation  = p_stw_torque;
        update.right_saturation  = p_stw_torque;
        m_joystick_handler.updateEffect(Effect::InertiaEffect, update);

        update.left_coefficient  = p_stw_torque;
        update.right_coefficient  = p_stw_torque;
        update.left_saturation  = p_stw_torque;
        update.right_saturation  = p_stw_torque;
        m_joystick_handler.updateEffect(Effect::FrictionEffect, update);
}
```

Figure 2.24: Updates for spring, damper, friction and inertia effects in joystickdriver.cpp

3 Vehicle Dynamics Model

In this section a brief overview on the Vehicle Dynamics Model is developed. What follows is not a complete description of the model, but just a really simplified overview of what the Vehicle Dynamics Model contains and its sub-systems. The original model used in this thesis has been developed by Sogol Kharrazi, a researcher at VTI in Linköping. The name in Simulink of the model is *Simulink_Car.mdl*.

More detailed information about the VDM can be found in Karsolia A., (2014:06, ”To be published”), Master’s thesis, Department of Applied Mechanics, Division of Vehicle Engineering and Autonomous Systems, Chalmers University of Technology, Göteborg, Sweden.

Therefore, is recommended to read this chapter first before going in the file in Simulink, so that it is possible understand the structure and how it works. Some comments are also included in each block in the Simulink file with further explanations.

3.1. Structure of the model

The model was created in MATLAB and Simulink and then converted to C++ code.

It can be compiled and then loaded on the Simulator PC, being supported by the software.

As has been stated in Section 1.3, one of the main characteristics of the Vehicle Dynamics Model is the flexibility. The Simulink model needs to be clear, easy to understand and well organized. So an important starting point for the project is its organization.

The original model is divided into several subsystems, such as a real vehicle can be divided, where inputs enter to the left and outputs exit to the right. In *Figure 3.1* is shown the model layout.

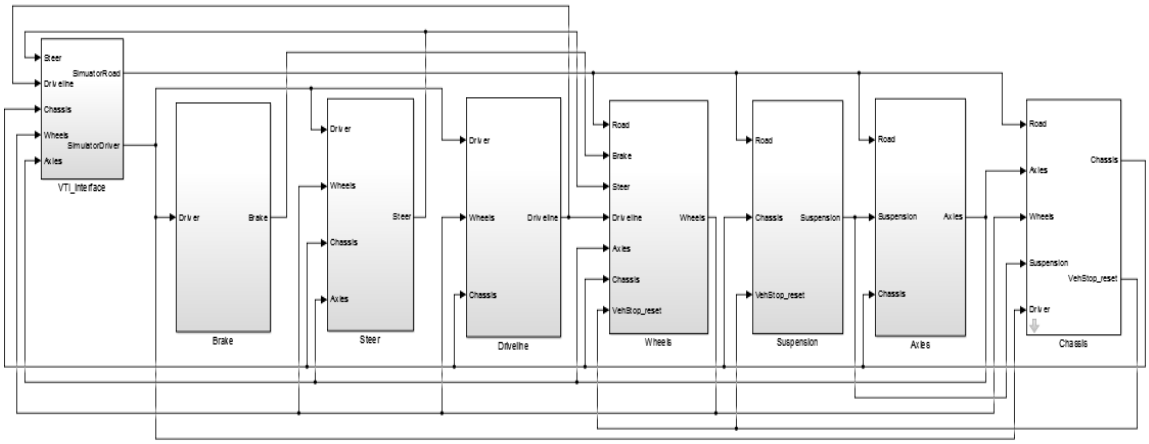


Figure 3.1: Overview of the original Vehicle Dynamics Model

As represented in *Figure 3.1*, the Simulink VDM is made up of seven main interconnected blocks, plus on the left there is the block that shows the VTI interface.

The main components, with a really short description, are listed below:

1. Braking system: it is used to compute the braking torques at each wheel of the vehicle. It will be analyzed in detail in Chapter 4.

2. Steering system: its main purpose is to calculate the steer angle of the wheels as a function of the driver's input through the steering wheel and driving conditions. Another output of this system is the steering wheel torque.

3. Driveline system: it gives the driving torques of the vehicle. It also models the behaviour of the transmission and the engine of a real vehicle.

4. Wheels system: it symbolizes the behaviour of the tires of the real vehicle.

5. Suspension system: it represents the suspension system of the vehicle. Its main objective is calculating the load transfers generated when driving.

6. Axles system: it defines the axles system of the vehicle.

7. Chassis model: in this component, the differential equations used to calculate the vehicle motion are solved.

8. VTI Interface: in this interface all the inputs and the outputs signals are defined in the UDP block of this system.

The different components of the VDM are studied deeply in Karsolia A., (2014:06, "To be published") Master's thesis, Department of Applied Mechanics, Division of Vehicle Engineering and Autonomous Systems, Chalmers University of Technology, Göteborg, Sweden.

3.2 Inputs and outputs of the model

To be developed, the VDM requires some inputs from the Desktop Driving Simulator and then it generates some other information (model outputs).

3.2.1 Inputs

The VDM inputs, listed in *Table 3.1* are mainly related to the driver behaviour and the environmental conditions (especially from the road).

Table 3.1: VDM inputs

INPUT	DESCRIPTION
watchdog	Vehicle model watchdog counter
Reset	Signal to reset the model to original state
Steering wheel angle	Steering wheel position. Positive for left turn [rad]
Throttle position	Throttle pedal position. Values between[0,1], being 1 full throttle
Clutch position	Clutch pedal position. Values between [0,1], being 1 for pedal fully pressed
Gear manual shift position	Gear selected by the driver [1,12], 0 = neutral
Brake pressure at master cylinder	Pressure generated by the driver when braking [0,(inf)] or pos [0,100] [Pa]
Brake pedal position	Brake pedal active
Longitudinal velocity	Maximum longitudinal speed along X-axis [m/s]
Road-tire friction coefficient	Friction coefficient between the tire and the road. Independent for each tire
External forces applied to the centre of gravity	3 components of the external forces applied to the COG, according to the vehicle's coordinate system [N]
External moments applied to the centre of gravity	3 components of the external moments applied to the COG, according to the vehicle's coordinate system [Nm]

Starting from the steering wheel angle, the first seven inputs listed above are the parameters controlled by the driver when using the Desktop Driving Simulator, in the same manner it would be when driving a real vehicle.

Then there is a parameter describing the road design, that is the tire-road friction coefficient.
It can be used to simulate different tire-road contacts, like different asphalt conditions or slippery roads, but also to simulate different driving conditions, like a flat tire, being the friction coefficient defined independently for each tire.

The last two VDM inputs are two vectors containing external forces and moments applied to the vehicle's COG. They are usually used to simulate lateral winds when driving or an impact with other vehicle.

3.2.2 Outputs

The outputs listed in *Table 3.2* are needed for the Desktop Driving Simulator to run an experiment.

Table 3.2: VDM outputs

OUTPUT	DESCRIPTION
watchdog	Vehicle model watchdog counter
IDNR	Identity number of vehicle model
Engine speed	Engine rotational speed, measured in [rad/s]
Engine torque	Torque generated by the engine [Nm]
Steering wheel torque	Torque to be generated in the steering wheel, measured in SI units [Nm]
Simulator reference front sensor	-
Log vector	-
Centre of gravity sensor	-
Front wheel angle	-
Z centre of gravity	Centre of gravity along Z-axle

All these outputs are sent to other subsystems of the Desktop Driving Simulator.

3.2.3 Names of the signals used in the model

A naming standard was used in the model in order to make changes in the simplest way. It includes some accurate rules for creation of signal names. The structure of the names is the following:

Quantity_SourceDescription

where the Quantity indicates what kind of signal it is, for example velocity, acceleration or angle, then after an underscore there is the Source and the Description of the signal. The Source explains where it is used, while the Description is a short description of the signal.

In *Table 3.3* below are listed the quantities used in the Vehicle Dynamics Model.

Table 3.3: List of signals used in Vehicle Dynamics Model

SYMBOL	NAME	UNIT
angle_tb	Torsion bar angle	rad
ax	Longitudinal Acceleration	m/s^2
ay	Lateral Acceleration	m/s^2
beta	Vehicle Side Slip	rad
delta	Steer Angle	rad
dzdx		
dzdy		
Fx_body	Tire Longitudinal Force in Vehicle body coordinate	N
Fx_tire	Tire Longitudinal Force	N
Fy_body	Tire Lateral Force in Vehicle body coordinate system	N
Fy_tire	Tire Lateral Force	N
Fz	Vertical Force/Axle Load	N
Fz_spr_damp	Vertical Force with Spring & Damper	N
GR_tot	Total Gear Ratio	-
LatSlip	Tire Lateral Slip	rad
LongSlip	Tire Longitudinal Slip	rad

Mz	(Self) Aligning Torque	Nm
phi	Roll angle	rad
phi_2dot	Roll Acceleration	rad/s^2
phi_dot	Roll velocity	rad/s
psi	Yaw angle	rad
psi_2dot	Yaw Acceleration	rad/s^2
psi_dot	Yaw velocity	rad/s
SWA	Steering Wheel Angle	rad
teta	Pitch Angle	rad
teta_2dot	Pitch Acceleration	rad/s^2
teta_dot	Pitch velocity	rad/s
Tq_brk	Brake Torque	Nm
Tq_eng	Engine Torque	Nm
Tq_SW	Steering Wheel Torque	Nm
Tq_SW_wo_damp	SWT without damper	Nm
Vx	Longitudinal Velocity	m/s
Vy	Lateral Velocity	m/s
w_eng	Engine Rotational Velocity	rad/s
w_whl	Wheel Rotational Velocity	rad/s
X	Longitudinal Position in global coordinate	m
Y	Lateral Position in global coordinate	m
z		
Zcg	Vertical Position of C.G.	m
Zcg_2dot	Vertical Acceleration of C.G.	m/s^2
Zcg_dot	Vertical Velocity of C.G.	m/s

4 The Braking System

The braking system model is a sub-system of the Vehicle Dynamics Model. It needs to generate the braking torques that allows the VDM to reduce speed and keep vehicle stand-still.

4.1 Brief description of the braking system

Most of vehicles are equipped with discs brakes in the four wheels, except for some small cars that still use drum brakes on the rear axle. A layout of the braking system is shown in *Figure 4.1*, which includes front disc brakes and rear drum brakes.

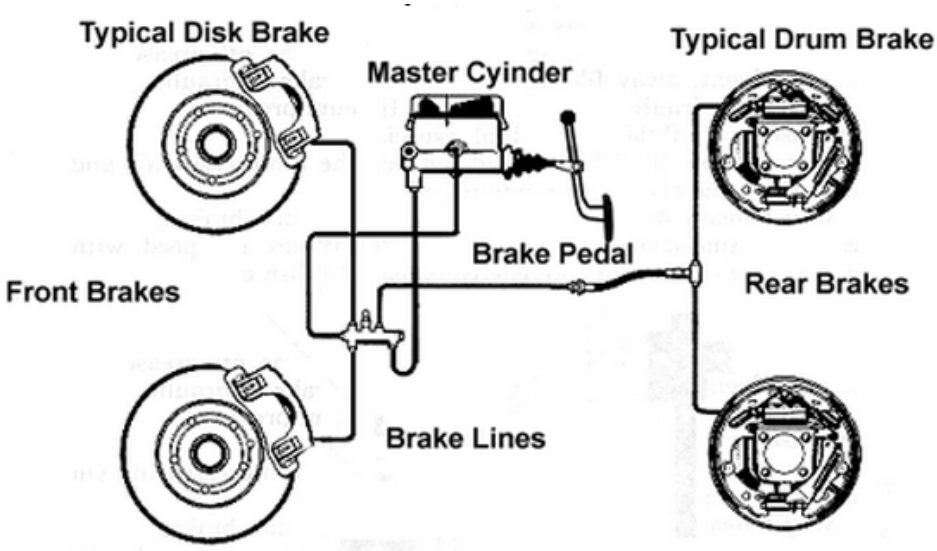


Figure 4.1: Typical braking system with front disc brakes and rear drum brakes

As shown in *Figure 4.1*, the braking system is made up of the following parts: brake pedal, the master cylinder, the brake lines and the braking devices, where there are a rotor (disc or drum) and the friction surfaces (brake pads).
The master cylinder, moved by the driver when he presses the brake pedal, generates an hydraulic pressure. Between the brake pedal and the master cylinder, there is a boost system that increases the brake pressure generation. The brake lines transmit the brake pressure from the master cylinder to the brake callipers that are able to convert the kinetic energy into heat, pressing the friction surfaces against the rotors.

A brake disc system consists of the brake disc, which is attached to the wheel and rotates with it and a brake calliper in a fixed position, installed on the suspension upright. The brake calliper holds the brake pads in touch with the

brake disc and when the brake pressure arises, it presses the pads against the disc generating the braking.

The floating calliper is the most used in the passenger cars. As shown in *Figure 4.2*, only one piston is used to press the pads.

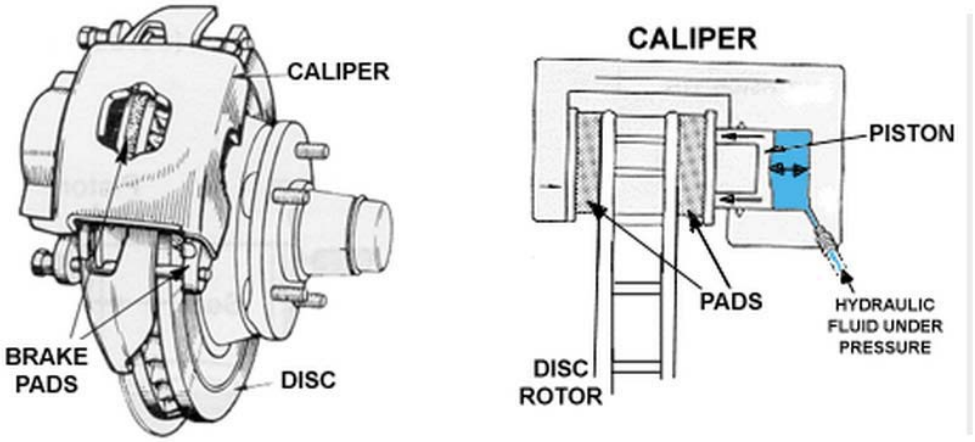


Figure 4.2: Brake disc assembly and floating calliper schematic representation

The braking system model, knowing the pressure at the master cylinder, calculates the braking pressure at each calliper and then, from this one, the braking torque.
The pressure at the master cylinder should be the same of the pressure at the calliper pistons, but actually that does not happen. Indeed, there is a reduction of the vertical load at the rear tires, caused by the load transfer generated from the rear axle to the front axle when braking in a real vehicle. So the rear tires could lock and it is really dangerous, because the motion is unstable.

To prevent rear wheels from locking up and to allow the front axle locking before than the rear, there are some control valves on the hydraulics of the braking system that modulate the pressure in the rear brakes.

4.2 Anti-lock Braking Systems (ABS)

Anti-lock braking system (ABS) has the main goal to prevent wheels from locking up during hard braking, but it also tries to maximize the braking forces generated by the tires. To do that, it prevents the longitudinal slip ratio, which will be defined in Section 4.2.3, from exceeding an optimum value.

It is known that under normal braking conditions, the driver controls the brakes. However, during severe braking or on slippery roadways, when the driver causes the wheels to approach lockup, the anti-lock system takes over. ABS (*Figure 4.3*) modulates the brake line pressure independent of the pedal force, to bring the wheel angular velocity back to the slip level range that is necessary for optimal braking performance. The system shuts down when the vehicle speed is below a pre-set threshold.

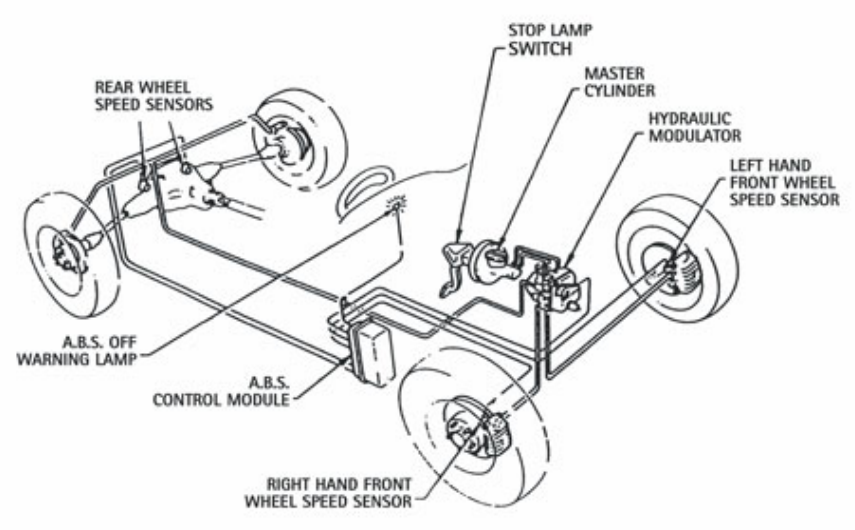


Figure 4.3: Schematic representation of the ABS system mounted on a vehicle

4.2.1 Purposes of the ABS

Three are the main targets of the anti-lock braking systems:

1- to reduce stopping distances
2- to improve stability
3- to improve steerability

Below a brief explanation of these important functions.

Mass of the vehicle, the initial velocity and the braking force influence the **Stopping Distance**. It can be minimized remaining constant the mass of the vehicle and the initial velocity and maximizing the braking force. However, for almost all road surfaces, the frictional coefficient has an optimum value (peak). So, if the wheels of a vehicle are kept near this value, maximum frictional force can be reached and minimum stopping distance as well.

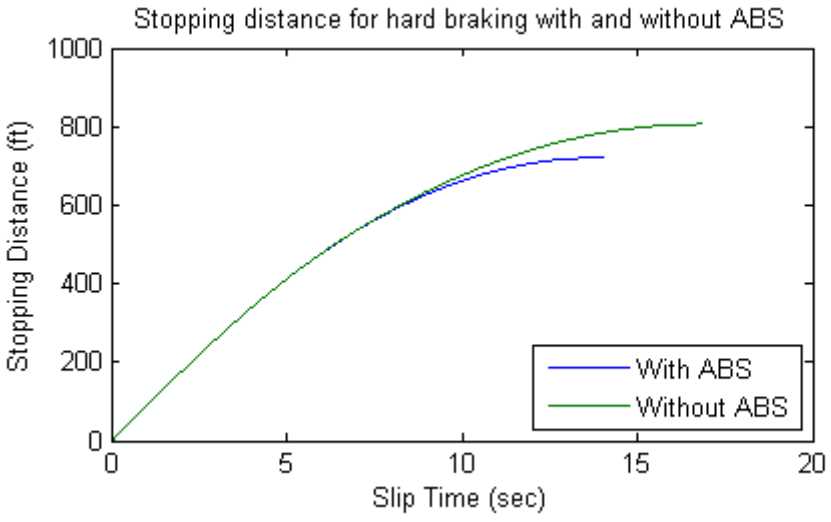


Figure 4.4: Relation between the slip time and the stopping distance

Regarding the **Stability**, is known that even if two main goals of the braking systems are decelerating and stopping, not always maximum frictional force is needed. For example, if the vehicle is on a slippery road, more braking force is obtainable on one side of the vehicle than on the other side. If maximum braking force is applied on both sides, there will be a yaw moment. It will contribute to vehicle instability, but also it will tend to pull the vehicle to the high friction side. So the operator has to make excessive steering corrections to contrast the yaw moment. If the ABS keeps the slip of the rear wheels at the level where the lower of the two friction coefficients peaks, then lateral force is high, even if not maximized and this contributes to stability.

Steerability is important for the possibility of steering around an obstacle during braking and controlling lateral forces.

4.2.2 Components of the ABS

The ABS system consists of the following three main components: wheel speed sensors, a hydraulic modulator and an electronic control unit.

- **Wheel-Speed Sensors**

The speed sensors (*Figure 4.7*) have the purpose to communicate when a wheel is locking up. They are usually located at each wheel.

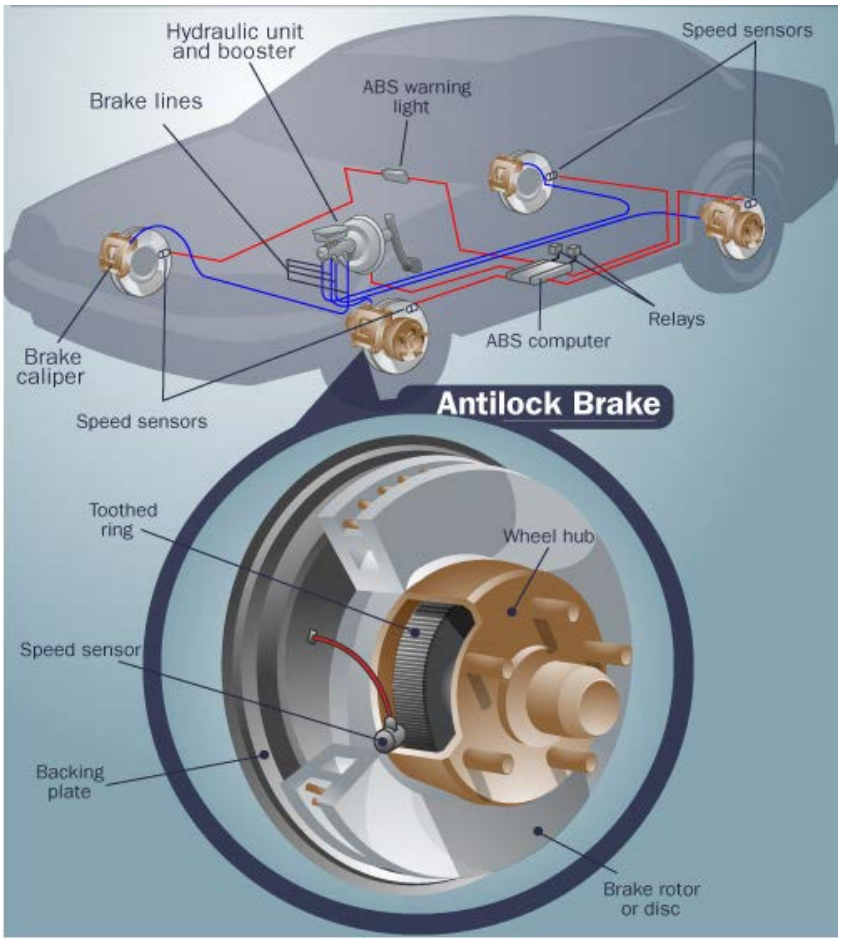


Figure 4.5: Anti-lock brake components

- **Hydraulic Pressure Modulator**

Manipulating the valves in the brake system, this electro-hydraulic device (*Figure 4.5*) is able to reduce, hold and restore the pressure of the wheel brakes.

The valve usually has three positions:

- In position one, the valve is open; pressure comes to the brake from the master cylinder. So the driver controls the brake, allowing the amount of brake pressure to the desired by the driver to be applied to the brake.

- In position two, the valve blocks the line, the brake is isolated from the master cylinder, because the valve blocks the line. In this way, if the driver presses the brake pedal harder, the pressure does not increase.

- In position three, the pressure from the brake is released from the valve.

The hydraulic pressure modulator is placed in the engine compartment. In this way the length of the lines to the brake master cylinder are minimized. Moreover, the hydraulic modulator very often includes a pump, an accumulator and a reservoir.

Particularly, the pump has the function to put the pressure, that the valve releases from the brakes, back; when a valve reduces the pressure in a line, the pump is there to get the pressure back up (*Figure 4.6*).

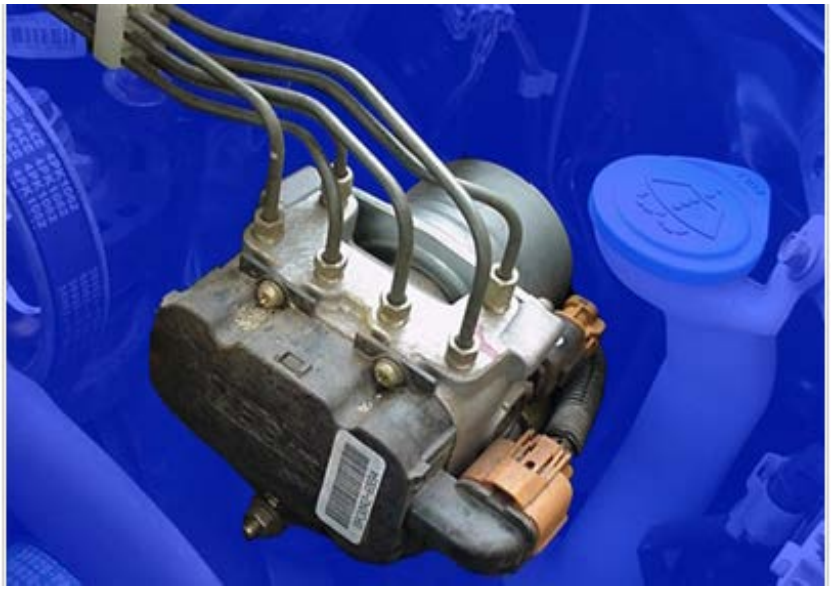


Figure 4.6: Anti-lock brake pump and valve

- **Electronic Control Unit (ECU)**

It calculates the wheel rotational speed and the acceleration receiving, amplifying and filtering the sensor signals. It also estimates the speed of the vehicle, using the speeds of two diagonally opposed wheels. Comparing this reference speed with the speeds of each wheel, is possible to obtain the slip at each wheel. These signals need to avoid the locking, alerting the ECU.

The latter sends a signal to trigger the pressure control valve solenoids of the pressure modulator to modulate the brake pressure.

Figure 4.7: Wheel-speed sensor with pulse ring

Moreover, the ECU is also able to control the errors in the ABS system, using wheel-speed sensors, the ECU itself, pressure-control valves, shutting down the whole ABS.

4.2.3 Friction coefficient and slip ratio in the ABS system

The ABS system is influenced by:

- ✓ the value of the tire-road friction coefficient, related to the wheel slip ratio
- ✓ the initial longitudinal velocity of the vehicle
- ✓ the brake effort distribution from front to rear
- ✓ the rate of application of the brake torque

The first factor is now analyzed in depth.
As has been said in Section 4.2.1, the slip is one of the most important parameters that concerns ABS.

For convenience a slip ratio is defined according to:

$$\lambda = \frac{V_X - \omega R}{V_X}$$

The nomenclature in above equation is presented as follows:

V_X = linear velocity of a wheel of the vehicle

ω = rotational speed of a wheel of the vehicle

R = radius of the wheel

$\boldsymbol{\lambda}$ = slip ratio

The vehicle steerability and stability can be explained with the relation between the tire-road friction coefficient and wheel slip ratio. Depending on some factors, like the road surface type conditions (dry or wet), the tire side-slip angle, the vehicle speed, the slip ratio between tire and road, the friction coefficient changes in a very wide range, but it usually has an optimum value at

a particular value of the wheel slip ratio. This value depends on the road surface.

As is easy to understand looking at *Figure 4.8*, for almost all type surfaces, the tire-road friction value is optimum when the wheel slip ratio is approximately 0.2 and worst when the wheel slip ratio is 1, that means the wheels are locked. The optimum slip value decreases as tire-road friction decreases.

So, goal of the ABS system is to regulate the wheel slip ratio to target value of 0.2.

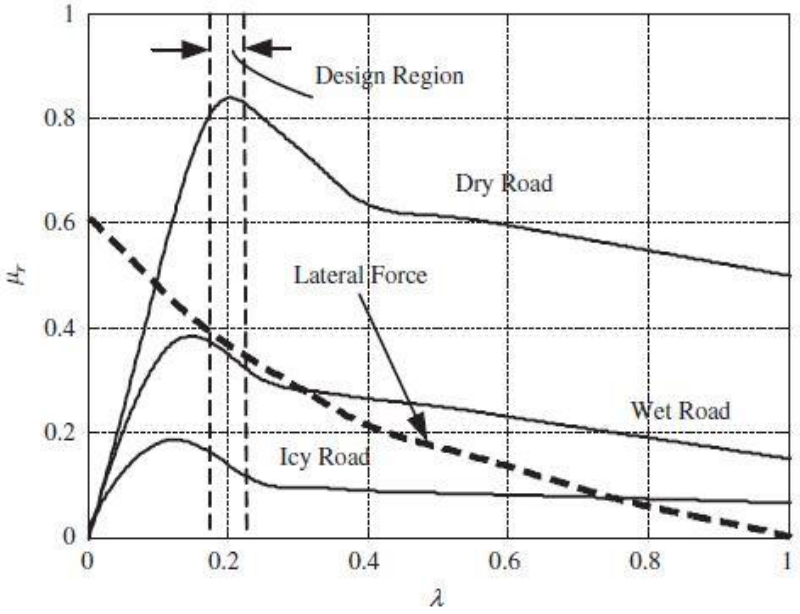


Figure 4.8: Frictional coefficient of road surface vs. Wheel slip ratio

4.2.4 How the ABS works

During all the time, the ECU controls the speed sensors searching for decelerations in the wheels that are not included in the normal braking. It will experiment a quick deceleration before a wheel locks up.

Such as a deceleration is impossible, the ECU performs a really fast procedure: it decreases the pressure to that brake until it sees an acceleration, then it increases the pressure until it sees the deceleration again. At the end, the tire slows down at the same rate as the car, with the brakes keeping the tires very near the point at which they will start to lock up. This gives the system maximum braking power.

4.3 Anti-lock braking system model

To be able to run a simulation with an ABS-equipped vehicle and study the braking performance, an ABS system was modelled. The information about the ABS system were found in Benito G. and Nilsson H., (2006): *Vehicle Stability Control for Roadside Departure Incidents by Steering Wheel Torque Superposition*, Master's thesis, Departments of Applied Mechanics & Signals

and Systems, Chalmers University of Technology, Göteborg, Sweden and in the MATLAB / Simulink tutorial files. Their thesis was a sort of vehicle model documentation about the S2, Chalmers Driving Simulator.

Particularly, the ABS subsystem in Simulink included in the S2 vehicle model was fitted in the original Vehicle Dynamics Model of this thesis. This was a non-trivial and time consuming task (*Figure 4.9*).
The file with the appropriate changes has been called in Simulink as *Simulink_car_simulator_ESC.*

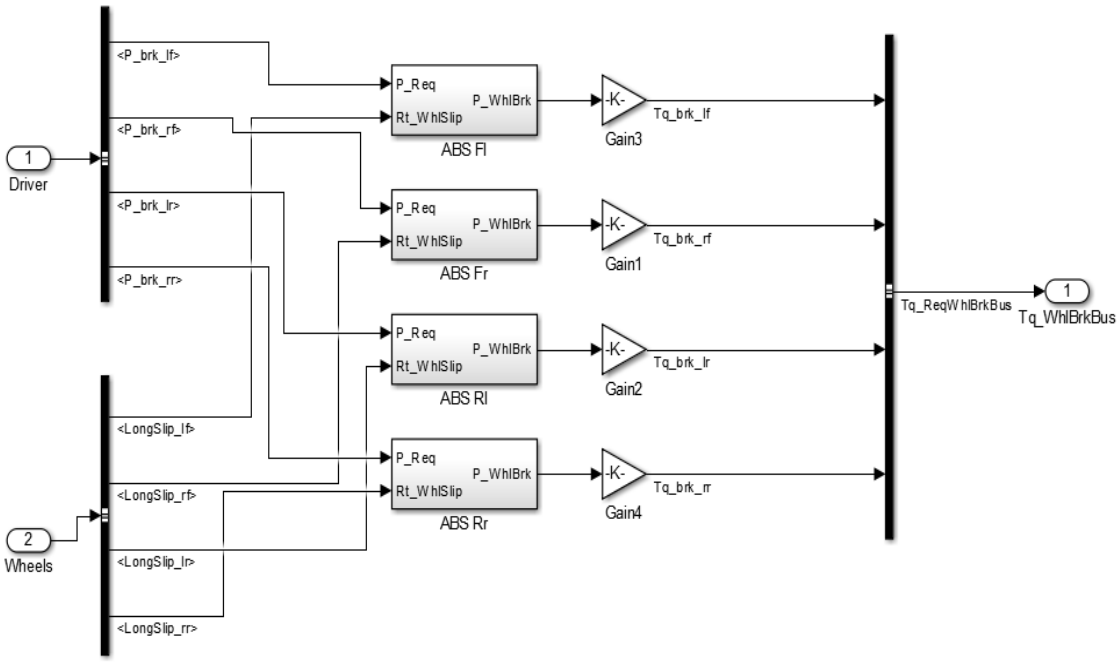


Figure 4.9: ABS Simulink subsystem in the Vehicle Dynamics Model

As shown in *Figure 4.9*, there are two main inputs: from the driver and from the wheels. From the driver there are the four braking maximum forces, or better the four pressures of the four wheels and from the wheels the four longitudinal slips of the four wheels. These inputs (the requested pressures and the wheel slip ratios) come to the four subsystem of the four wheels , the four ABS subsystem of each tire, and they generate as input the pressures. The gains amplify these signals and the final requested brake torques are obtained.

Try now to analyze the ABS Simulink model for one tire.
Within the model, there is a PI (proportional-integral) controller that is able to keep the slip ratio in a limited range when the brakes are applied. In this range, the tire creates some lateral force assuring the steerability of the car. The PI controller is fed by the difference in real slip and the desired slip value. The value used for the desired slip was 0.15. So, as stated in *Section 4.2*, if the slip exceeds this value, the pressure to the brake and accordingly the torque, is decreased until the slip is below the desired value and then the brake pressure is increased again over 0.15. In this way, the slip is held oscillating close to the desired slip rate value.

The ABS model used in the Desktop Driving Simulator, knowing the vehicle data, the tire-road interaction and the brake torque, calculates the wheel speed and the slip, comparing this value with the desired value and executing the control.
In the model has been hypothesized that the vehicle does not perform curves.
The ABS Simulink model for one tire is shown in *Figure 4.10*.

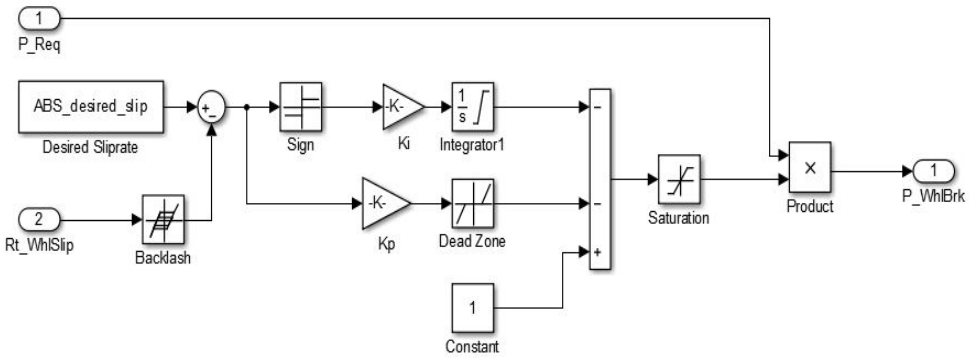


Figure 4.10: The ABS Simulink model for one tire

In *Table 4.1* below, are shown some values used in the model for the Simulink model, together with their units.

Table 4.1: Values of some variables used in the ABS Simulink model

NAME	VALUE
desired_slip	0.15
Ki	2
Kp	10
backlash	0.01
brake_max_torque	69038.8
brake_distr_front	0.7
brake_distr_rear	0.3
dead_zone(start)	-10000
dead_zone(end)	0.4

5 Offline and online simulations

In this chapter the offline and online simulations and their results will be presented.

The simulations are one part of a model validation.
The validation of a Vehicle Dynamics Model is a non-trivial and consuming task and consists of obtaining adequate and accurate data, representative of the real vehicle behaviour and comparing these data with the model behaviour. The differences between the values can be minimized by tuning the model response, through parameter optimization.
As the VDM is one part of the Desktop Driving Simulator system, it needs to work properly as a part of the whole simulator. For this reason, the model validation will also include some simulator experiments.

Due to the lack of resources and time, VDM validation was not performed, so this section concerns only the simulations part. The offline simulations are obtained only using MATLAB and Simulink, while the online simulations are performed driving the Desktop Driving Simulator. So, the main goals of the experiments will be to compare the online simulations versus the offline ones and make some considerations on driver behaviour and influence.

5.1 Description of the experiments

The experiments developed and performed are based on the ISO standards. An automatic transmission was used during the tests (no gearstick, no clutch). They can be divided into open-loop and closed-loop tests, where in the first one there is no driver's influence, while the closed-loop test takes into account the driver's behaviour.

In *Table 5.1* below are shown some conditions of the experiments: for all of these, the model is parameterized with respect to a Volvo S40; the road surface is asphalt and it is completely flat, since it is hard and time consuming to reproduce the test track surface asperities. The transmission layout used is FWD (Front Wheel Drive), where the engine drives the front wheels only.

Table 5.1: Some conditions of the experiments

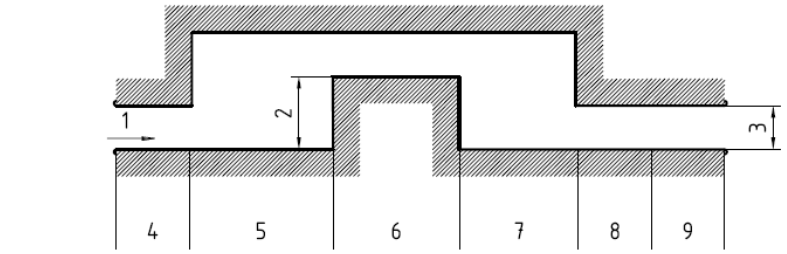
	Straight Ahead Acceleration (Offline & Online simulations)	Straight-Line Braking (Online simulations)	Double-Lane Change (Online simulations)
Vehicle	Volvo S40	Volvo S40	Volvo S40
Road surface	Asphalt	Asphalt	Asphalt
Road profile	Flat	Flat	Flat
Drive system	FWD	FWD	FWD
ABS	-	OFF/ON	ON
Speed	0 - 150 kph	100 – 0 kph	(80 ± 3) kph

The first open-loop experiment (**Straight Ahead Acceleration**) consists on driving the vehicle in a straight line and reaching the maximum speed (in this case 150 kph) in a short time.

The second one (**Straight-Line Braking**) consists on driving the vehicle in a straight line and performing a full braking manoeuvre in a short time. From this test, the braking performance of the model can be studied (only online simulations)
In this case, was tested this manoeuvre also with an ABS-equipped vehicle, starting braking when the speed reached was around 100 kph.

Regarding the closed-loop test, a basic case, that is a **Double-Lane Change** manoeuvre, for verification of the model has been performed. This manoeuvre is usually used to test vehicle handling and behaviour in extreme conditions. In this manoeuvre, the driver must follow a cone track describing the double-lane change at a predefined speed (80 ± 3 kph), without touching any cone. The lengths of track sections are fixed, while the width of the entry and side lane are dependent on the width of the vehicle, that in this case is 1.78 m.. From this experiment, the driver feeling and perception of the manoeuvre can be studied.

Figure 5.1 shows a Double-Lane Change track, as defined in the ISO standards (*Table 5.2*).



Key
1 Driving direction
2 Lane offset
3 Width
4 Section 1
5 Section 2
6 Section 3
7 Section 4
8 Section 5
9 Section 6

Figure 5.1: Double-Lane Change track and designations of sections

Table 5.1: Dimensions of the Double-Lane Change track

Dimensions in metres

Section	Length	Lane offset	Width
1	15	—	1,1 × vehicle width + 0,25
2	30	—	—
3	25	3,5	1,2 × vehicle width + 0,25
4	25	—	—
5	15	—	1,3 × vehicle width + 0,25
6	15	—	1,3 × vehicle width + 0,25

5.2 Input parameter data

Starting from *Table C.1* to *Table C.13* described in Appendix C, all the parameters used in the VDM are defined, including values and units. The passenger vehicle model is parameterized with respect to a Volvo S40 (1.6 ton.), shown in *Figure 5.2*.

Figure 5.2: Volvo S40

It is possible to refer to the .m file "*VehicleData_tool.m*": it is a MATLAB file, where there are all the input vehicle data used in the model.

5.3 Offline simulations

To run the offline simulations, the original Vehicle Dynamics Model provided and developed by VTI, was fitted to the so-called Karlsson A.'s evaluation tool (*Simulation_tool_v1* folder on the Simulator PC's desktop). Regarding that and the general structure of the offline simulations, more detailed information can be found in Karlsson A., (2014): "*Chalmers Vehicle Verification Process*" Master's Thesis, Department of Applied Mechanics, Division of Vehicle Engineering and Autonomous Systems, Chalmers University of Technology, Göteborg and Karsolia A., (2014:06, "To be published"), Master's thesis, Department of Applied Mechanics, Division of Vehicle Engineering and

Autonomous Systems, Chalmers University of Technology, Göteborg, Sweden as well.

So, the "*NewModel.slx*" was loaded in Simulink and afterwards, the .m file "*main_simulation_script.m*" in MATLAB R2014-a, keeping attention typing number [2] in the line "*settings.tests_to_run*".

The results of the offline simulations are shown in Appendix E for the different experiments studied. In Section 5.5 a brief discussion of the results obtained will be done.

5.4 Online experiments

The Vehicle Dynamics Model is a part of a bigger system, the Desktop Driving Simulator. After that the VDM has been simulated in a standalone computer, it has been downloaded to the Desktop Driving Simulator and used in real time. Some problems, especially at the beginning, appeared, so a long debugging process has been done to remove all these defects and get the model running in the Desktop Driving Simulator in the best possible way.

So some online simulations have been performed, keeping always in mind the realism and that is not possible to analyse the VDM behaviour in the driving simulator without consider the surrounding systems.

For the online simulations of the first open-loop test, the **Straight Ahead Acceleration**, the files provided have been loaded, that is the Simulink file "*simulink_car.mdl*" and the following MATLAB files, "*VehicleData.m*" and "*vti_def.m*", where there are the parameters used for the interface with the simulator.

For the online simulations in the case of the **Straight-Line Braking**, the files loaded have been the modified Simulink file, that is "*Simulink_car_simulator_ESC*", which include the ABS system and the specific .m files, "*vti_def.m*" and "*VehicleData.m*", which incorporates "*VehicleControl.m*", where there are the functions of ESC and ABS.

Running the simulation with the ABS-equipped vehicle, is important to remember to switch off the ESC (Electronic Stability Control) and CC (Cruise Control) in the MATLAB file "*VehicleControl.m*".

To log data, before running the simulation, one needs to go on the left of the Qt Creator User Interface, go to *Projects* and then select *Run.* So, in *Arguments* type: "*-p astazerosimchalmers –d j –S –f 1 –v 127.0.0.1 5004 –a"*.
For converting the log files (*.bin*) to float arrays which can be read in MATLAB, a conversion program was added and some MATLAB scripts for reading the converted log files as well.
So, after that the simulation is terminated, the project "*conv.pro"* in Qt Creator has to be open. The program is located in:

Computer/LocalDisk(C:)/astazerosimchalmers/project/astazerosimchalmers/conv/conv.pro

It is enough set "*conv*" as *Active Project* and select *Run.*
The logged data (see also Appendix D) can be found in:

Computer/LocalDisk(C:)/astazerosimchalmers/project/astazerosimchalmers/data

The results of the online simulations are shown in Appendix E for the different experiments studied. In Section 5.5 a brief discussion of the results achieved will be done.

5.4.1 Double-Lane Change test

This experiment has been performed in accordance to the International Standards Organization (ISO), as described in ISO 3888-1.
This part of the experiment has been done in a straight and flat road, where three sets of cones appear describing three double-lane change manoeuvres. One of the tasks of this test is studying the lateral response and the steering wheel force feedback.
Four different drivers drove the simulator and participated to the experiment.

5.4.1.1 Performing the DLC manoeuvre

To execute this test is recommended a speed of (80 ± 3) kph. The cones track starts in the right lane of the road, changes to the left side and comes back again to the right, so the driver should change to right lane when starting driving. Since there are three sets of cones, is convenient drive through the first set at slow speed and get used to the cone track and then drive through the other two sets of cones at a speed between 75 and 85 kph.
The throttle position during the manoeuvre should be held constant.

It is important point out that the vehicle must be kept in the road all the time. If the car goes out of the road, the simulation stops in a sharp way.

To perform this kind of experiment, a cone track event was added.
Following the process described in Appendix B, one has to open this event in Qt Creator, whose file is in:

Computer/LocalDisk(C:)/astazerosimchalmers/project/astazerosimchalmers/conetrackevents.cpp

```
LaneChangeIso3888_1::LaneChangeIso3888_1(int id, const std::string& tag) :
        ConeTrackBase(id, tag, 18)
{
}

void LaneChangeIso3888_1::init() {
        if(DynamicRoadActorPtr ove = qSharePointerDynamicsCast<DynamicRoadActor>(m_ove)) {
                OpenDrivrPosition p0(ove->getPosition() + 150);
                p0.setR(0);

                double cone_w = m_cones[0]->getShape().getWidth();

                double ove_w = ove->getShape().getWidth();

                double A = 1.1 * ove_w + 0.25 + cone_w;
                double B = 1.2 * ove_w + 0.25 + cone_w;
                double C = 1.3 * ove_w + 0.25 + cone_w;
```

Figure 5.3: Double-Lane Change event in conetrackevents.cpp

set up the event as suggested in Appendix B and then is also possible to set up the timer (how long after the cones appear) and the distance between the cones and the sets of cones, as shown in *Figure 5.3.*

5.5 Discussion

A discussion about the simulations results can be found in this section, but only the most significant results are described, for sake of brevity. The aim of this discussion is try to explain the plots obtained for a better understanding of the reader and to determine if the desired characteristics of the model have been fulfilled.

In Appendix E are shown the results obtained, for both the open-loop tests and close-loop test in different conditions. In the experiments, some variables like imperfections of the roads, wind, unbalanced characteristics in the vehicle are approximately considered constant.

Straight Ahead Acceleration

Regarding the Straight Ahead Acceleration maneuver, a comparison among offline simulations (using MATLAB and Simulink) and online simulations (driving Desktop Driving Simulator) has been done. The main variables to study are the vehicle longitudinal velocity and the vehicle longitudinal acceleration.

For values of longitudinal velocity (*Figures E.1* and *E.11*) from 5 kph up to 30 kph, the online simulation matches well the offline one and the plots are roughly equals. But at the beginning, in the online simulation there is an initial delay. The latter can be explained by the fact that when someone drives the Desktop Driving Simulator, has to press the brake pedal before starting the simulation and then, after few seconds, he can start driving. It is a bug of the Desktop Driving Simulator, that should be studied in deeper.
In the offline simulation, the brake is due to too low longitudinal acceleration.

The longitudinal acceleration (*Figures E.2* and *E.12*) maybe is the most interesting variable in this basic case. The graph in the online simulation is quite similar to the plot of the offline one if this one is shifted right of 5 seconds. Also here there is the initial delay, but then, between the time of 2.5 sec. and 5 sec., there is a plateau and at a time equal to 6 sec. can be seen a dip. Probably, they depend on some interference driving the simulator or in the logged data.

The tire longitudinal traction force (*Figures E.9* and *E.13*) for the four wheels (Front Left, Front Right, Rear Left and Rear Right) and the pitch angle (*Figures E.7* and *E.14*) in both the offline and online simulations, present a very good match. The online ones show, as in previous, some initial small discrepancies, that reach their maximum in a range between 5 and 6 sec.. Regarding the pitch angle in the online simulation, it can be noticed that it is really small (less of -1.5°) and it has a negative value, because it depends on the coordinate system of the vehicle. Moreover, the curve is not too smooth, maybe due to the number of points considered not too high.

In Appendix E can be found more other results for the offline simulation for some variables, such as side slip angle and yaw velocity.

Double-Lane Change

The Double-Lane Change maneuver was performed by four different drivers with an ABS-equipped vehicle and without ESC system. No offline test has been performed for this case. From this experiment, the driver behavior and perception can be studied.

The characteristic plotted in *Figure E.15* shows the route executed by the drivers. The position of the cones, according to the ISO 3888-1, is marked by a circle. Only the second driver had never driven the simulator, instead the other participants to the test was not the first time they used it. Despite this, it seems that the cones are run over by all the drivers. There are mainly two reasons for that. It could be due to the fact that is not possible to look at the dimensions of the vehicle during driving the simulator, but it also is very likely that most drivers have difficulties with this cone track, performing a Double-Lane Change maneuver at 80 kph in a real vehicle. So that they have difficulties in the simulator is expected, assuming Desktop Driving Simulator is more difficult, since drivers have not correct feedback through seat acceleration. Hence, indications that desktop gives somewhat realistic vehicle maneuver for such maneuvers. Possibly, the Desktop Driving Simulator is a little bit too difficult to handle compared to real vehicle.
Last consideration that can be done is that in all the tests, the drivers understeer due to saturation tire forces in the front axle and they oversteer, especially during the counter steering phase.
An interesting variable in this test is the side slip angle. *Figure E.16* shows this angle as function of time for the four wheels of the vehicle (FL, FR, RL, RR) for each single driver. Comparing the plots, it can be noticed that the third driver drove harder than the others, because the peak is higher, instead the fourth performed a milder maneuver, but he turned sooner (understeering).
However, the value of the side slip angle is really small (less of 1 degree), so there are not too many problems to handle it and the result is satisfactory.

Anyway, the evaluation of all tests demonstrate that, even if the test was developed for testing lateral dynamics, it was found that the longitudinal dynamics has a strong influence, which explains a considerable amount of scatter in the data and thus reflects on the results.

Straight-Line Braking

The main goal of the Straight-Line Braking test is studying the performance of the braking system, but can also be evaluate the vertical dynamics in accelerating and braking maneuvers, even if here the same considerations done above are appropriated.

Figure E.17 shows the vehicle longitudinal acceleration, which is generated by the PID controllers. The ABS curve (in red) is in delay than the no-ABS curve, because ABS must have the time to activate, but however the two curves in the plot are too similar, so maybe the result is not so reasonable.

It is possible to study that in a better way looking at *Figure E.18*, where is shown the longitudinal slip for the Front Left and Front Right, which is a very important factor during a full brake maneuver from 100 kph to full stop, with and without ABS system. For a clearer understanding, the scale on the X-axis has been changed, to underline the braking performance. Hence, a zoom starting from the time of 19 sec. until 22 sec. has been done. It can be noticed that the value of the longitudinal slip is too low (around 0.02), it should be much higher. So maybe the brakes were not applied enough and ABS did not activate at all, probably because the input signals in the model were too small. Actually, it is obvious that ABS does something, but it is quite evident that it does not influence brake performance a lot. The reason of that could depend on tire model. If tire model does not have a peak, it could still be correct: that ABS limit slip, but with this tire model, it does not see the disadvantage with locking, since friction does not go down after peak. It is known that the tire-road friction value is optimum when the longitudinal slip is approximately 0.2; instead here the values are too low and so the ABS is not working well. Finally, in case of full braking the final value should be -1 (-100%), that means locked wheels, instead in the plot is around -0.05. Probably, the gains which have the function to amplify the signals in the ABS Simulink model, are not enough, they are too low.
The last characteristic plotted in *Figure E.19* shows the wheels rotational speed, obtained considering a wheel radius of 0.3m. The result is not so reasonable, mainly because the two curves are roughly equal and that means ABS was not activate at all.

So this maneuver was found hard to get any accurate and rational results.

Unfortunately, it was not possible to repeat the test experiments many other times, so the results are not so reliable.

6 Conclusions

This small chapter includes some main conclusions.

Operate with the driving simulator's code it is not simple, especially because there is not a documentation or something like that. Chapter 2 tries to address this issue or at least to explain how to run a simulation. Moreover, it requires C++ programming skills and a good knowledge of the software. There are many files and to implement new functionalities, is necessary manage different of them, also at the same time.

The offline and online simulations performed are roughly equal, show the same behavior, except for the initial delay and some interference driving the simulator.

The Double-Lane Change maneuver approximates the behavior of a vehicle in the case where the driver needs to quickly switch from one lane to the other and back. It proves the agility and capabilities of the vehicle in both longitudinal and lateral dynamics and also gives some indications about drivers feelings and perceptions.
The test is quite mild, even if drivers have not correct feedback through seat acceleration.

For the Straight-Line Braking was found hard to get any accurate and rational results. The plots show that the ABS system is not activate at all, probably because the input signals to the brake in the model are too low. So a deeper study about that, in particular for the tire model, is required.

Moreover, the test that were performed can be used in the ASTAZero Track.

Anyhow, the thesis can be considered a very stable base for further model tuning and future tests. With its modular structure and small manual, it can be considered as a good starting point for future improvement of the whole simulator system.

7 Future work

This last chapter includes some recommendations and suggestions for future work.

- Using only one computer running both simulations and the Vehicle Dynamics Model.
- Deeper study and validation of the model with real vehicle data and also with different vehicles.
- Keep much more attention to the driver's feelings and perceptions for the steering wheel feedback.
- Improve the VTI's code, in order to make it easier to exchange variables among the PCs.
- Make the VTI's code easier to operate with it without many difficulties.
- Improve VISIR program and create new different driving scenarios, such as a rainy or night scenario.
- Update the sound system to create more realistic sound cues.
- Increase of the amount of the experiments.
- Test maneuver that completely activates the Anti-lock braking system.
- Investigate deeper the tire model.
- Find the sources of the signal errors.

8 References

[1] Jacobson B., (2013): *Vehicle Dynamics, Compendium of the Course MMF062*. Division of Vehicle Dynamics and Autonomous Systems, Chalmers University of Technology, Göteborg, Sweden.

[2] Karsolia A., (2014,"To be published"): Master's thesis, Department of Applied Mechanics, Division of Vehicle Engineering and Autonomous Systems, Chalmers University of Technology, Göteborg, Sweden.

[3] Nordmark S., (1984): *Mathematical Model of a Four-Wheeled Vehicle for Simulation in Real Time*. VTI Report, publication n° 267 A 1984, Linköping, Sweden.

[4] Gómez Fernández J., (2012): *A Vehicle Dynamics Model for Driving Simulators*. Master's thesis, Department of Applied Mechanics, Division of Vehicle Engineering and Autonomous Systems, Chalmers University of Technology, Göteborg, Sweden.

[5] Obialero E., (2013): *A Refined Vehicle Dynamic Model for Driving Simulators.* Master's Thesis in Automotive Engineering, Department of Applied Mechanics, Division of Vehicle Engineering and Autonomous Systems, Chalmers University of Technology, Göteborg, Sweden.

[6] Augusto B. and Loureiro R. J. L., (2009): *Motion Cueing in the Chalmers Driving Simulator: A Model Predictive Control Approach*, Master of Science thesis, Department of Signals and Systems, Division of Automatic Control, Automation and Mechatronics, Chalmers University of Technology, Göteborg, Sweden.

[7] Kiril Z. Rangelov, (2004): *SIMULINIK Model of a Quarter-Vehicle with an Anti-lock Braking System,* Department of Mechatronic Design, Stan Ackermans Instituut, Eindhoven University of Technology, Netherlands.

[8] Morando A., (2014): *Development and Improvement of the Chalmers' Driving Simulator*, Department of Signals and Systems, Division of Automatic Control, Automation and Mechatronics, Chalmers University of Technology, Göteborg, Sweden.

[9] Andersson A., (2009): *Implementation, validation and evaluation of an ESC system during a side impact using an advanced driving simulator,* Department of Electrical Engineering, Linköping, Sweden.

[10] Kutluay E. and Winner H., (2012): *Assessment Methodology for Validation of Vehicle Dynamics Simulations using Double Lane Change Maneuver*, Technische Universität Darmstadt, Germany.

[11] Gillespie, T.D: *Fundamentals of Vehicle Dynamics*, ISBN 1-56091-199-9.

[12] Bharat Bhivate P., (2010): *Modelling & Development of Anti-lock Braking System*, Bachelor of Technology in Mechanical Engineering, National Institute of Technology, Rourkela 769008, India.

[13] H. Mirzaeinejad, M. Mirzaei, (2010): *A novel method for non-linear control of wheel slip in anti-lock braking systems*, Control Engineering Practice vol. 18, pp. 918–926.

[14] Benito G. and Nilsson H., (2006): *Vehicle Stability Control for Roadside Departure Incidents by Steering Wheel Torque Superposition, Developed for and Evaluated in the Chalmers Driving Simulator*, Master's Thesis, Departments of Applied Mechanics & Signals and Systems, Chalmers University of Technology, Göteborg, Sweden.

[15] ISO 3888-1 (1999): *Road Vehicles. Passenger cars. Test Track for a Severe Lane-Change Manoeuvre. Double Lane-Change.*

[16] Pacejka H.B., (2005): *Tire and Vehicle Dynamics*, 2nd Edition, SAE International, Englewood Cliffs, New Jersey, USA.

[17] Rajamani R., (2011): *Vehicle Dynamics and Control*, Second Edition, ISBN 978-1461414322, Springer, University of Minnesota, USA.

[18] Salaani, M.K., Heydinger, G.J. and Grygier, P.A., (2002): *Modelling and Implementation of Steering System Feedback for the National Advanced Driving Simulator*, SAE 2002-01-1573, SAE Technical Paper Series.

[19] BOSCH Automotive Engineering Handbook, 6th Edition. Bosch Gmbh.

[20] Fischer M., Sehammar H., Nilsson M., Lazic N., Weiefors H., (2011): *Advanced Driving Simulators as a tool in early Development phases of new Active Safety Functions*, Submitted to the 3rd International Conference on Road Safety and Simulation, Indianapolis, USA.

[21] [Online]: http://www.vti.se/en/research-areas/vehicle-technology/ driving-simulation/simulator-technology/

[22] [Online]: http://www.mdm.unifi.it/upload/sub/dispense/Rindi/Autoveicolo.pdf

[23] [Online]: http://www.multibody.net/teaching/msms/students-projects-2011/antilock-braking-system-abs/

[24] [Online]: http://www.mathworks.it/it/help/simulink/examples/modeling-an-anti-lock-braking-system.html

[25] [Online]: http://qt-project.org/doc/qt-5/gettingstarted.html

9 Appendices

Appendix A. Vehicle Dynamics theory

In this appendix are presented some basic vehicle dynamics concepts used in this master project, such as systems of coordinates, angles and rotations.

System of coordinates

The coordinate system (*Figure A.1*), used in this thesis, is assumed as the reference for the position of the vehicle.

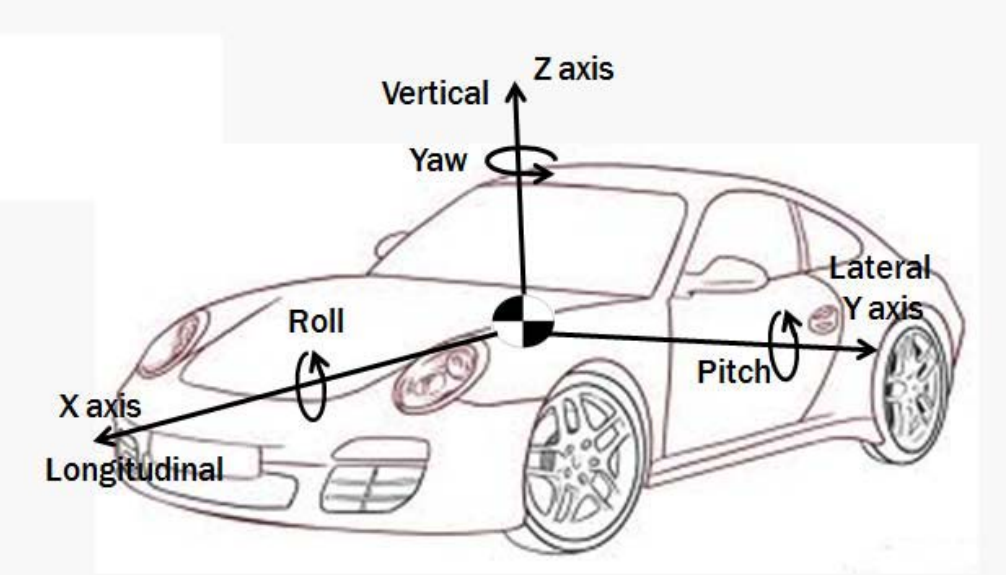


Figure A.1: System of coordinates fixed to vehicle's CoG

This system is in accordance to the International Standards Organization (ISO), as described in ISO 8855. Using it, the forward (longitudinal) movement of the vehicle is represented in the positive X-axis, instead the lateral movement is described by the Y-axis, being positive when oriented to the left (from the driver position) and the vertical movement is described in the Z-axis.

In addition to this coordinate system, a local coordinate system is used independently for each tire, always according to ISO 8855. The origin is located in the centre of each tire and the X-axis follows the heading of the wheel.

In *Figure A.2* is shown the coordinate system for a single wheel.

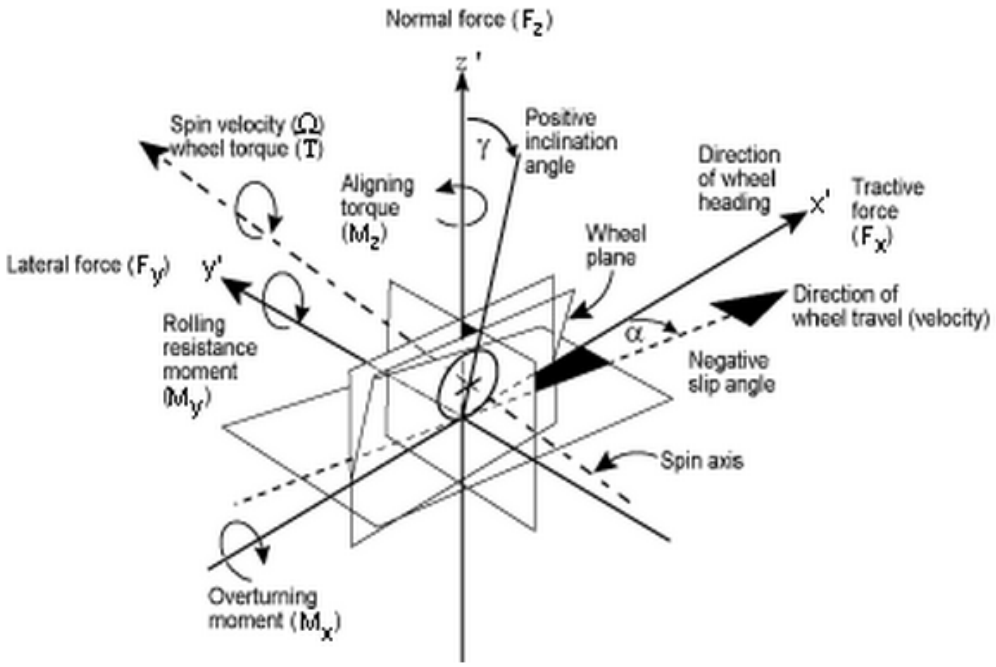


Figure A.2: Wheel local coordinate system, as defined in ISO 8855

Roll, pitch and yaw rotations

Roll, pitch and yaw are the rotations about the three axes of the centre of gravity (CoG) coordinate system. All three rotations are shown in previous *Figure A.1*.

Short description of these rotations:

• Roll rotation is defined around the X-axis (longitudinal). This rotation can be felt during lateral acceleration, for example during a turn or a lane change.

• Pitch is the rotation about the Y-axis (lateral). Pitch can be felt during braking or accelerating.

• Yaw rotation is defined around the Z-axis (vertical). It can be felt during cornering or slipping.

For a better understanding, some other vehicle dynamics concepts are briefly explained and they can be understood looking at *Figure A.3*.

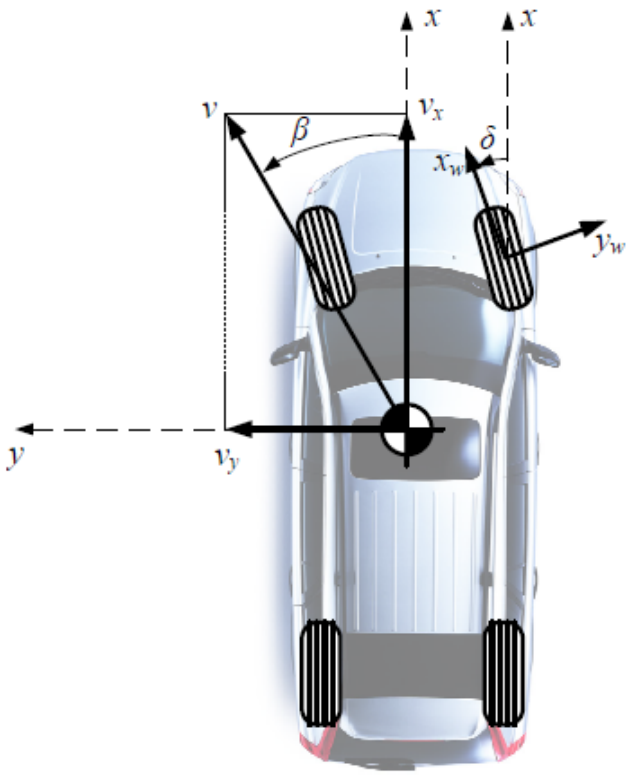


Figure A.3: Angles in the coordinate system

- Side slip angle

  The side slip angle, β, is the angle between a rolling wheel's actual direction of travel and the direction towards which it is pointing. This angle also allows to understand how much the vehicle is sliding, so it is important in the active safety area.

- Course angle

  The course angle is the angle between the direction to a fixed reference object and the tangent line to the path over the ground the vehicle wants to follow.

- Steering wheel angle

  The steering wheel angle, δ, is the angle between the X-axis of the wheel and the X-axis of the centre of gravity coordinate system of the vehicle.

  Under-steering
  The term under-steering is defined as: the vehicle develops less side slip angle than intended by the driver when turning the steering wheel. This means that the car turns less than intended.

  Over-steering
  The term over-steering is defined as: the vehicle develops more side slip angle than intended by the driver when turning the steering wheel. This means that the car turns more than intended.

Appendix B. How to set up Qt Creator

In this appendix is explained how to create and run a Qt Creator project. The goal is to provide to the user the instructions to start working with the software of the Desktop Driving Simulator in the fastest and easiest way.

Qt Creator User Interface

When Qt Creator is launched, appears the following Qt Creator User Interface (*Figure B.1*):

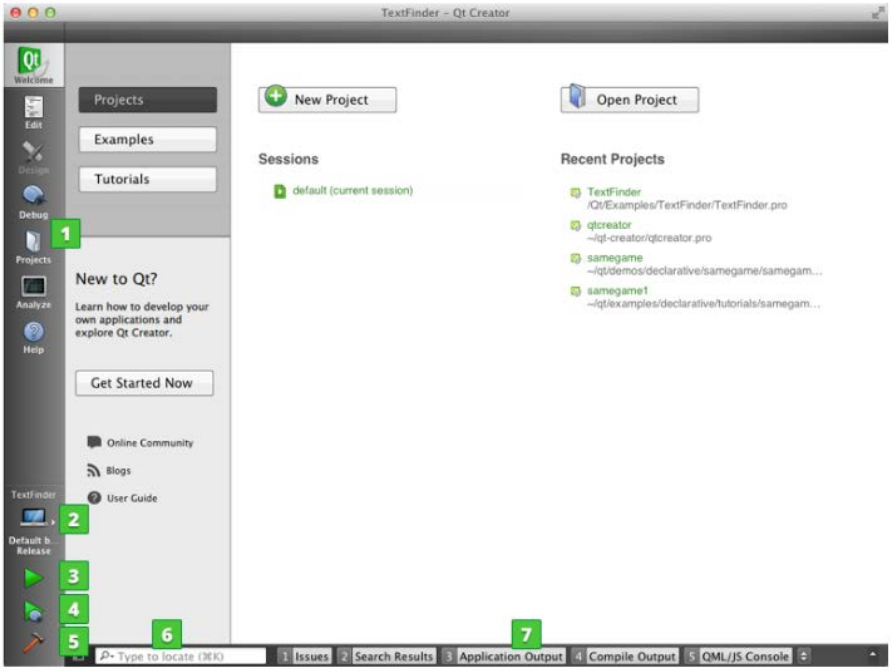


Figure B.1: Qt Creator User Interface

You can use the mode selector (1) to change to another Qt Creator mode. You can use the kit selector (2) to select the kit for running (3), debugging (4), or building (5) the application. Output from these actions is displayed in the output panes (7).
You can use the locator (6) to browse through projects, files, classes, functions, documentation, and file systems.

On the left of the Qt Creator User Interface, there are different icons that represent the modes you can use Qt Creator. The most important are:

- **Welcome mode** for opening projects

- **Edit mode** for editing project

- **Debug mode** for inspecting the state of the application while debugging

- **Projects mode** for configuring project building and execution. This mode is available when a project is open.

Create a new project

The well-functioning project used for the Desktop Driving Simulator has been developed thanks to VTI and called vsim12.

But firstly, how to start working with the Desktop Driving Simulator's software creating a new project is shown.

In order to do that, the Qt application has to be launched. Once is run *(Figure B.2)*, follow this path:

Select File > New File or Project > select the type of the project Applications > Qt Quick Application > Choose

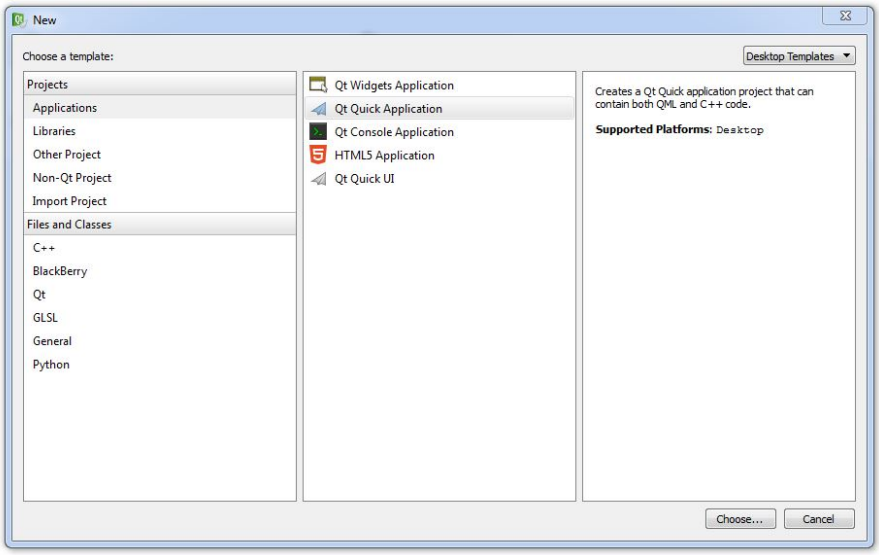


Figure B.2: Creating a new Qt Creator project

On the next screen, type the project's name and select a location for the project. To do this last task, check the option *Create in* and specify settings for it (*Figure B.3*).

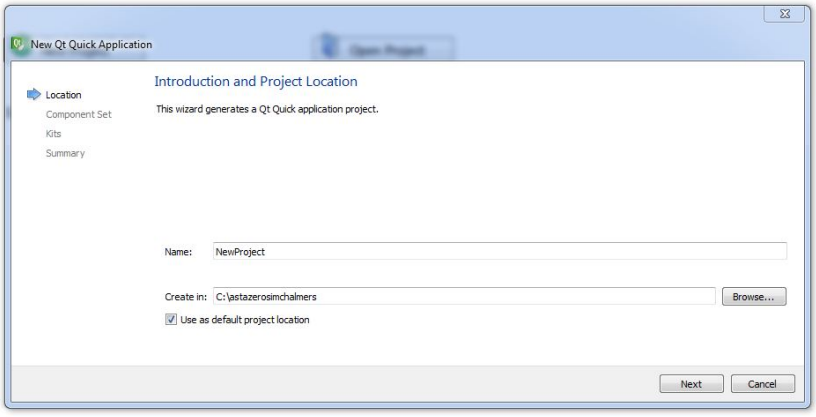


Figure B.3: Type the project's name

Afterwards, in the *Qt Quick component set* field, select the component set to use for the project.
Hence, select *kits* (set of values that define one environment) for running and building the project, and then click *Next*.

If you have built the project before, Qt Creator can use the existing build configuration to make the exact same build as found in the directory available to Qt Creator. To import a build, specify a directory in the *Import Build from section* and select *Import*.

Review the project settings, and click *Finish* to create the project.
When you have completed the steps, Qt Creator automatically generates the project with required headers, source files, user interface descriptions and project files.
So the project is now ready to be executed.

Open and run the project

When a new project is created, it must be loaded from the Qt Creator's main screen. Click on *Open File or Project* and search for the created file in the project folder.
Then the project explorer's page will appear (*Figure B.4*).

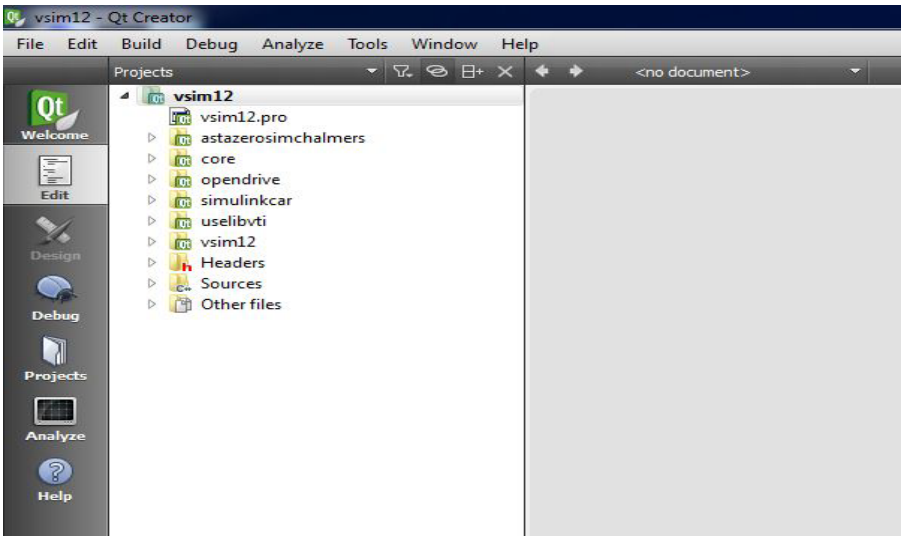


Figure B.4: vsim12 project's main page

In order to prepare the Simulator PC running the simulation, the project must be uploaded on it. This operation is called deploying.
In order to do that, on the project explorer, click on *Build > Deploy Project*.
When the project has been deployed, it can be effectively launched, clicking on *Build > Build Project* and then *Run*.

Now the Desktop Driving Simulator is ready to start a simulation.

Appendix C

Confidential.

Appendix D. Logged data

Tables D.1 and *D.2* below show a logged data template, including a short description and units.

Three coordinate systems are used:

1. Body fixed system, right-handed cartesian DIN-system:

- x is forward
- y is left
- z is upward
- CCW is positive angle

2. Track system, (unlinear road following) right-handed system defined in OpenDRIVE:

- s is position of OVE origo (defined as center point of front wheel axis) along chord line from the beginning of the road, calculated in xy-plane (no elevation taken into account)
- r is lateral position OVE (Own Vehicle) origin with respect to road center coordinate line
- h is height above road surface
- yaw is CCW positive angle with respect to center coordinate line tangent

3. Inertial system, world global right-handed cartesian system:

- X is east
- Y is north
- Z is up
- heading is CCW positive angle, with respect to east direction

More detailed info can be found in Karsolia A., (2014:06. “To be published”), Master’s thesis, Department of Applied Mechanics, Division of Vehicle Engineering and Autonomous Systems, Chalmers University of Technology, Göteborg, Sweden.

Table D.1: Logged data (1-27) template

	Data Field	Description	Unit
1	timer	Absolute simulator time since program start	s
2	ove_odometer	Distance driven	m
3	road_id	Current road id	-
4	s	longitudinal position, s in Track system	m
5	r	lateral position, r in Track system	m

6	yaw	Yaw angle relative to road tangent	rad
7	vx	Body fixed longitudinal velocity	m/s
8	vy	Body fixed lateral velocity	m/s
9	ax	Body fixed longitudinal acceleration	m/s^2
10	ay	Body fixed lateral acceleration	m/s^2
11	yaw_vel	Yaw velocity	rad/s
12	eng_torq	Engine torque	Nm
13	engine_rps	Engine revolution	rad/s
14	throttle	Throttle position, 0 no throttle	-
15	brake_pedal_active	Brake pedal active	-
16	brake_pedal_press	Brake pedal pressure	kPa
17	brake_force	Approx brake force applied to pedal	N
18	stw_angle	Steering wheel angle, CCW positive	rad
19	stw_torq	Steering wheel torque	Nm
20	left_indicator	Left indicator, 1 = active	-
21	right_indicator	Right indicator, 1 = active	-
22	gear	Gear number	-
23	event_id	Active event's id	-
24	event_state	Current state in active event	-
25	event_state_timer	Time spent in current state	s
26	X	Global X coordinate	m
27	Y	Global Y coordinate	m

28 – 61 Packet from Driving Simulator to VDM.

Table D.1: Logged data (28-61) template

	Data Field	Description	Unit
28	watchdog	Watchdog counter	
29	resetIn	Signal to reset the model to original state	
30-32	Fxyz_ext_cg	External Forces at centre of gravity	N
33-35	Mxyz_ext_cg	External torque at centre of gravity	Nm
36	SWA	Steering wheel angle	rad
37	gear_manual	Gear (1-12), 0 = neutral	-
38	Clutch	(0-1)	%
39	throttle	(0-1)	%
40	brake_pedal_input	Pressure [0-inf] or pos [0-100]	%
41-52	z_dzdx_dzdy	4 wheels*[z,dzdx,dzdy]	m
53-56	mu	Friction coefficient for 4 wheels	
57-60	P_brk_wheels	Brake Pressure [LF,RF,LR,RR]	Pa
61	Vx_max	Max. Longitudinal Velocity	m/s

62 – 119 Packet from VDM to Driving Simulator.

Appendix E. Simulations results

In this appendix simulations result of the experiments are shown.

Straight Ahead Acceleration. Offline simulations

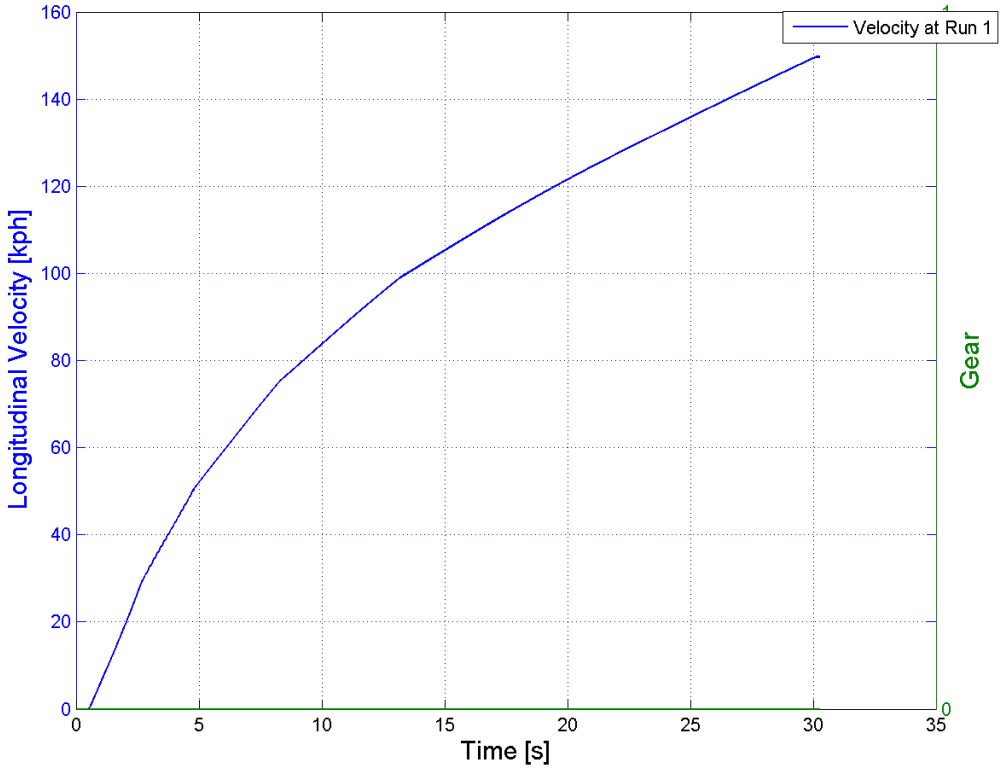


Figure E.1: Straight Ahead Acceleration (offline simulation). Longitudinal velocities acceleration test. The maximum velocity during the test was 150 kph. Brake due to too low longitudinal acceleration

Table E.1: Time between two velocities during ongoing full acceleration

Velocity change	Time
0 – 60 [kph]	5.62 [s]
0 – 100 [kph]	12.97 [s]
80 – 120 [kph]	10.25 [s]

Table E.2: Time for traveling the 402 meters

Distance	Time
402 [m]	18.93 [s]

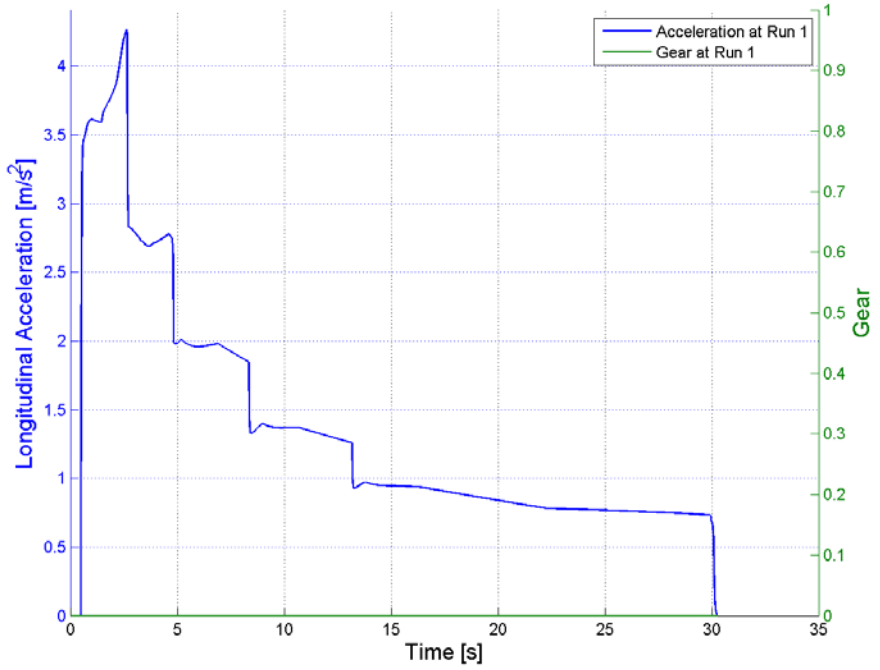


Figure E.2: Straight Ahead Acceleration (offline simulation). Longitudinal acceleration during test

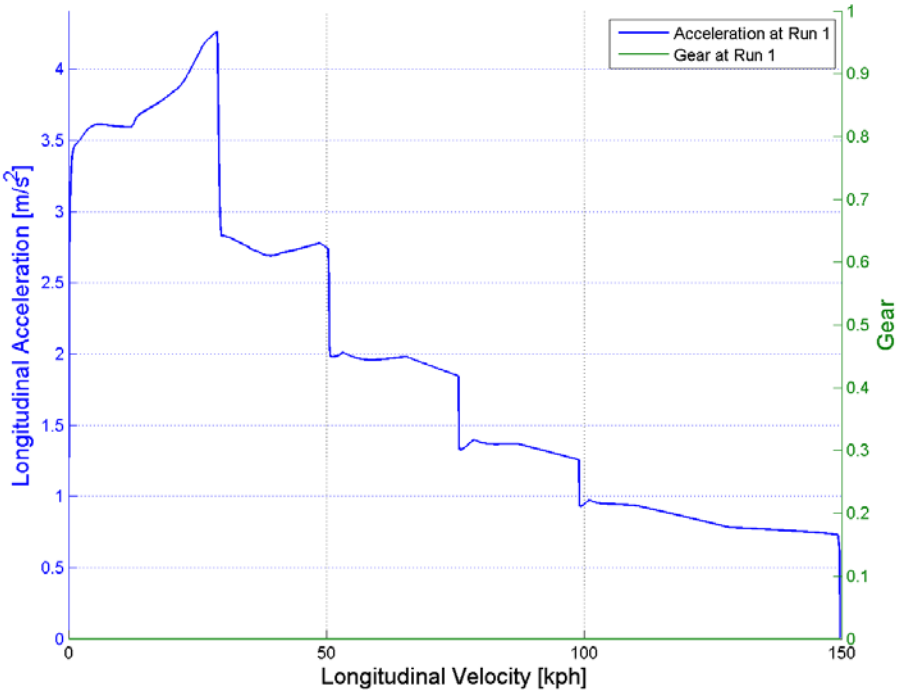


Figure E.3: Straight Ahead Acceleration (offline simulation). Longitudinal acceleration as a function of longitudinal velocity

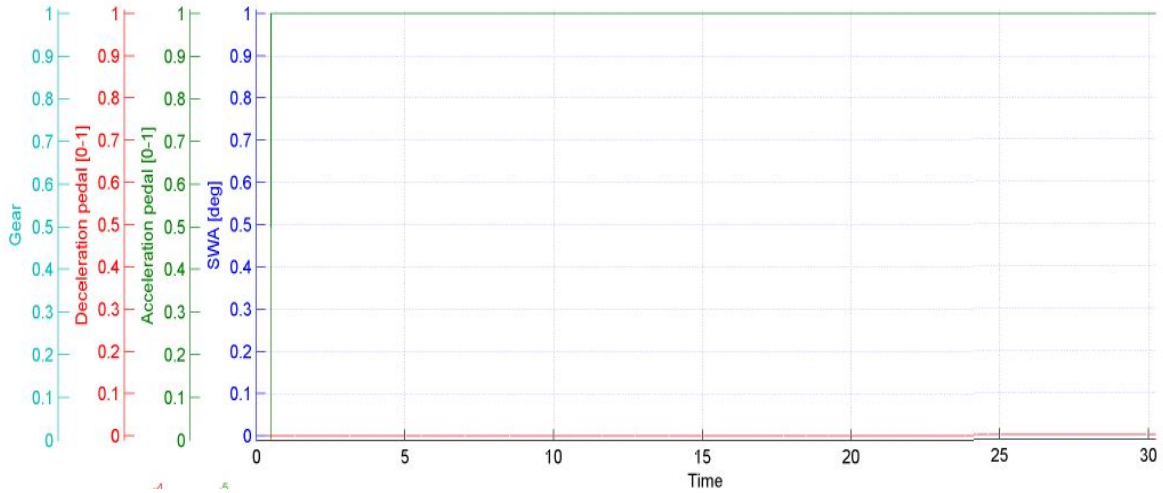


Figure E.4: Straight Ahead Acceleration (offline simulation). Evaluation plot from run 1. Steering wheel angle, acceleration pedal and deceleration pedal as functions of time

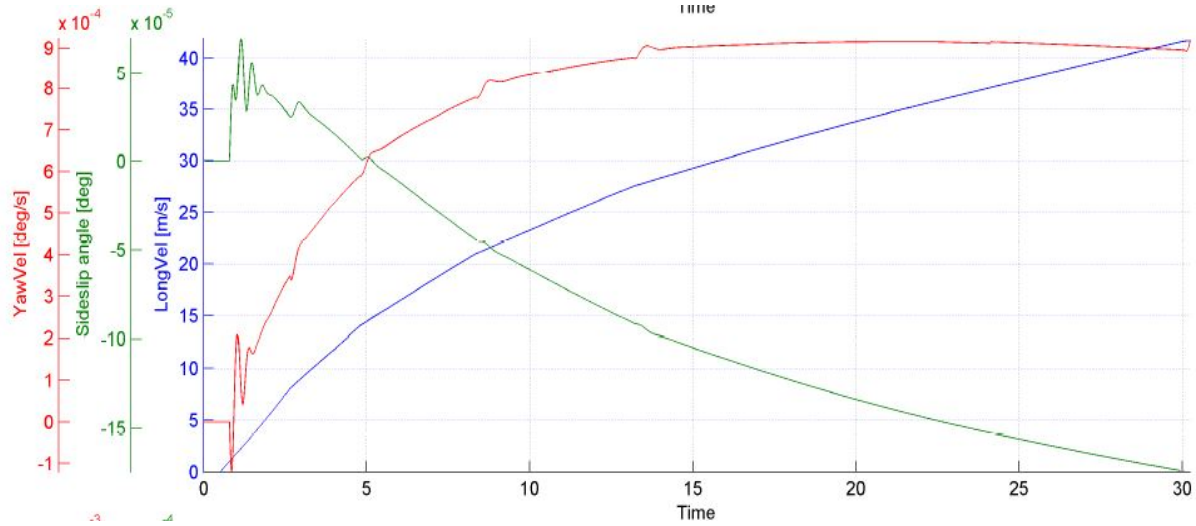


Figure E.5: Straight Ahead Acceleration (offline simulation). Evaluation plot from run 1. Longitudinal velocity, side slip angle and yaw velocity as functions of time

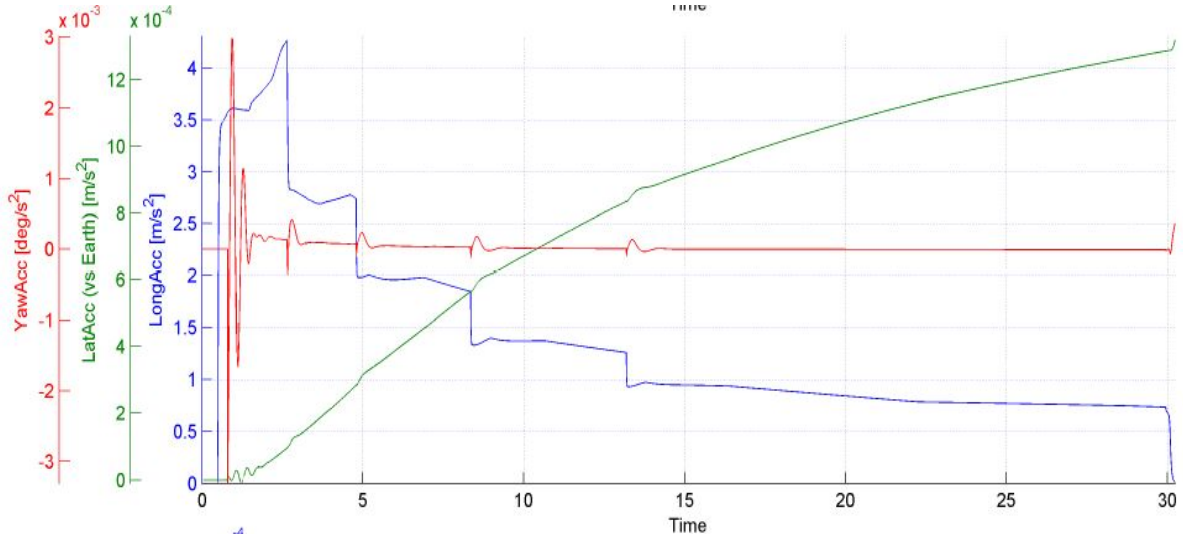


Figure E.6: Straight Ahead Acceleration (offline simulation). Evaluation plot from run 1. Longitudinal acceleration, lateral acceleration and yaw acceleration as functions of time

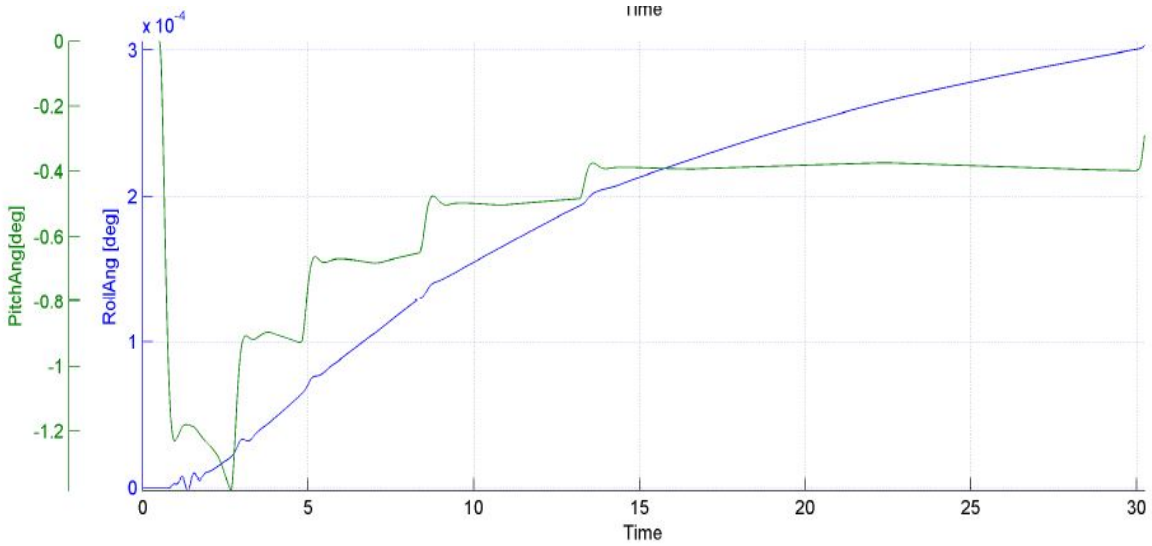


Figure E.7: Straight Ahead Acceleration (offline simulation). Evaluation plot from run 1. Roll angle and pitch angle as functions of time

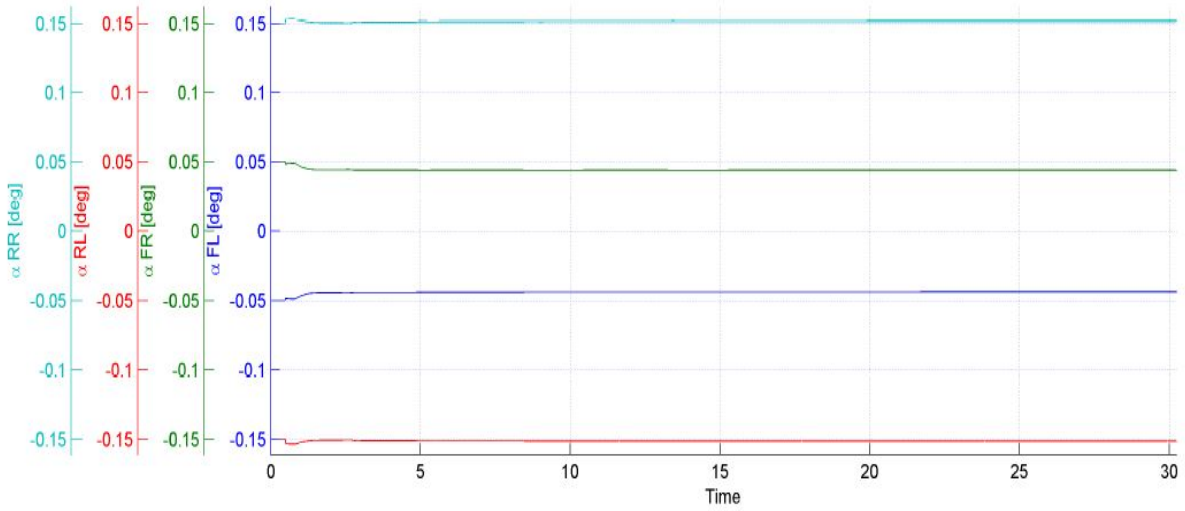


Figure E.8: Straight Ahead Acceleration (offline simulation). Evaluation plot from run 1. Tire slip angle for the four wheels (FL, FR, RL, RR) as functions of time

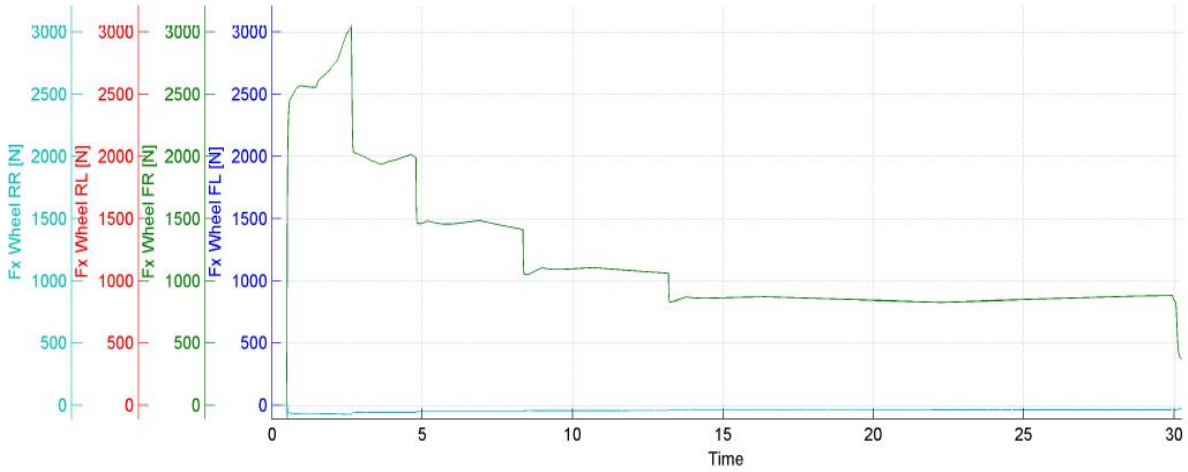


Figure E.9: Straight Ahead Acceleration (offline simulation). Evaluation plot from run 1. Tire longitudinal force for the four wheels (FL, FR, RL, RR) as functions of time

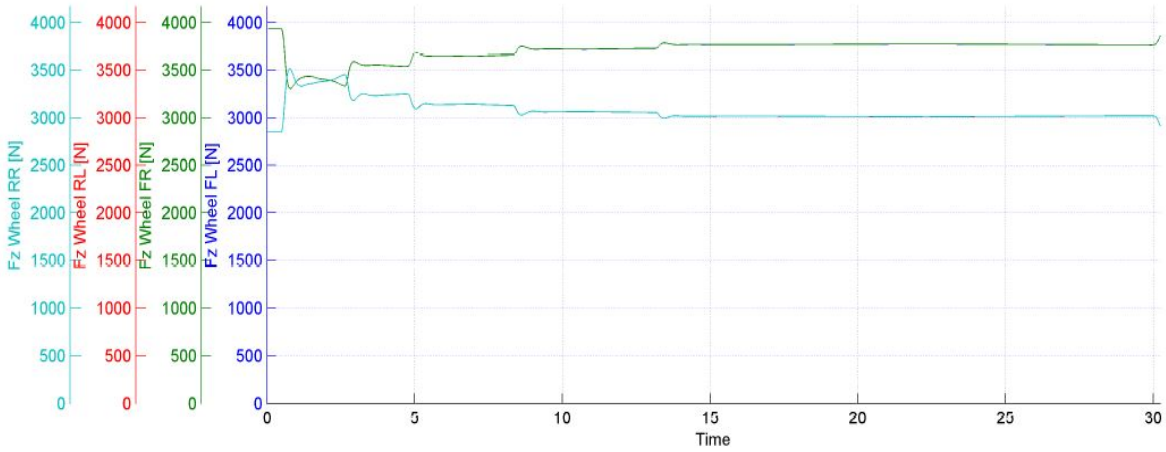


Figure E.10: Straight Ahead Acceleration (offline simulation). Evaluation plot from run 1. Tire vertical force for the four wheels (FL, FR, RL, RR) as functions of time

Straight Ahead Acceleration. Online simulations

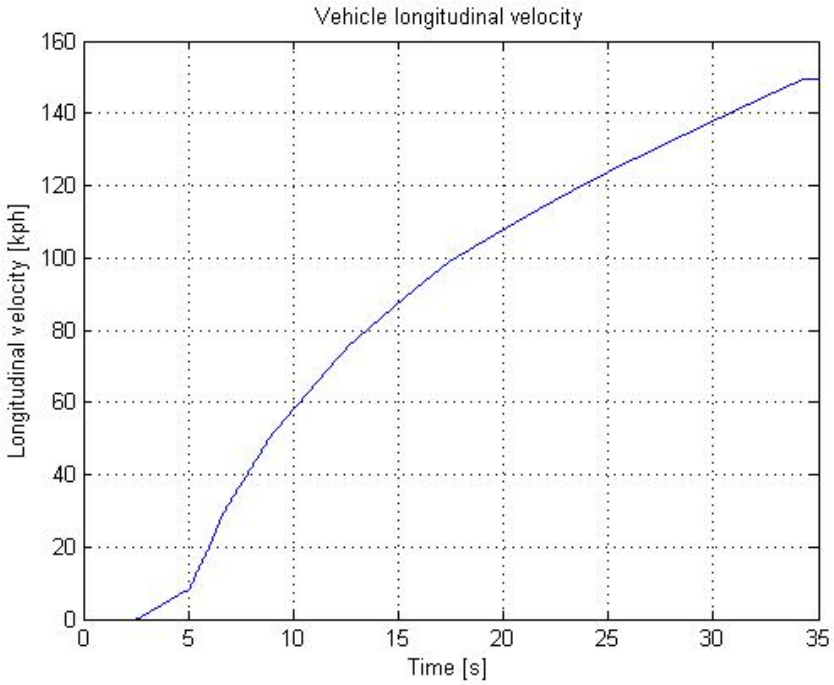


Figure E.11: Straight Ahead Acceleration. Longitudinal velocity during online simulation

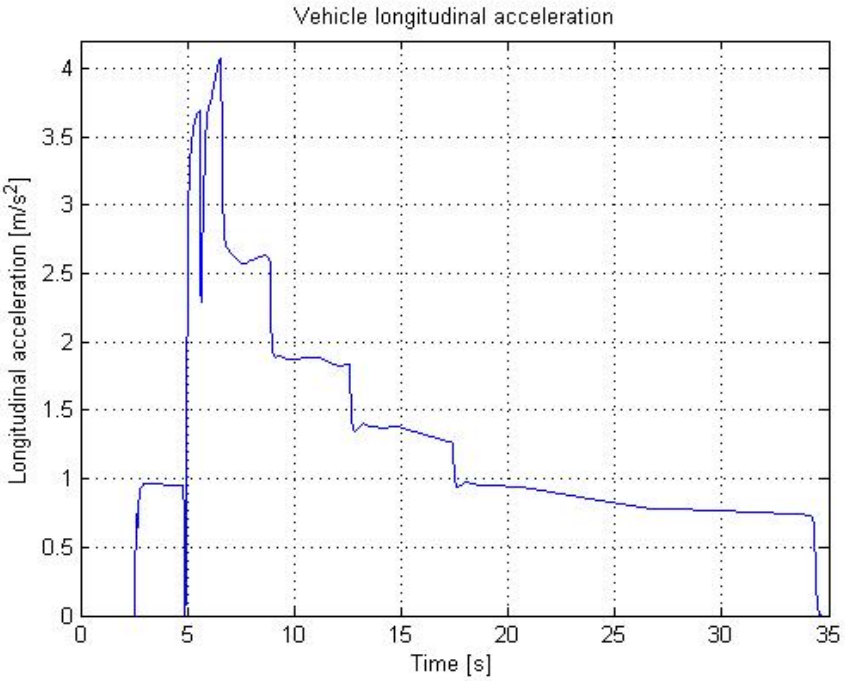


Figure E.12: Straight Ahead Acceleration. Longitudinal acceleration during online simulation

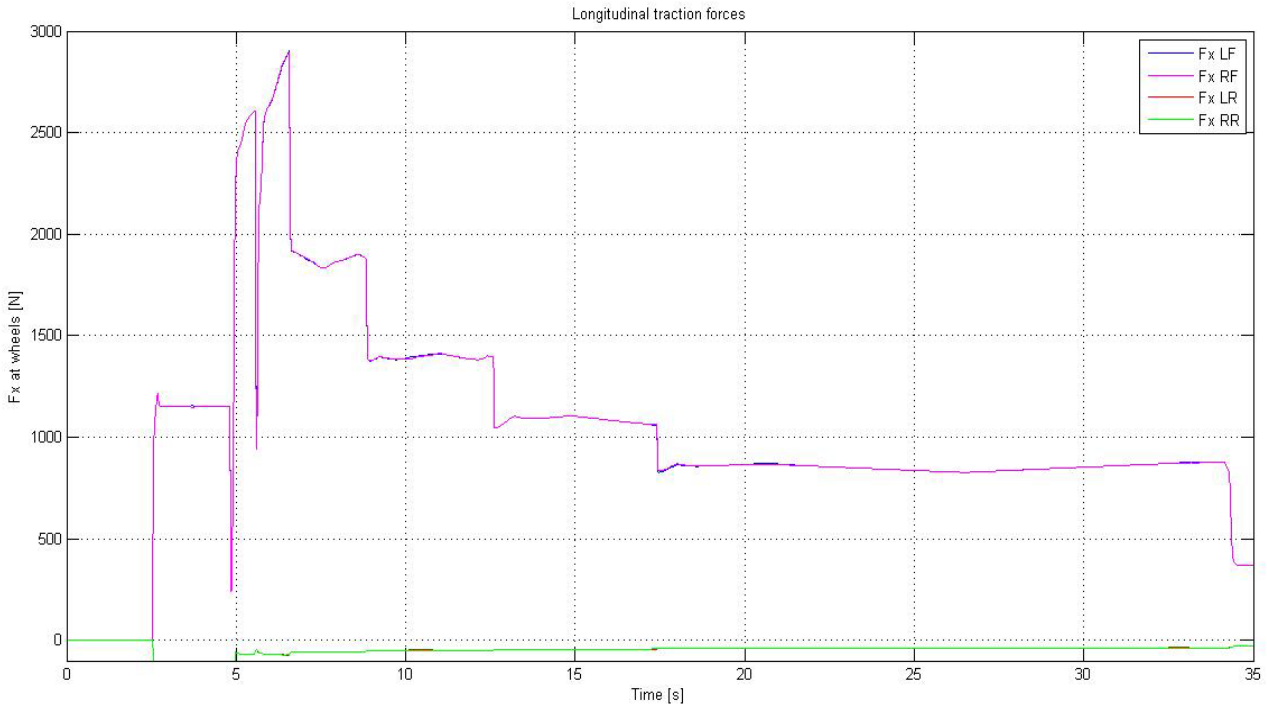


Figure E.13: Straight Ahead Acceleration. Tire longitudinal traction force for the four wheels (FL, FR, RL, RR) during online simualtion

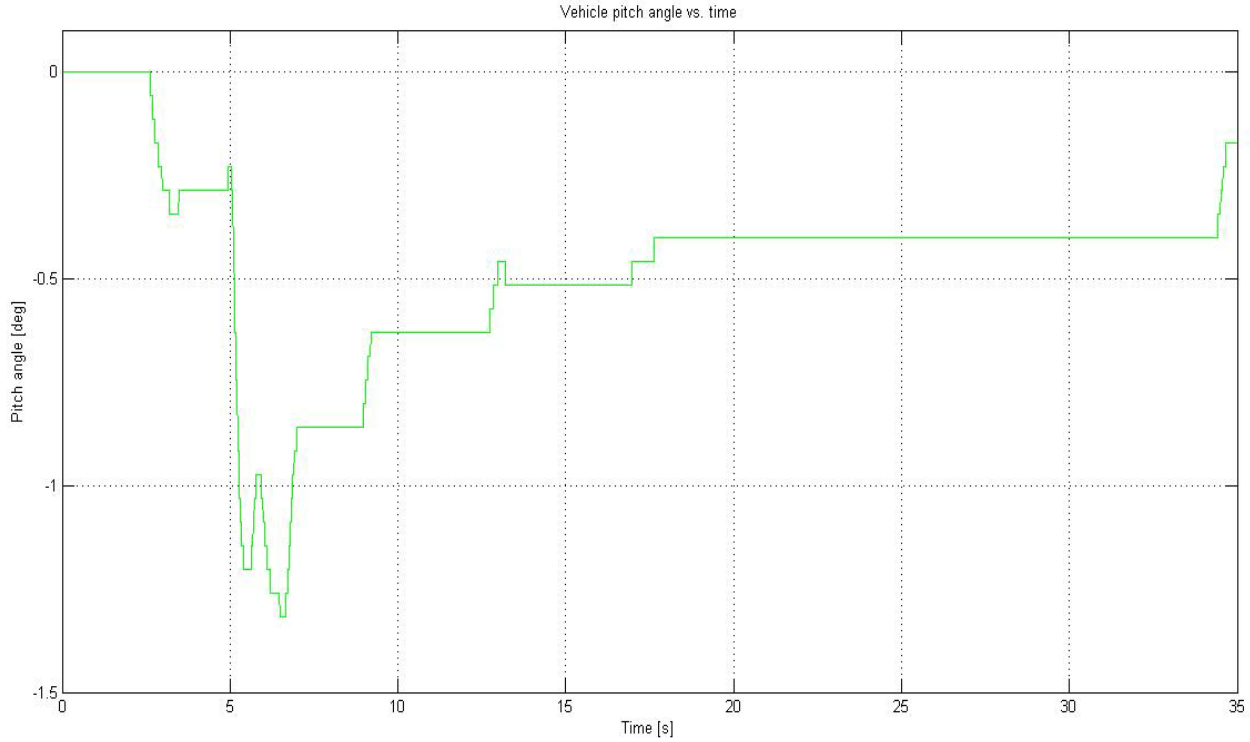


Figure E.14: Straight Ahead Acceleration. Pitch angle during online simualtion

Double-Lane Change

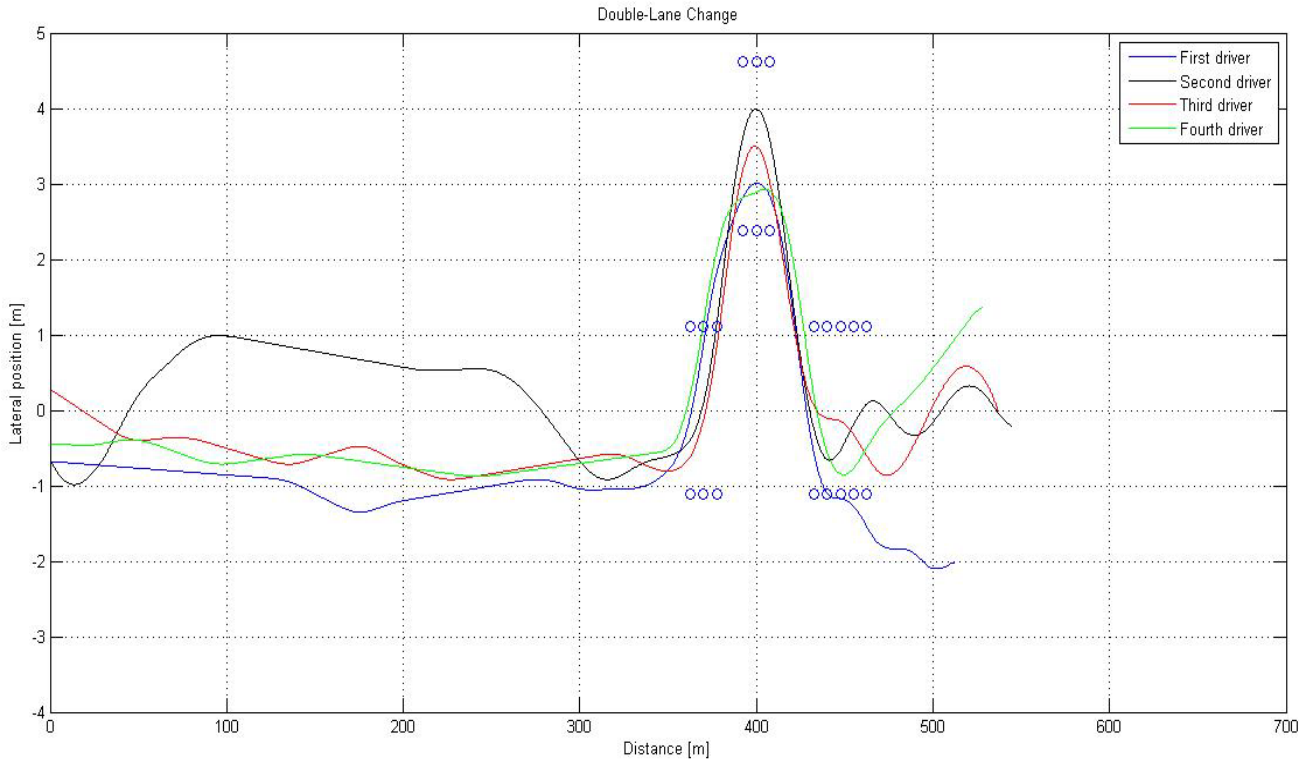


Figure E.15: Double-Lane Change. Distance performed by the four different drivers. Position of the cones is marked with a circle

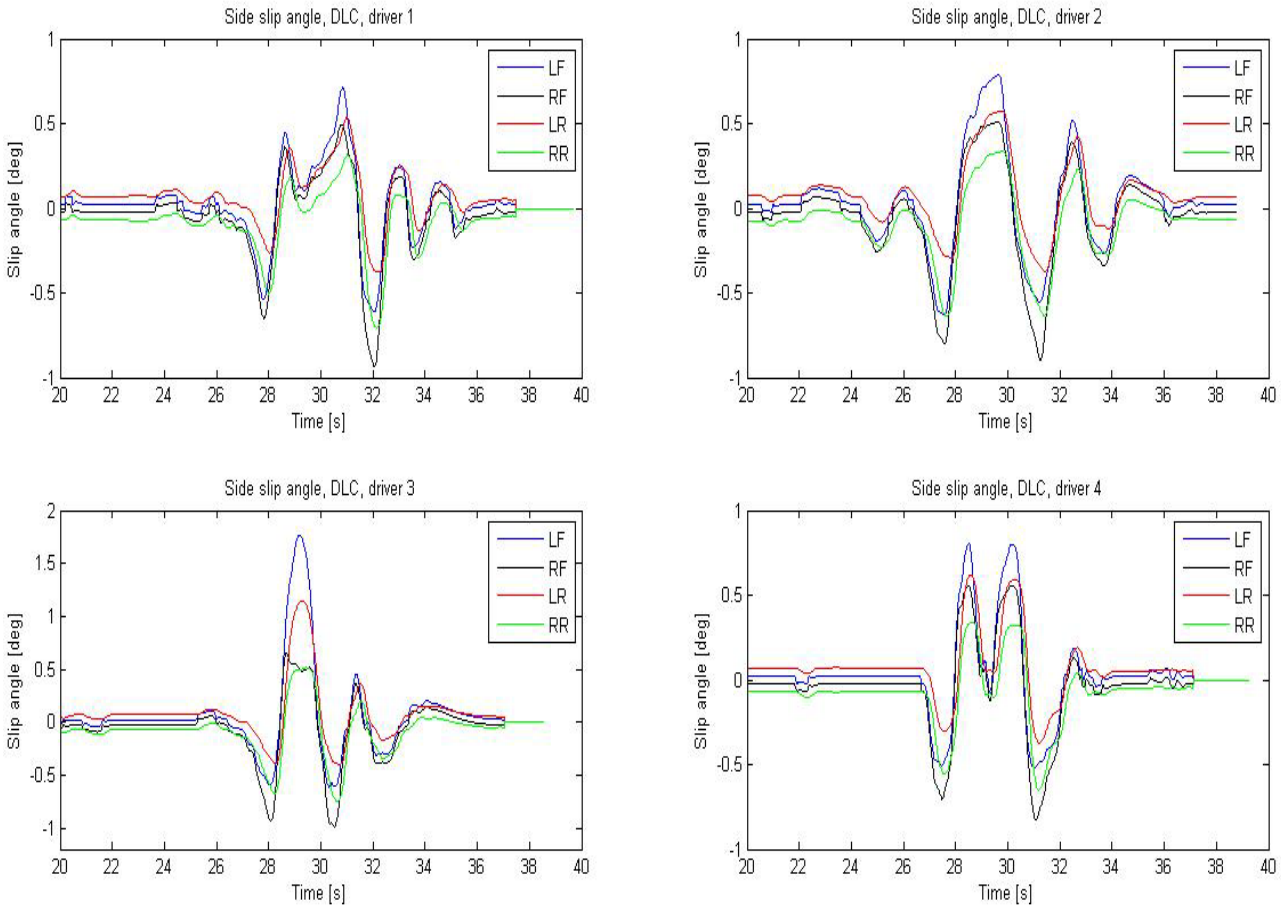


Figure E.16: Double-Lane Change. Side slip angle as function of time for the four wheels (FL, FR, RL, RR) for each single driver

Straight-Line Braking

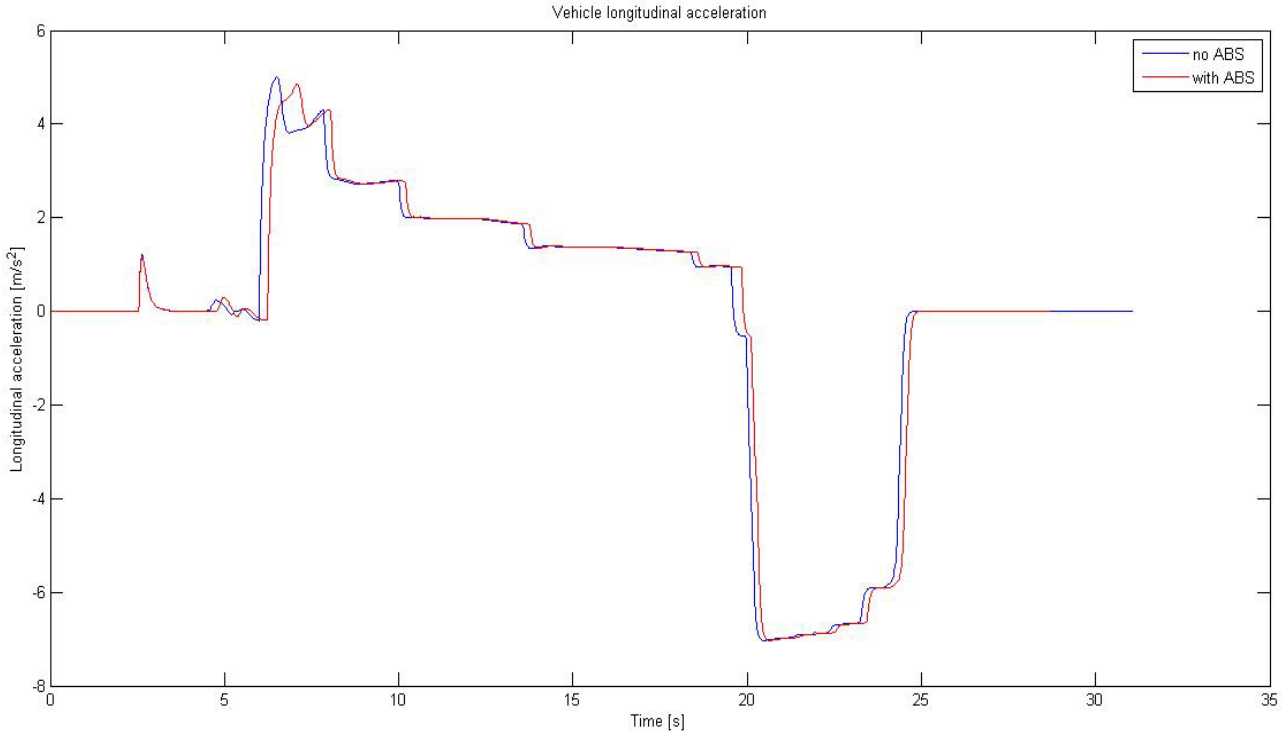


Figure E.17: Straight-Line Braking. Vehicle longitudinal acceleration during the test with and without ABS system

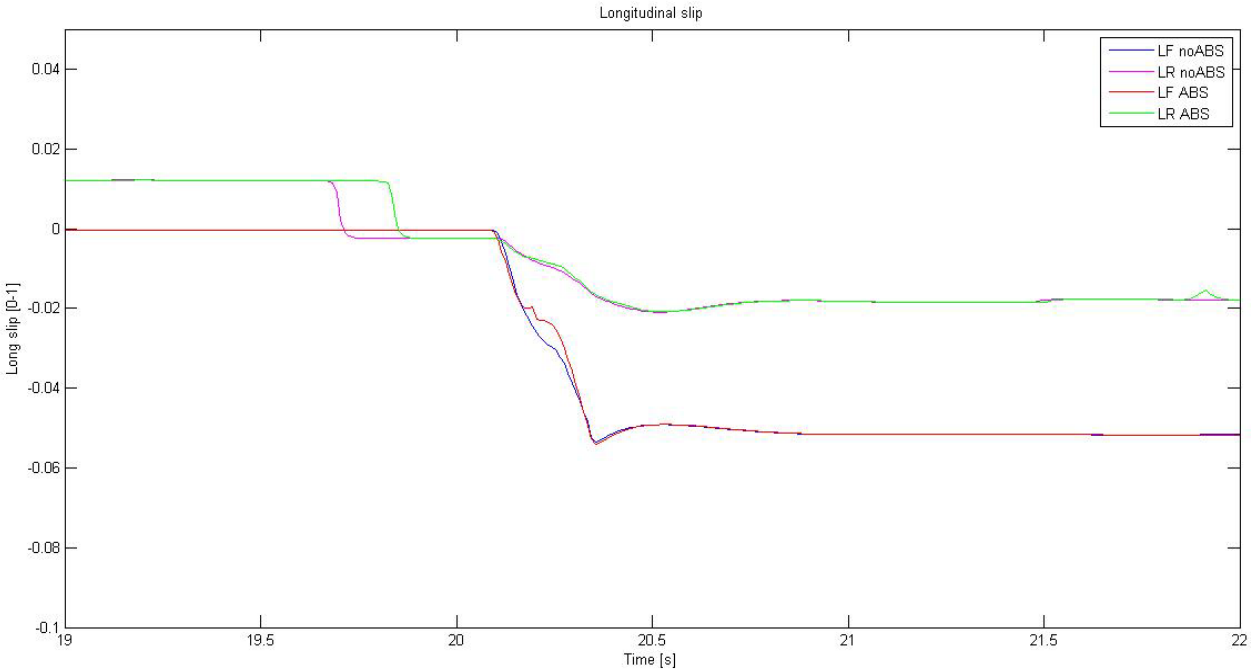


Figure E.18: Straight-Line Braking. Longitudinal slip for LF and LR as function of time, in both cases (with/without ABS)

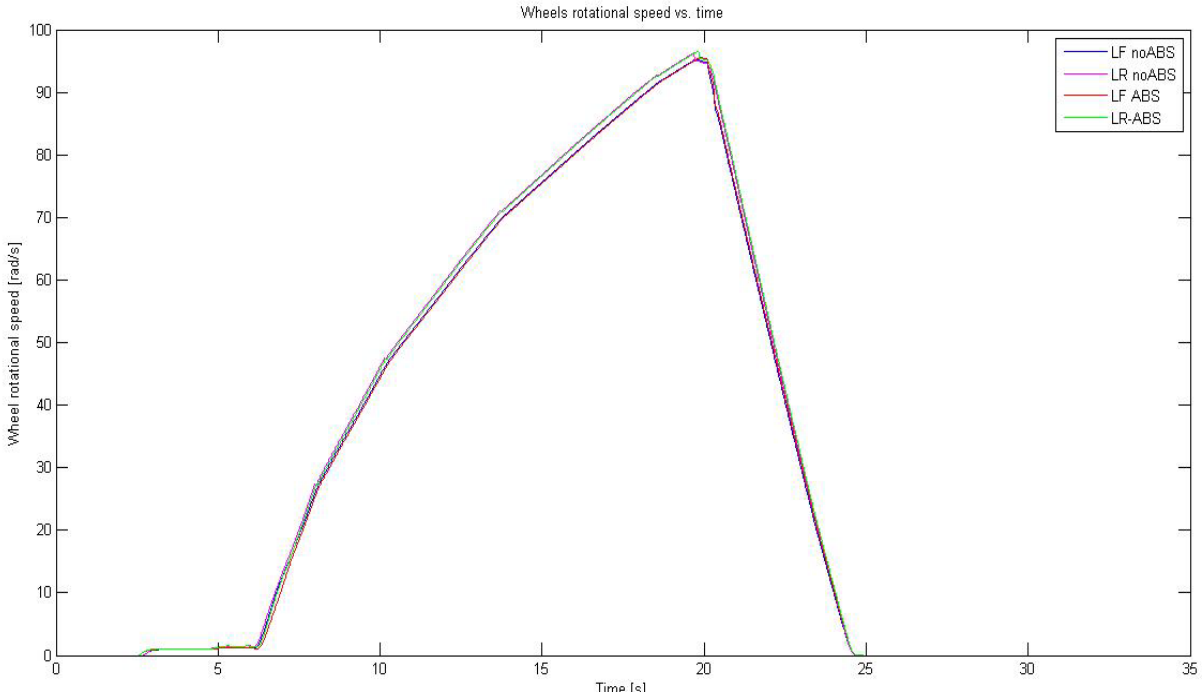


Figure E.19: Straight-Line Braking. Wheels rotational speed for LF and LR as function of time, in both cases (with/without ABS)