University of Pretoria

MSC422 Research Project

Implementation of a Hardware-in-the-Loop Simulation for an Automobile's ABS

Author
Alexander Berndt

Supervised by
Dr Theunis Botha

This is the final report for the MSC422 Research Project

November 20, 2016

Executive Summary

Introduction

Hardware-in-the-Loop (HiL) simulation is a relatively new concept which is currently used worldwide to test real hardware and validate controller designs. This testing method is most beneficial when some aspects of a system's physical hardware are too complex or computationally demanding to simulate.

Purpose

This research project's objective is to implement an HiL test of a Land Rover's ABS.This would allow the Vehicle Dynamics Group (VDG) of the University of Pretoria to design and validate different ABS algorithms without needing to transport the automobile to a test track and driving/testing it there.

Procedure

The report delves into the design of a SimuLink © simulation consisting of vehicle and wheel dynamics components, two different tyre models (modified Pajecka and LuGre) and an industry standard ABS algorithm. This simulation is validated in a pure simulation and then modified to communicate with actual hardware to perform an HiL simulation.

In order to achieve this goal, embedded computers are programmed to communicate with the necessary sensors, allowing for a HiL environment.

Results Obtained

Simulation results yield the successful controlled braking of a vehicle using a Bosch ABS algorithm. The HiL setup is validated and a successful HiL test is conducted.

Conclusion

In conclusion, it can be stated that a successful HiL setup was implemented for an ABS system. This system was validated using simulation and two different tyre models.

Abstract

Simulating a vehicle's ABS is a laborious task due to it's physical complexity. The developed mathematical models representing the braking hydraulics are often too complex to solve in a real-time environment which impedes the development of ABS algorithms. This report documents the development of a Hardware-in-the-Loop simulation which allows ABS algorithms to be tested in real-time. The report describes the modelling process that was followed as well as the experimental results obtained from a Hardware-in-the-Loop simulation.

Acknowledgements

I would like, first and foremost, to thank Dr Theunis Botha for his guidance and enthusiasm throughout the project. A special thanks also goes to all the members of the Vehicle Dynamics Group, in particular, Mr Wietsche Penny, Mr Reyn Kapp and Mr Herman Hamersma for their general support and their collaborative attitude which resulted in a fantastic learning experience for me this past year.

Contents

Nomenclature

Greek Symbols

Symbol	Description	Units
α	Rotational acceleration	$\mathrm{rad\,s^{-2}}$
μ_c	normalized coulomb friction	N/N
μ_s	normalized static friction	N/N
ω	Rotational velocity	$\mathrm{rad\,s^{-1}}$
ϕ	Angle	rad
σ_0	rubber longitudinal lumped stiffness	$\mathrm{m^{-1}}$
σ_1	rubber longitudinal damping coefficient	$\mathrm{s\,m^{-1}}$
σ_2	viscous relative damping coefficient	$\mathrm{s\,m^{-1}}$
θ	Angle	rad

Roman Symbols

Symbol	Description	Units
a	Acceleration	$\mathrm{m\,s^{-2}}$
F	Force	N
g	Graviational Acceleration	$\mathrm{m\,s^{-2}}$
J	Rotational Inertia	$\mathrm{kg\,m^2}$
M	Mass	kg
s	Slip	%
T	Torque	N m
v_r	relative velocity	$\mathrm{m\,s^{-1}}$
v_s	stribeck relative velocity	$\mathrm{m\,s^{-1}}$
z	Tyre deformation	m

Superscripts

Symbol	Description	Units
$\ddot{}$	Second derivative	
$\dot{}$	Derivative	

List of Figures

List of Tables

Introduction

1.1 Background

The automation of a system generally requires a mathematical model of this system coupled with a controller. For example, when a controller needs to be designed for an automobile, it is very important that accurate mathematical models of the automobile's subsystems (e.g. the suspension dynamics and braking system) are established such that the physical reality of the automobile is well represented [Halvorsen, 2014]. The accuracy of these mathematical models is of vital importance because this directly influences the effectiveness of the controller and allows for increased robustness and overall capability. [Isermann et al., 1999]

In order to create a sufficiently accurate mathematical model of a physical system, the physics of the system needs to be considered. Often times, however, systems are too difficult to model due to their complexity. Another common problem is that a system's mathematical model is too computationally demanding to allow for a real-time simulation.

At this point, a Hardware-in-the-Loop (HiL) simulation becomes almost a necessity. HiL simulation is a method where the sub-system that is too difficult to model or too computationally demanding for a real-time simulation, is physically connected to the simulation with sensors and actuators. This simulation is such that it allows the physical hardware to interact with the rest of the simulation in real-time.

The ABS of an automobile is a good example of hardware that is computationally tedious and difficult to simulate in a real-time simulation [Anakwa et al., 2001]. This is especially true for the modelling of the hydraulic valves of the ABS system. This research project will aim to implement a HiL simulation of an automobile's ABS so that ABS algorithms can be tested in real time.

1.2 Previous Work on HiL Simulations

Seeing as HiL simulations are a core component to a manufacturer's ability to quickly develop complex engineering products, the literature available on HiL simulations is abundant.

Isermann et al. considers two HiL simulation cases. The first being of relatively slow processes such as heating systems, the second being an HiL simulation of an internal combustion engine. The paper provides details of the mathematical models used for the simulation as well as a comparison of the real-time simulation results with experimental data obtained from actual diesel engines [Isermann et al., 1999].

Sun et al. invented and implemented an HiL simulation method for the Alpha Magnetic Spectrometer on the International Space Station. This paper delves into the system framework, implementation and results of this simulation [Sun et al., 2016].

Lee and Suh analysed commercial electronic control units and components for ABS and TCS systems using HiL simulation [Lee and Suh, 1999].

Other available literature includes Svaricek's paper on the automatic valuation and verification of ABS controllers by using a hardware-in-the-loop simulation [Svaricek, 1999].

1.3 Problem Statement

The Vehicle Dynamics Group (VDG) of the University of Pretoria requires the testing of an ABS algorithm of an automobile, in this case a Land Rover Defender. In order to test the brake performance, the Land Rover can be transported to a testing-track and evaluated there. An alternative, however, is to use an HiL simulation of the ABS brake system of the Land Rover and run a simulation in the VDG laboratory.

However, the VDG currently does not have a HiL setup for this testing procedure. The problem statement is therefore the VDG's need for an HiL setup that can be used to test the Land Rover's braking performance in the laboratory.

1.4 Motivation and Objectives for this Research

The Vehicle Dynamics Group has a Land Rover which has already been connected to a variety of systems which allows for autonomous control of the vehicle. The development of the controller of the vehicle is a very important aspect of the VDG's research. This includes the braking controller and ABS algorithm used in the Land Rover.

The results of this research project will provide the VDG with the ability to test ABS algorithms and braking controllers in the laboratory without needing to transport the Land Rover to a test track.

1.5 Overview of this Report

This report first delves into the theory behind ABS systems and basic vehicle dynamics models. The report then covers an experimental ABS simulation using the aforementioned models. A Hardware-in-the-Loop setup is then described and implemented, of which the results are also documented. Finally, a conclusion is drawn regarding the validity of the experiment.

Literature Study

This literature study aims at delving into the critical aspects of research required to implement an HiL simulation of an automobile's ABS. This includes the structure and fundamental components of a successful HiL system, working fundamentals of ABS systems as well as the relevant vehicle dynamics relations required to implement the simulation of a braking vehicle.

2.1 Hardware-in-the-Loop (HiL) Simulation

2.1.1 Overview of HiL Simulation

Hardware-in-the-loop (HiL) simulation is a relatively new concept and was only first implemented approximately 20 years ago. The HiL design process originated in the aviation industry but has since branched out to many other fields such as the automotive, marine and defence industries [Gomez, 2001]. HiL testing and simulation has since become an invaluable development process for almost all technical product development industries. The nature of HiL testing allows for much faster product development as well as the inclusion of highly complex systems in simulations.

Halvorsen lists the following reasons HiL simulations are conducted [Halvorsen, 2014]:

1. It allows the inclusion of a physical system, which is too difficult to model, in the simulation so that the overall controller can be tested and altered.
2. To allow for stationary testing and development of usually moving vehicles.
3. It gives system developers and engineers the ability to test actual hardware in real-time.
4. Safety is increased tremendously with HiL since human operators can train in much safer environments than reality. An example being flight simulators for airplane pilots.
5. Parallel systems engineering is also simpler with the use of HiL since various engineering teams can develop systems without overlooking integration issues.

2.1.2 The HiL procedure

Halvorsen published a paper stating the three main steps towards implementing a successful HiL simulation and testing procedure [Halvorsen, 2014]. The three steps are:

1. *Mathematical Model Development*
   A mathematical model needs to be developed which represents the real environment that the hardware will be used in.
2. *HiL simulation*
   Couple the software and hardware and test the hardware using the mathematical model setup mentioned in the previous step.
3. *Implement the hardware in reality*
   If the hardware performed satisfactorily in the HiL test, implement it in a real environment.

2.1.3 Requirements for Successful HiL Simulation

Implementing a successful HiL simulation naturally necessitates a set of requirements that need to be fulfilled. A paper published by Hosam K Fathy for the University of Michigan has recognised six 'enablers' that need to be addressed to ensure successful HiL implementation [Fathy et al., 2006].

1. *Sensor and actuator fidelity, bandwidth and unobtrusiveness.*
   Since an HiL simulation requires the constant communication between simulation software and hardware, the sensors and actuators that enable this communication need to function well. The simulation needs to capture the hardware's behaviour. This requires the sensors and actuators to have a high enough bandwidth for real-time communication, high fidelity and low to zero obstruction to the hardware system.
2. *Digital signal processing and signal conditioning*
   Sensor data often has an unavoidable amount of noise. This data therefore needs to go through accurate and efficient signal processing to ensure that the state of the hardware is accurately represented by the data.
3. *Fast processors, RTOS and fixed step integration*
   An HiL simulation requires real-time interaction between the simulation and the hardware. It is therefore of paramount importance that the embedded computer (such as a dSPACE MicroAutoBox II [dSPACE.de, 2016]) has a sufficiently fast processor and real-time data monitoring capabilities such as a Real-Time Operating System (RTOS). [Liu, 2000]
4. *Data acquisition accuracy and dynamic capabilities*
   The sensors chosen to measure the relevant inputs and outputs of the hardware that is 'in-the-loop' need to be sufficiently accurate at measuring the measured state of the hardware.
5. *Accurate and efficient modelling*
   The mathematical modelling of the system needs to conform to two major require-

ments in order to ensure an effective HiL implementation. The first being that the accuracy of the model must be such that it sufficiently encompasses the physics of the system so that the hardware can react accordingly and represent physical reality as best possible. The second is that the model must be computable in real-time. It is recognized that these two requirements typically conflict and that a model is considered to be acceptable if it successfully balances these two requirements.

6. *Hardware and Software integration*
   For an HiL simulation to take place effectively, it is not sufficient to only have an accurate model and a hardware setup that communicate in real time. Successful integration of hardware and software is a very important aspect of efficient HiL simulations.

2.1.4 Architecture of the HiL Setup

A schematic of the overall HiL simulation set-up is shown in Figure 2.1 [Gomez, 2001]. This schematic depicts the fundamental structure and components of an HiL setup. The internal architecture of the HiL Simulator block of Figure 2.1 is shown in Figure 2.2.

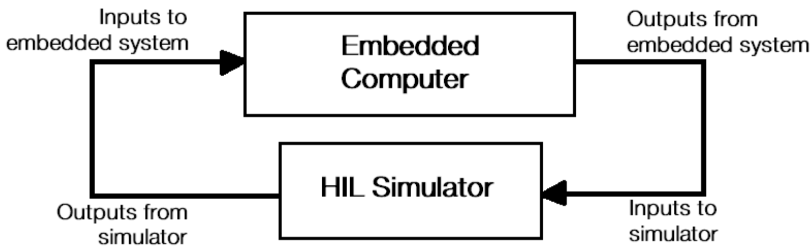

Figure 2.1: Basic System Architecture for an HiL Setup [Gomez, 2001]

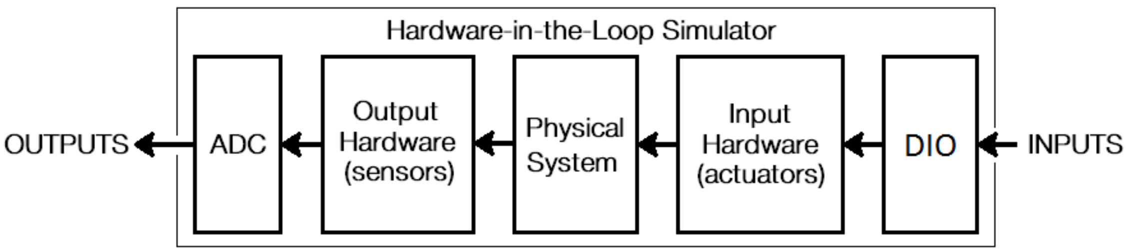

Figure 2.2: Components of Hardware block for HiL Simulation

2.1.5 The Need for Real-Time Computing Capabilities

Since the model that is simulated mathematically needs to interact with the physical system, the embedded computer needs to be capable of running the model in real-time. Real-time computing (RTC) requires software and hardware to comply to certain time constraints. [Kay, 2012].

It is important to note that real-time models need to run at a fixed step size. Considering Figure 2.3, T_{step} is the specified time step (and is the inverse of the simulation frequency),

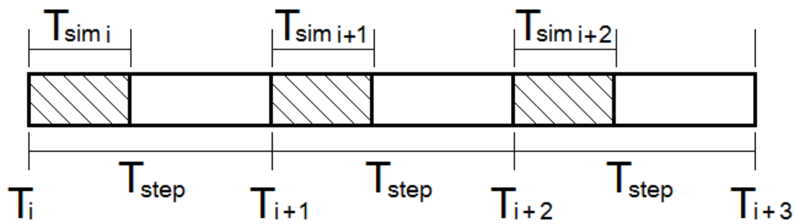

Figure 2.3: Schematic of a real-time simulation process

T_{sim} denotes the time required to solve the simulation model and T_i is the absolute time at that given point in time.

In order run at real-time, the simulation host computer must solve the model faster than the time-step of the simulation. This can be achieved by either increasing the step size T_{step} or, alternatively, by increasing the computational power which will result in a shortened T_{sim}. Another constraint of RTC is the time T_i must correspond with the actual absolute time at that point.

Model Solution Convergence

Increasing the step-size to ensure the simulation time to be below the required step-size can induce errors in the simulation model. Since simulations typically use numerical solvers, convergence of solutions needs to be ensured which usually requires a minimum step frequency. Additionally, some mathematical models are stiff and require very fine step sizes in order to converge. An example of such a model is the LuGre tyre model.

Real-Time Specific Computers

SimuLink© has a Real-Time library which allows for a normal computer to simulate a real time simulation. However, since SimuLink© runs on an operating system, accurate real-time functionality can not be guaranteed. However, with a simple mathematical model and fast external communication, an HiL system could potentially be set-up using this library. [Mathworks.com, 2016].

For simulations that are more computationally demanding, a system such as the dSPACE MicroAutobox II can be utilized. This device has a dedicated Real Time Operating System (RTOS) and is therefore designed to run programs in real-time. It also has very high computational power allowing it to solve complex mathematical models quickly. [dSPACE.de, 2016].

In order to ensure successful and accurate HiL testing, equipment needs to be carefully chosen to ensure that it complies with the aforementioned factors.

2.2 The ABS Braking System

2.2.1 Overview

ABS stands for anti-lock braking system. These systems are designed to prevent wheel lock when the brakes of the car are applied [Transport.ca, 2016]. ABS was first developed for aeroplane undercarriages in the 1940's. However, they were mostly mechanical and their capabilities were far from that of modern ABS systems [Wabco, 2011]. ABS brakes were first included on automobiles in the 1950's and have since evolved into such a necessity for safe driving that they are now mandatory for all production automobiles.

ABS brakes are especially effective when the vehicle is driving on wet surfaces and less breaking force is required to induce wheel lock due to the low friction coefficient at the tyre and the road surface interface.

2.2.2 Objective of ABS Systems

The objective of ABS is two-fold. They are [Wabco, 2011]:

1. **Ensuring shortest stopping distance**
   By ensuring the wheels do not lock up when the brakes are applied, the coefficient of friction between the tyre and the road is at it's peak. This is shown in Figure 2.4. The maximum braking coefficient is at 20% slip which takes place at point 2. By ensuring the slip between the tyre and the road is maintained at 20%, vehicle deceleration can be maximized.

2. **Ensuring vehicle steer-ability**
   Vehicles are found to have practically no lateral grip and vehicle steering therefore becomes practically impossible. By ensuring the wheels do not lock-up, ABS systems ensure vehicles can still be steered around obstacles during deceleration.

Other results of ABS systems include a decrease in tyre wear and ensuring consistent braking characteristics on various carriageway surfaces [Transport.ca, 2016].

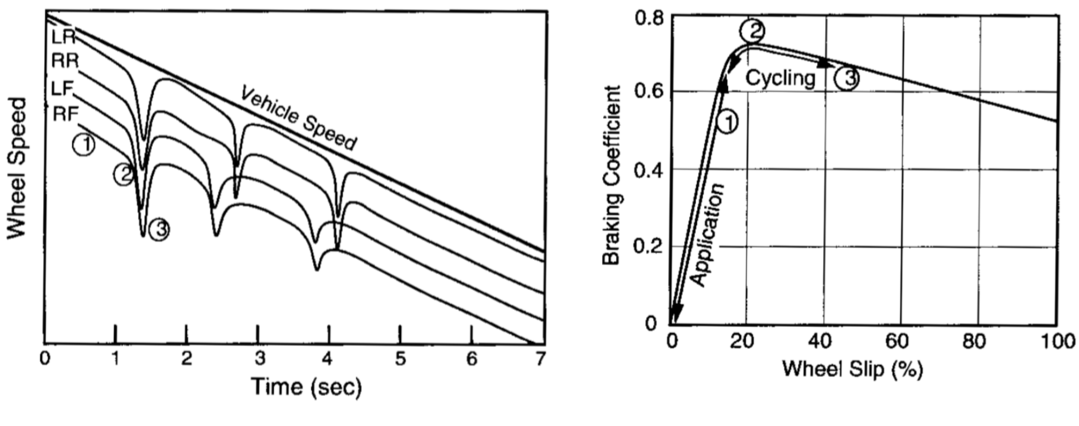

Figure 2.4: Wheel slip versus braking coefficient [Gillespie, 2000]

2.2.3 ABS Components

Figure 2.5 below shows a basic schematic of the components of an ABS system. For simplicity, the components of only one wheel are shown.

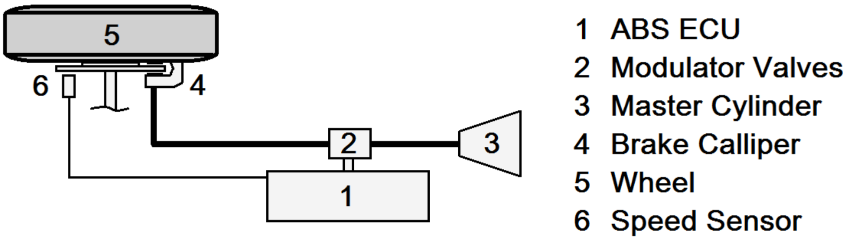

Figure 2.5: Simplified representation of the components of ABS

Figure 2.6 depicts a schematic (adapted from a Wabco ABS schematic [Wabco, 2011]) of the hydraulic lines and modulator valves of an ABS system. The modulator valves, dump reservoir, pump and motor are encapsulated in block 2 of Figure 2.5. As before, only the components of one wheel are shown.

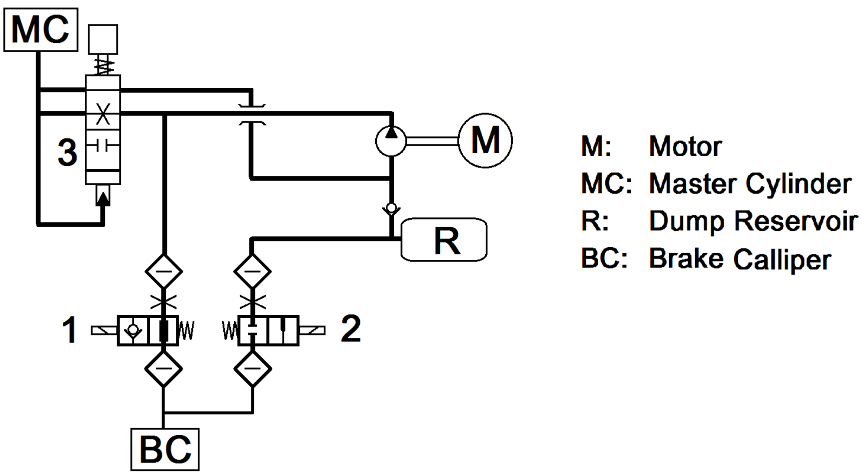

Figure 2.6: Simple schematic of ABS brake hydraulic lines and modulator valves

The solenoids at 1 and 2 of Figure 2.6 are switched independently to activate different ABS phases. These phases are known as pump, dump and hold. With each solenoid either on or off, one of these three configurations can be achieved. The ABS algorithm calculates which phase the ABS should be in and sends these signals to the respective modulator valve's solenoids [1].

2.2.4 ABS Working Principles

As can be seen in Figure 2.4, the ABS system strives to keep the wheels at a certain slip percentage (usually around 20 %). Due to a delayed transient wheel response, the ABS braking process usually cycles between points 1 and 3 continuously until the vehicle reaches standstill.

[1]Diagram adapted from Wabco ABS schematic

The basic flow of procedures is as follows:

1. The ABS ECU measures the wheel speed. From this (and compared to the vehicle speed), various parameters such as wheel acceleration and slip can be determined.
2. Based on these wheel parameters, the ABS algorithm determines the ABS Phase (which translates to the required states at the ABS modulator valves).
3. The ABS Phase (pump, dump or hold) changes the pressure in the brake calliper. This correlates to a change in braking torque which, in turn, allows the wheel acceleration to change.
4. Based on the new wheel speed, the process is repeated until the vehicle reaches standstill.

2.2.5 ABS Algorithms

Of the many ABS algorithms that have been developed over the years, a popular algorithm is known as the HVE Bosch Version 1 ABS Algorithm [Day and Roberts, 2002]. This algorithm is based on wheel accelerations and wheel slip thresholds. The algorithm utilizes a maximum slip threshold as well as a set allowable maximum and minimum wheel acceleration. It uses an 8 phase cycle in which each cycle is a function of the pre-defined thresholds.

Figure 2.7 shows the output of a typical Bosch algorithm . Note the brake pressure relative to the wheel acceleration and wheel velocity as the algorithm cycles through Phase 1 to Phase 8.

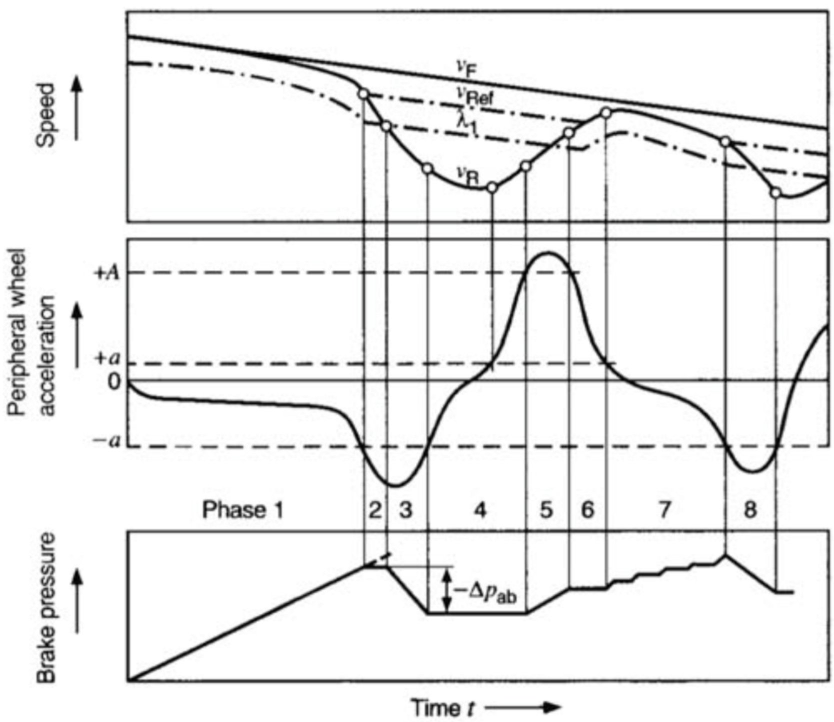

Figure 2.7: Bosch algorithm flow pattern [Day and Roberts, 2002]

2.3 Vehicle Dynamics

2.3.1 Overview

A vehicle under braking experiences a variety of forces applied to it's chassis and other components. In order to determine these forces, a good understanding of the vehicle's . This section considers the theory of a two-wheeled vehicle model which will be used in the HiL simulation setup.

2.3.2 2D Vehicle Model

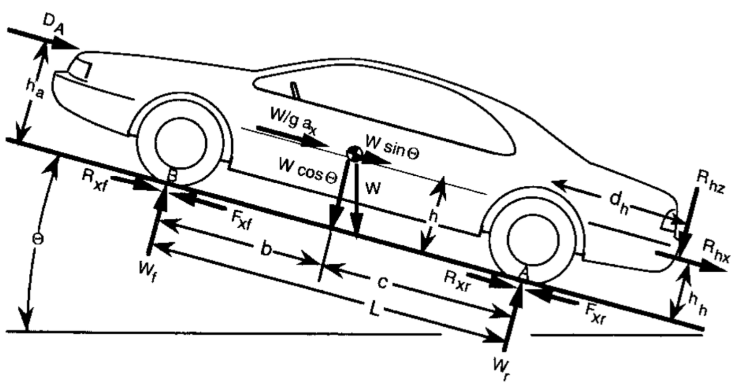

Figure 2.8: Vehicle Free Body Diagram [Gillespie, 2000]

Figure 2.8 depicts a detailed two dimensional free body diagram of the forces acting on an automobile. Ignoring the tow hitch forces, the axle loads W_f and W_r can be determined using Equation 2.1 and 2.2. The aerodynamic drag can be calculated using Equation 2.3.

$$W_f = \frac{Wc\cos(\theta) - \frac{W}{g}a_x h - D_A h_a - Wh\sin(\theta)}{L} \tag{2.1}$$

$$W_r = \frac{Wb\cos(\theta) + \frac{W}{g}a_x h + D_A h_a + Wh\sin(\theta)}{L} \tag{2.2}$$

$$F_D = C_d \rho A \frac{V^2}{2} \tag{2.3}$$

2.3.3 Vehicle Wheel Dynamics

In order to determine the slip of a wheel, we need to look at the tractive forces and braking torques at the wheel. A free body diagram of one of the vehicle's wheels is shown in figure 2.9. The forces and torques acting on the wheel are represented in equation 2.4.

$$-T_b(t) + F_{rolling} r_{wheel} + F_{traction} r_{wheel} = J\frac{\mathrm{d}\omega}{\mathrm{d}t} \tag{2.4}$$

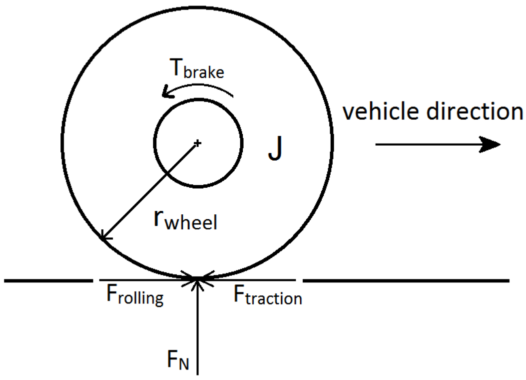

Figure 2.9: Wheel Free Body Diagram

2.3.4 Tyre Models

Tyre models are used to determine the forces and moments experienced by the tyre at the tyre-road interface under different conditions. Tyre models range from the simpler Pajecka model (also known as the magic formula) to dynamic models such as the LuGre model to highly complex three-dimensional models such as the F-tyre model.

Definition of Slip

The tyre friction force is highly dependent on the slip between the tyre and the road surface. Figure 2.10 shows typical friction coefficients as a function of wheel slip for different surfaces. Let the vehicle have velocity v and wheel with rotational velocity ω and wheel radius r. For braking, wheel slip s is defined as [Ayman, 2011]:

$$s = 1 - \frac{\omega r}{v} \tag{2.5}$$

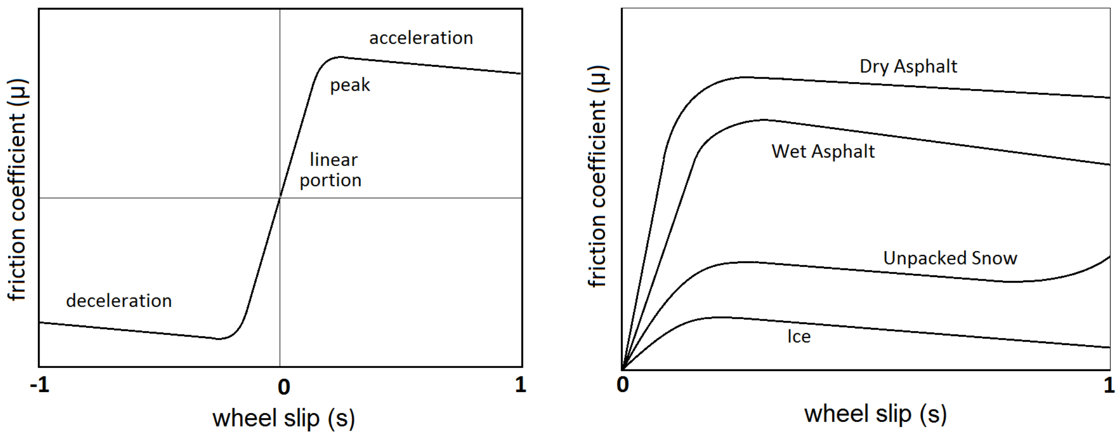

Figure 2.10: Friction Coefficients relative to wheel slip and for different road surfaces

Pajecka '94 Tyre Model

The Pajecka '94 tyre model relates longitudinal force to slip and normal force and is represented by Equation 2.6 [Garcia, 2015]. The coefficients in the equation are determined by using the relations specified in Table 2.1. These coefficients are, in turn, composed of the parameters shown in Table 2.2. Using experimental data for a given tyre, these parameters can be optimized to obtain very accurate tyre models. F is the longitudinal force, Fz is the vertical force and s is the percentage slip.

$$F = D\sin(C\arctan(Bx1 - E(Bx1 - \arctan(Bx1)))) + V \tag{2.6}$$

Coefficient	Name	Formula
C	Shape factor	$C = b0$
D	Peak factor	$D = Fz(b1Fz + b2)$
BCD	Stiffness	$BCD = (b3Fz^2 + b4Fz)e^{-b5Fz}$
B	Stiffness factor	$B = BCD/(CD)$
E	Curvature factor	$E = (b6Fz^2 + b7Fz + b8)(1 - b13\text{sgn}(s + H))$
H	Horizontal shift	$H = b9Fz + b10$
V	Vertical shift	$V = b11Fz + b12$
Bx1	(composite)	$Bx1 = B(s + H)$

Table 2.1: Coefficients of Pajecka formula [Garcia, 2015]

Parameter	Role	Units
b0	Shape factor	
b1	Load influence on longitudinal friction coefficient	1/N
b2	Longitudinal friction factor	
b3	Curvature factor of stiffness/load	$\frac{N}{\%kN^2}$
b4	Change of stiffness with slip	$\frac{N}{\%}$
b5	Change of progressivity of stiffness/load	$1/kN$
b6	Curvature change with load squared	
b7	Curvature change with load	
b8	Curvature factor	
b9	Load influence on horizontal shift	$\%/kN$
b10	Horizontal shift	$\%$
b11	Vertical shift	N
b12	Vertical shift at load = 0	N
b13	Curvature shift	

Table 2.2: Pajecka 94 longitudinal force parameters [Garcia, 2015]

LuGre Tyre Model

In LuGre Tyre Model for HMMWV [Mikkola, 2014], Aki Mikkola discusses various forms of the LuGre tyre model which differ in complexity and accuracy. The LuGre tyre model considered here is a lumped parameter, two dimensional tyre model that takes factors such as stiction, the Stribeck effect, stick slip, zero slip displacement and hysteresis into account. The principle is based on the tyre being treated as a bristle that undergoes deformation under acceleration as shown in Figure 2.11.

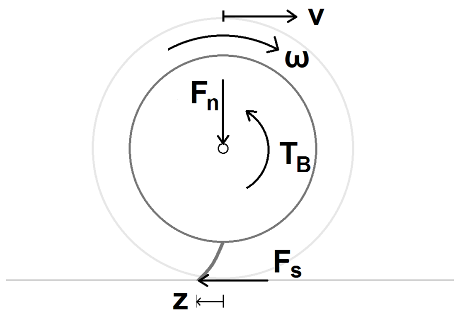

Figure 2.11: LuGre tyre model with tyre deformation z [Mikkola, 2014]

Let z be the bristle deformation, then the deformation of the bristles with respect to time can be expressed as:

$$\frac{dz}{dt} = v_r - \frac{\sigma_0 |v_r|}{F_g} z \tag{2.7}$$

where F_g is the friction denoted in equation 2.8, σ_0 is the rubber longitudinal lumped stiffness and v_r is the relative velocity between the two sliding surfaces defined in equation 2.9.

$$F_g = \mu_c + (\mu_s - \mu_c) \exp\left(- \left| \frac{v_r}{v_s} \right|^{-0.5} \right) \tag{2.8}$$

$$v_r = v - r\omega \tag{2.9}$$

For the aforementioned equations, μ_c is the normalized Coulomb friction, μ_s is the normalized static friction and v_s is the Stribeck relative velocity. The friction force based on the time-dependent bristle deformation in equation 2.7 can be expressed as follows:

$$F_s = \left(\sigma_0 z + \sigma_1 \frac{dz}{dt} + \sigma_2 v_r \right) F_n \tag{2.10}$$

where F_n is the normal force, σ_1 is the longitudinal lumped damping coefficient and σ_2 is the viscous relative damping.

2.3.5 Relating Brake Pressure to Torque

HiL simulation requires a measurement of the brake torque supplied by the callipers. If a relation can be found that relates brake pressure to torque, this parameter can be easily determined. By altering the equations in Shigley's Mechanical Engineering Design [Budynas and Nisbett, 2013], we arrive at Equation 2.11 which can be used to relate braking torque to hydraulic pressure using uniform wear theory.

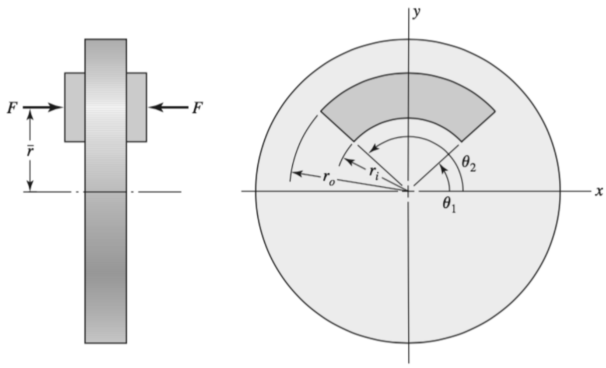

Figure 2.12: Disk Brake Parameters [Budynas and Nisbett, 2013]

$$T_{brake}(t) = \frac{\pi D^2 f(r_o^2 - r_i^2)}{4(r_o - r_i)} P_{hydraulic}(t) \tag{2.11}$$

Where T_{brake} is the braking torque; $P_{hydraulic}$ is the hydraulic pressure at the calliper cylinder; D is the slave cylinder diameter of the brake calliper; f is the friction coefficient between the pad and the disk; r_i and r_o are indicated on the figure as the inner and outer radius of the pad relative to the disk centre.

2.4 Literature Validity and Quality

The literature used in this literature survey comprises of a variety of sources. This includes textbooks such as Thomas Gillespie's Fundamentals of Vehicle Dynamics [Gillespie, 2000] and Shigley's Mechanical Engineering Design [Budynas and Nisbett, 2013]. This also includes journals from world-renowned conferences such as the annual IEEE conference as well as articles from world-class universities such as the University of Darmstadt, Telemark University and the University of Michigan.

While conducting the literature study, careful consideration was taken to ensure that the information obtained was credible and of academic value. This was done by considering the background of the article, the conference name, associated university and the author(s).

2.5 Conclusion

In conclusion, the literature study considered the important components of a successful HiL setup. Additionally, relevant theory was obtained regarding the dynamics of vehicles in braking in order to establish an accurate model later on. The working of automobile ABS was also investigated. This was done while ensuring the use of credible sources as discussed in the previous section.

Simulation

In order to develop a HiL testing setup for ABS brakes, the simulated components of the HiL simulation need to be developed. This includes sophisticated vehicle models, brake models and tyre models.

Ultimately, the goal is to implement a sophisticated three-dimensional tyre model (such as the FTyre model) in the HiL environment. However, seeing as the main objective of this project is to develop a working HiL setup, a simpler model is used. This model will consist of:

1. An inertia-based wheel dynamics model.
2. A lumped parameter LuGre tyre model and a Pajecka '94 (Magic Formula) tyre model.
3. A 2D vehicle model as prescribed in Fundamentals of Vehicle Dynamics [Gillespie, 2000].

3.1 Wheel Dynamics Model

The wheel free body diagram is shown in Figure 3.1. The model is simplified by ignoring rolling resistance.

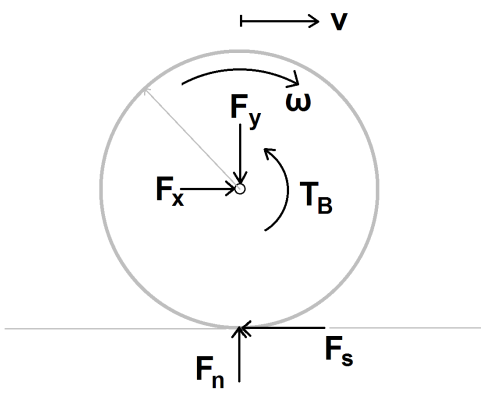

Figure 3.1: Wheel Dynamics Free Body Diagram

The equation governing the wheel motion with inertia J is expressed as:

$$J\frac{d\omega}{dt} = -T_B + F_s r_w \tag{3.1}$$

where T_B is the brake torque, F_s is the friction force applied by the road on the tyre and r_w is the wheel diameter.

3.2 Pajecka '94 and LuGre Tyre Models

By fitting tyre models to experimental data of the tyres used on the target vehicle, both tyre models can be optimized to best represent the vehicle's wheels. A least-squared optimization method is used with a Nelder-Mead solver.

Pajecka '94 Optimization

The definition of the Pajecka '94 model is shown in Section 2.3.4. By considering the equations in Table 2.1, it is noted that some of the coefficients are a function of Fz[2]. This means that a minimum of three different datasets, each at a different load Fz, are required to find a single solution. Since only a single Fz load's data is known, it is proposed that Equation 3.2, a simplified version of the Pajecka '94, be used to model the tyre.

The modified Pajecka formula is shown in equation 3.2. The parameters C_1 to C_4 that best fit the experimental data provided by the VDG are shown in Table 3.1 and Figure 3.2 shows the optimized curve relative to the experimental data [Garcia, 2015].

$$F = C_1 \sin(C_2 \arctan(C_3 s - C_4(C_3 s - \arctan(C_3 s))))F_N \tag{3.2}$$

Parameter	C_1	C_2	C_3	C_4
Value	0.6377	2.3095	6.1244	1.0050

Table 3.1: Pajecka '94 Model Parameters

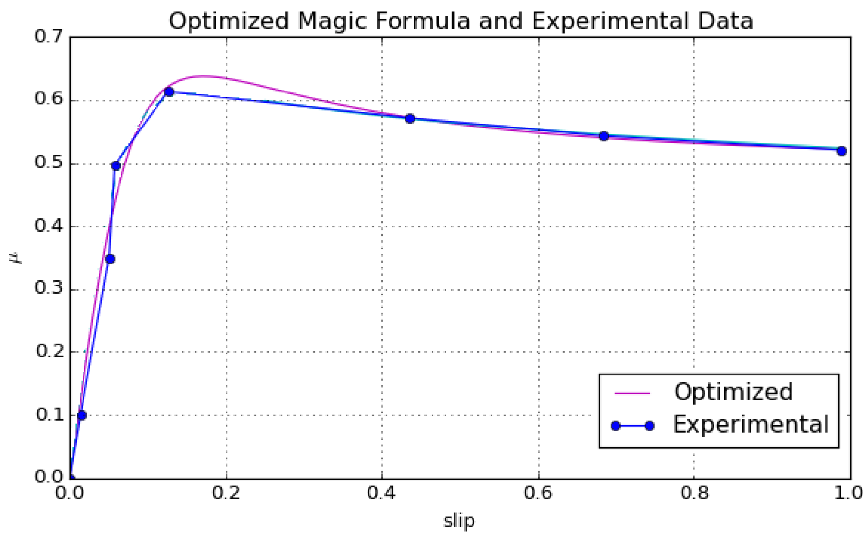

Figure 3.2: Optimized modified Pajecka curve and experimental data

LuGre Model Optimization

Since the LuGre model is velocity dependent, the experimental data was used to fit the LuGre model at a vehicle velocity of $5\ \mathrm{ms}^{-1}$. This is because the experimental data was obtained at low velocity ($\approx 5\ \mathrm{ms}^{-1}$). The optimized LuGre model parameters are shown in Table 3.2. Figure 3.3 shows the experimental data and the LuGre tyre model with the given parameters as well as the LuGre curves for higher vehicle velocities.

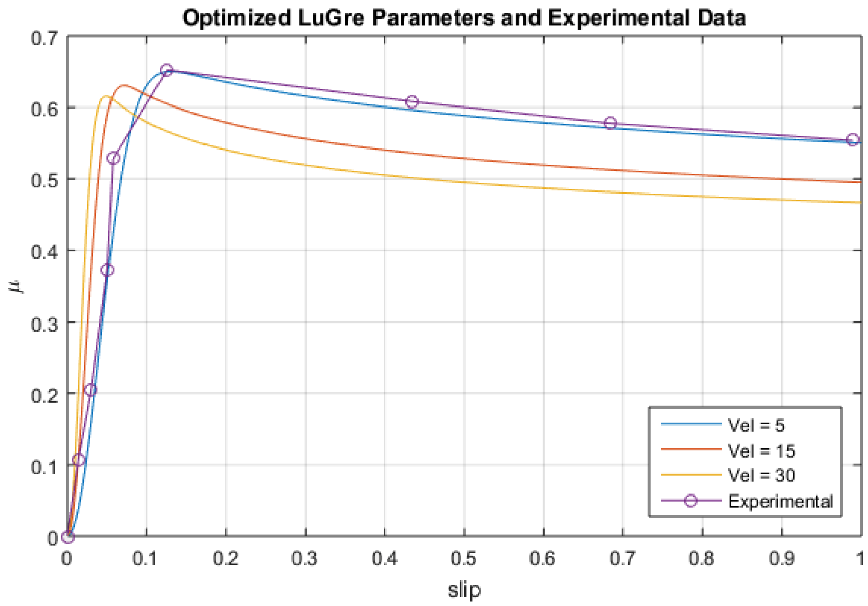

Figure 3.3: Optimized modified LuGre curve and experimental data

Parameter	σ_0	σ_1	σ_2	μ_c	μ_s	v_s
value	83	0.01	0.001	0.68	0.38	3.61

Table 3.2: LuGre Model Parameters

Reason For Two Tyre Models

The reason that two tyre models are proposed for simulation is explained in the discussion of Section 2.1.5. The convergence of mathematical models is critical for a successful simulation which means a minimum step size is required. However, with real-time simulations, available computing power limits the model's complexity the allowable step size.

Since the Pajecka '94 model does not have a convergence criteria, the simulation step size is not limited by it. Hence, a successful real-time simulation is more probable with the available computing resources. The LuGre model, however, has a minimum required step size (found to be around 0.5 ms) to ensure convergence. If the available computing resources do not allow for real-time simulation higher than 1kHz, the Pajecka '94 model could allow a slower simulation rate.

3.3 2D Vehicle Model

As described in the Literature study, a simple 2D vehicle model is implemented for the vehicle model. This model takes variable axle load, basic aerodynamics and road slope into account [1] This model does not take vehicle yaw movement into account. Despite this, the vehicle model is configured for four wheels which would allow for a model to take yaw motion into account in future work. The Land Rover's geometric parameters used in this model are shown in Appendix F.

3.4 ABS Algorithm

The simulation will utilize a modified Bosch ABS algorithm as developed by the VDG. This algorithm comprises of 8 different phases based on wheel slip and wheel rotational acceleration. These 8 phases each coincide with a pump, dump or hold input to the ABS modulator.

3.5 Simulation Results

The Bosch ABS algorithm was run in conjunction with the aforementioned tyre and vehicle models and yielded the results shown in Figures 3.4 through 3.7 [2].

Figure 3.4 shows the velocity curve of the vehicle as well as each of the four wheels. It can be noted how the wheels tend to lock up and then oscillate at a slip value of approximately 20% until the vehicle reaches standstill.

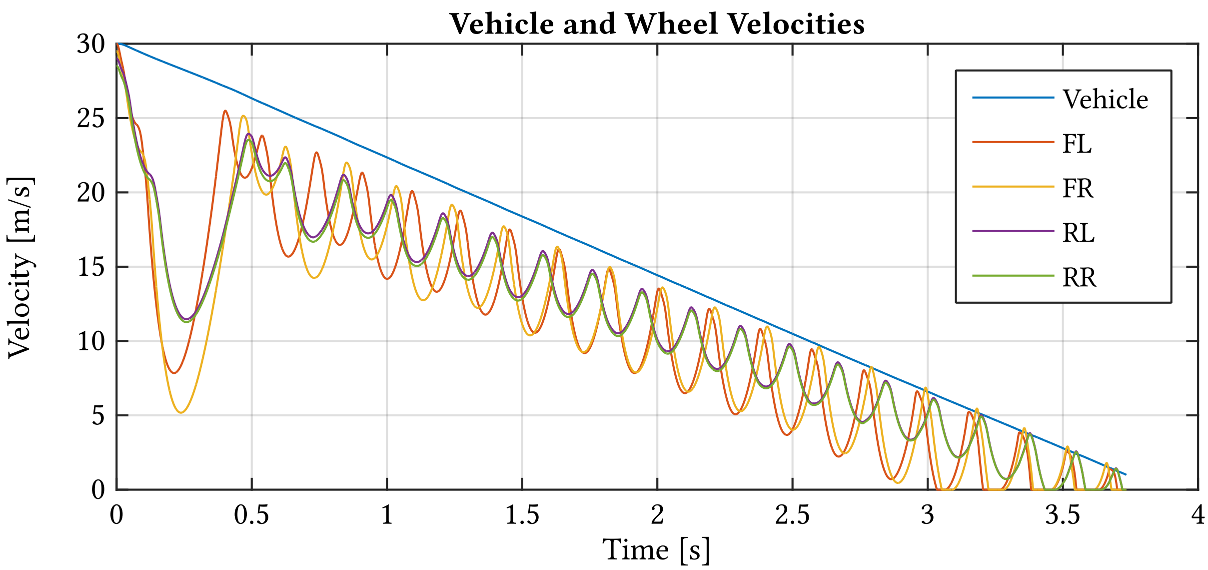

Figure 3.4: Simulated wheel and vehicle velocities

[1]The equations pertaining to the vehicle model are discussed in Section and in Appendix F
[2]Shorthand for wheel specification: Front Left (FL), Front Right (FR), Rear Left (RL), Rear Right (RR)

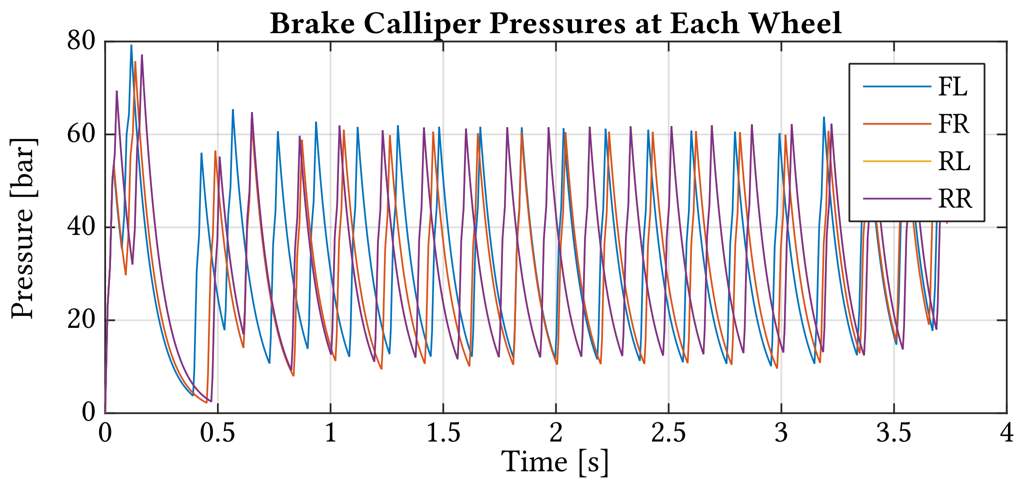

Figure 3.5: Simulated brake pressures

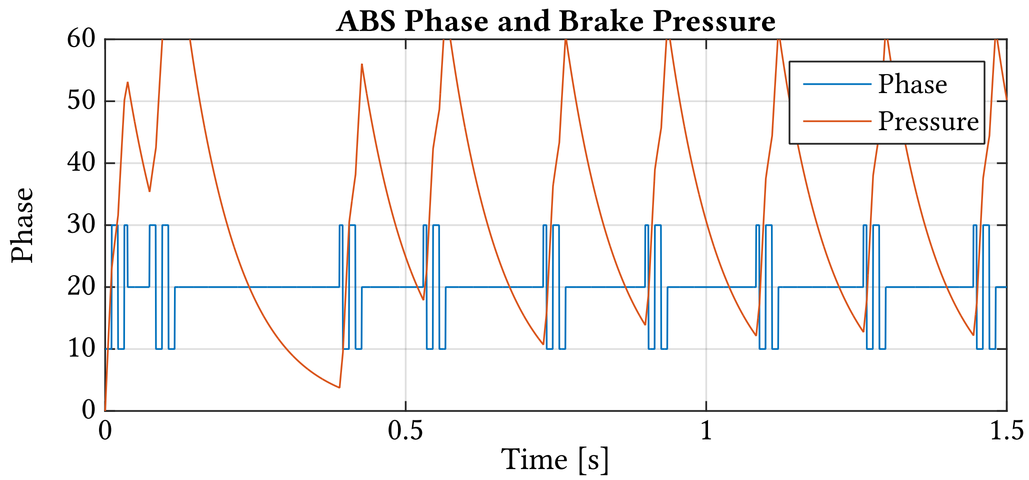

Figure 3.6: Front left wheel ABS phase and corresponding pressure

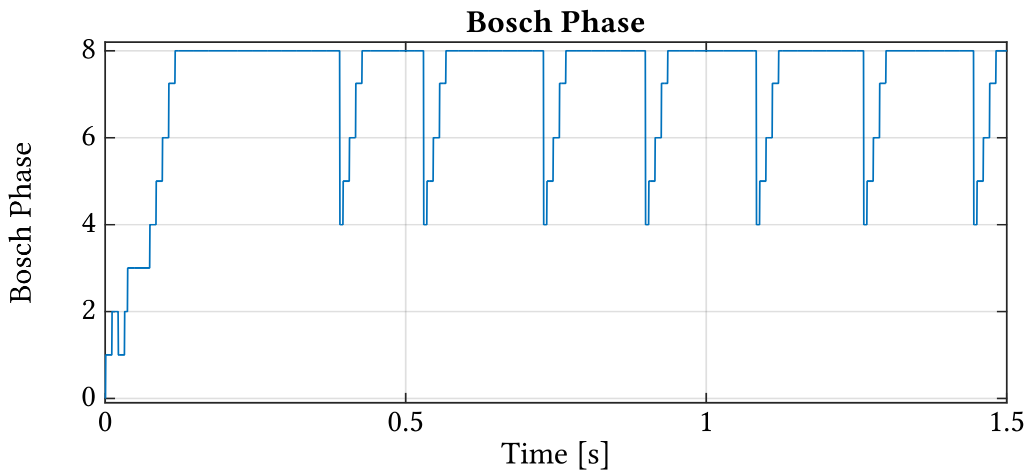

Figure 3.7: Bosch algorithm phases

3.6 Conclusion

This chapter proposed a simple vehicle braking model consisting of a two-dimensional LuGre tyre model, an inertia-based wheel dynamics model and a two-dimensional vehicle dynamics model. A modified Bosch ABS algorithm was tuned to work with the model. This model was implemented in the SimuLink© environment and validated by considering the deceleration of the vehicle under extreme braking conditions (potential wheel-lock). Finally, the results were shown of the simulated vehicle deceleration, ABS modulator phases, ABS algorithm phases as well as the brake pressure.

Synthesis to HiL Environment

In order to create a HiL setup, a variety of items need to be addressed such as the ability of the simulation to communicate with physical hardware through sensors and actuators. This chapter discusses the requirements that need to be met and method to be followed in order to synthesize the simulation developed in Chapter 3 into a HiL setup.

4.1 Simulation Setup Conversion

Figure 4.1 below shows a simplified block diagram of a pure simulation based model and the equivalent HiL model. The Simulation diagram was implemented in SimuLink© as discussed in Chapter 3. The ultimate goal is to convert the simulation into the HiL Setup.

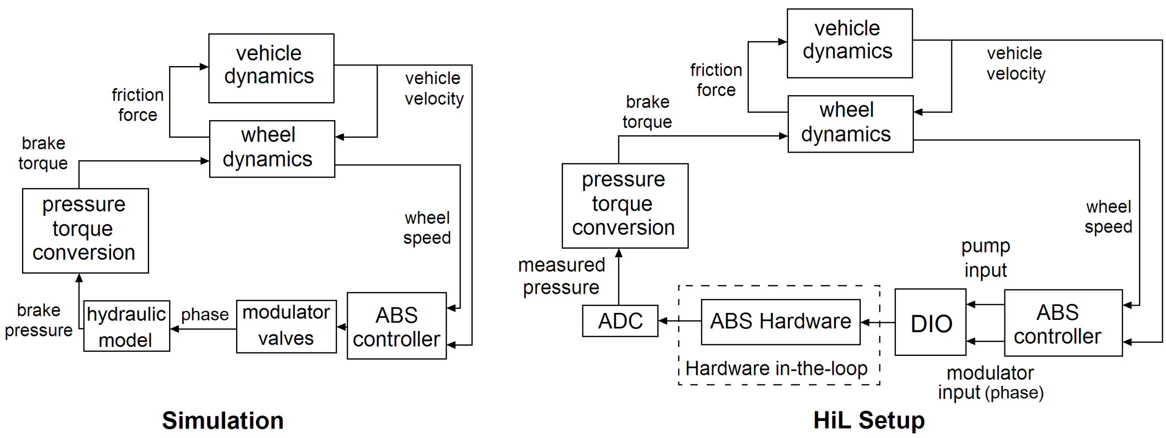

Figure 4.1: Full simulation and HiL comparison

4.2 Determining Inputs and Outputs

The ABS system comprises of the Electronic Control Unit (ECU) and the physical hardware such as the modulator valves and piping systems to each wheel as discussed in Section 2.2.3. This section delves into the required inputs and outputs of each of these sub-systems such that the interface between hardware and software can be fully defined. This process of determining these software-hardware interfaces is a very important aspect of the HiL simulation as discussed in section 2.1.3 item 1.

4.2.1 Inputs

Seeing as the ABS controller will be simulated using the Bosch ABS algorithm, the only inputs required by the ABS hardware are the modulator phases for each wheel, as well as a possible input to activate an actuator to apply the brake pressure. The inputs are:

1. ABS Phase FL[1]
2. ABS Phase FR
3. ABS Phase RL
4. ABS Phase RR
5. Pump State
6. Actuator Input

4.2.2 Outputs

After the hardware receives the modulator valve phases for each wheel and the brake actuator input, the brake pressure will change. The individual brake pressures at each wheel are the measured outputs of the hardware. The individual brake pressures will be measured using pressure sensors and fed back into the simulation. The outputs are:

1. Pressure FL
2. Pressure FR
3. Pressure RL
4. Pressure RR
5. Shutter Valve [2]

4.3 Conclusion

This section delved into the synthesis between a simulation based exclusively on mathematical models and a HiL simulation. The portion of the simulation that would be replaced by physical hardware and the interface between the hardware and the simulation was discussed. Additionally, the hardware's inputs and outputs to the simulation were specified and an indication of the data package size required for streaming was given.

[1]Shorthand for wheel specification: Front Left (FL), Front Right (FR), Rear Left (RL), Rear Right (RR)
[2]The shutter valve is a logic sensor that is triggered by applying the brake pedal

HiL Implementation

Chapter 4 revealed the synthesis of a HiL simulation from a purely mathematically based simulation environment. This section aims at considering the actual implementation of the HiL simulation. Important topics discussed in this section are the HiL components used as well as the communication mediums used between the different HiL setup components specific to this system.

5.1 HiL Setup and Components

Figure 5.1 below shows the components required to connect the simulation to the required sensors to measure the relevant parameters of the physical hardware.

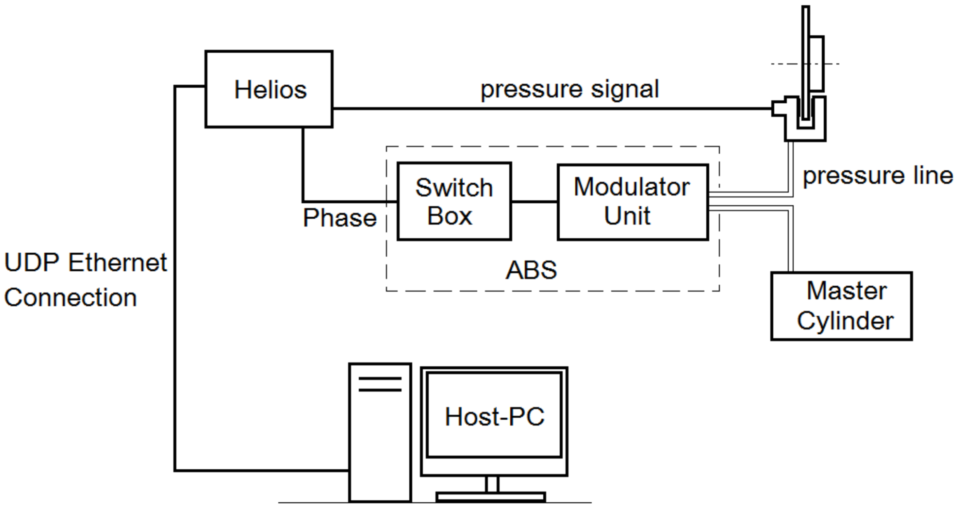

Figure 5.1: HiL Setup

Each component shown in Figure 5.1 is required to implement the connection between the hardware's physical variables we are interested in and the simulation. Each of the relevant components are discussed below.

5.1.1 SimuLink© Real-Time and UDP Ethernet Connection

In order to run a HiL simulation, a real-time computing environment is needed. SimuLink© has a library called SimuLink© Real-Time which provides a simulated real-time interface between a host computer and external hardware. For this application, the real-time communication will take place using a UDP (User Datagram Protocol) Ethernet connection between the Helios embedded computer and the host computer.

Since UDP connections do not perform any hand-shaking, data packets are occasionally lost, meaning that the data sent and received by the simulation is momentarily interpreted as zero. This can be detrimental to the simulation. To overcome this problem, a simple method of message integrity was written to overcome this momentary loss of data. This method is explained in Section 5.2.2 .

5.1.2 Helios Embedded Computer

The Helios board is an embedded computer used primarily for data acquisition due to its numerous DIO and ADC ports [Diamond, 2016]. The Helios is installed in the target vehicle and connected to the ABS modulator valve inputs and pressure sensors at the brake callipers as well as additional ports such as the actuator switch, ABS pump input and shutter valve.

The Helios board runs Linux and is programmed with C++ to perform tasks such as:

1. Establish UDP connection through its Ethernet port
2. Read the pressure sensors with it's ADC
3. Send the ABS phases, pump status and brake actuator state to the relevant hardware using its DIO

5.1.3 ABS Switch Box

The ABS Switch Box can be seen as the interface between the low-voltage digital I/O pins sent from the Helios and the higher voltage signals that are required by the ABS Modulator Unit. This component basically comprises of relays and switches which allow a low-current, low-voltage (max. 20mA and 3.3 V) signal to be converted to the required 12 V signal.

In order to ensure the safety of the Helios board, an opto-coupler switch was used for this specific application. This ensures that no reverse current from the solenoid coils can harm the DIO of the Helios.

5.1.4 ABS Modulator Unit

The ABS modulator unit is the physical hardware comprising of the solenoid valves connected to each wheel's brake pressure line. The Modulator Unit receives the signals for each of its internal components from the ABS Switch Box.

5.1.5 Pressure to Torque Conversion

Previous work was done with ABS systems on the same target vehicle (Land Rover Defender 1997 model) as used for this HiL setup. This work was done by the VDG and documents the pressure to torque relationship as $T = (271\mathrm{N\,m/MPa})P_{\text{calliper}}$. This information was determined experimentally using a wheel-force transducer [Penny and Els, 2016] and is used in the HiL simulation.

5.2 Communication

Communication between the host computer and the Helios embedded computer will take place using the UDP Ethernet communication protocol. The input and output data and it's requirements are summarized in Tables 5.1 and 5.2 respectively.

5.2.1 Data Types and Structure

Parameter	Possible values	Data type	size
ABS phase FL	0;1;2;3	uint8_t	1 byte
ABS phase FR	0;1;2;3	uint8_t	1 byte
ABS phase RL	0;1;2;3	uint8_t	1 byte
ABS phase RR	0;1;2;3	uint8_t	1 byte
ABS pump state	binary	uint8_t	1 byte
Actuator input	real value	single	4 bytes

Table 5.1: Hardware inputs and data-types

Parameter	Possible values	Data type	size
Pressure FL	real value	single	4 bytes
Pressure FR	real value	single	4 bytes
Pressure RL	real value	single	4 bytes
Pressure RR	real value	single	4 bytes
Shutter Valve	binary	uint8_t	1 byte

Table 5.2: Hardware outputs and data-types

5.2.2 UDP Connection Fidelity

UDP connections are fast and ideal for transfer of non-secure data. However, since no handshaking takes place between the sender and the receiver, it is not clear whether or not the data has been successfully received. When data is unsuccessfully received, this is known as 'dropping' a packet. Dropping a packet causes the receiver to receive a value of zero instead of the actual sent value. Naturally, this can be detrimental to the simulation.

To overcome this, a simple message fidelity structure was put in place. This method works as follows:

Each UDP sender sends two predetermined constant values to the receiver. The receiver knows what these constant values are and checks whether or not the message contains these values. If it does, it can be assumed that this particular data packet was successfully received and that the rest of the data is reliable and can be used in the simulation. If, however, the message does not contain these constants, it can be assumed that the packet was lost. If this happens, the other variables are assigned their previous value until a new successful data packet is received.

5.2.3 Sampling Frequency

The sampling frequency of both UDP computers is important. If the sampling frequencies differ, the rate at which UDP packets are sent is also different. When this is the case, the computer with the lower frequency starts lagging behind the higher frequency messages and real-time communication is compromised.

If systems must run at different sampling frequencies, it is advised to implement a buffer overflow clearer to ensure that the most recent message is read. If this is not done, the message buffer will simply increase in size and the one computer will increase in lag behind the other.

5.3 Methodology

The method used to run the experiments is simple. It consist of the following steps:

1. Supply power to Helios and ABS Switch Box
2. Establish connection between host computer and Helios [1]
3. Run SimuLink© model, ensuring successful UDP communication
4. Activate the actuator to start ABS braking

[1]This can be done using Putty.exe as a tool to SSH into the Helios and WinSCP to obtain files on the Helios

Results and Discussion

This chapter serves to represent the results obtained from various HiL tests. These tests were done to confirm the HiL setup for future use. Section 6.1 aims at confirming the HiL setup by showing a successful HiL implementation and results. The successful switching of the modulator valves is also illustrated with experimental data. Section 6.2 provides the data for the two HiL tests. Section 6.3 is a discussion of the obtained results.

6.1 Confirmation of HiL Setup

This section aims at illustrating the success of the implemented HiL test. Figure 6.1 shows the ABS phase [1] sent to the front left wheel brake line as well as the pressure at the front left brake calliper.

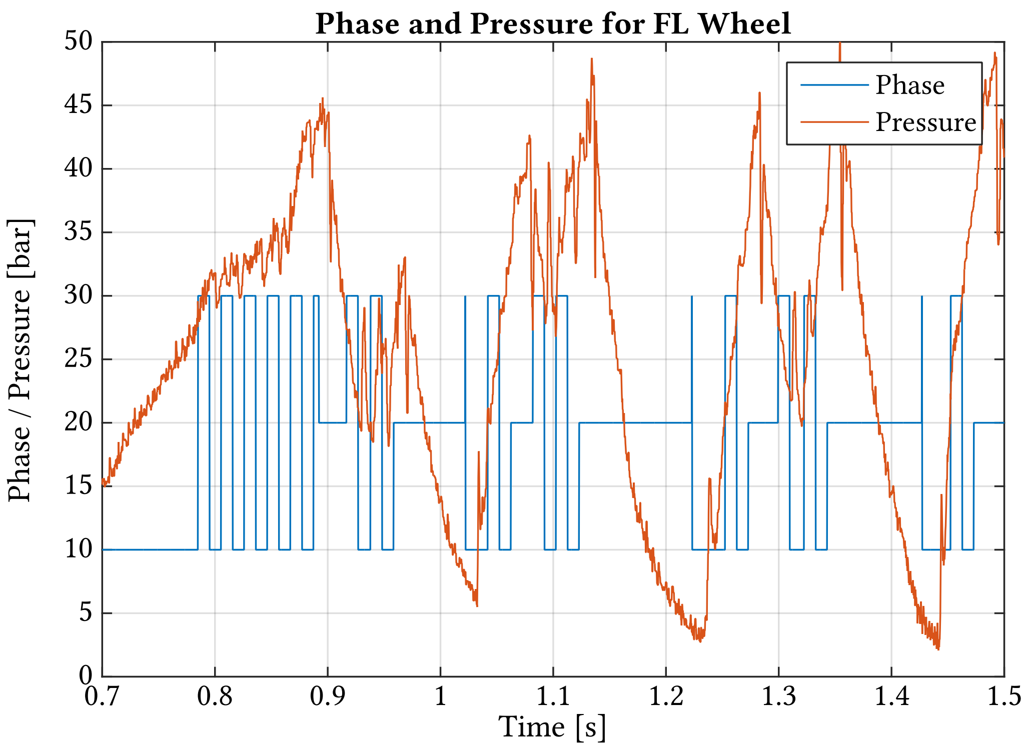

Figure 6.1: FL Wheel Brake Pressure and Phase

[1]ABS phases are: 10 - pump; 20 - dump; 30 - hold

Figures 6.2 and 6.3 show the missed ticks for the HiL simulation using the modified Pajecka and LuGre models respectively. The relevance of missed ticks is discussed in Section 2.1.5.

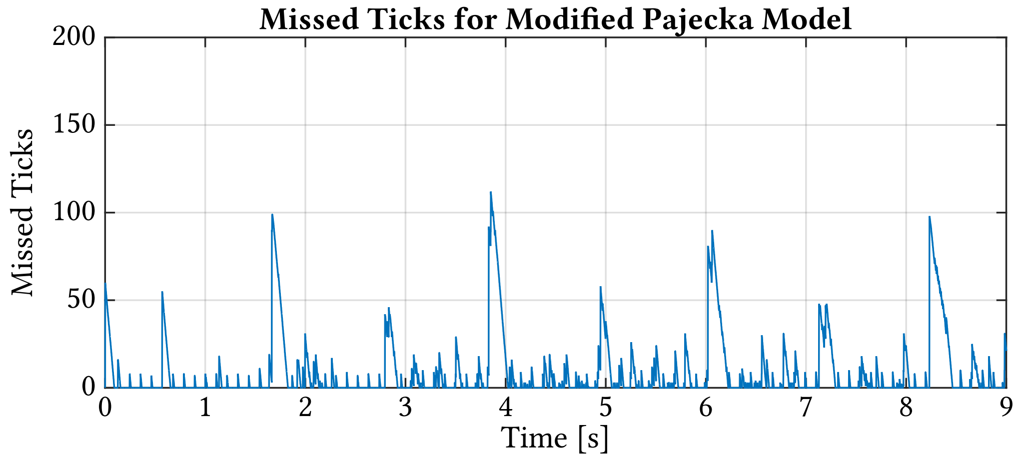

Figure 6.2: Missed ticks for modified Pajecka HiL test

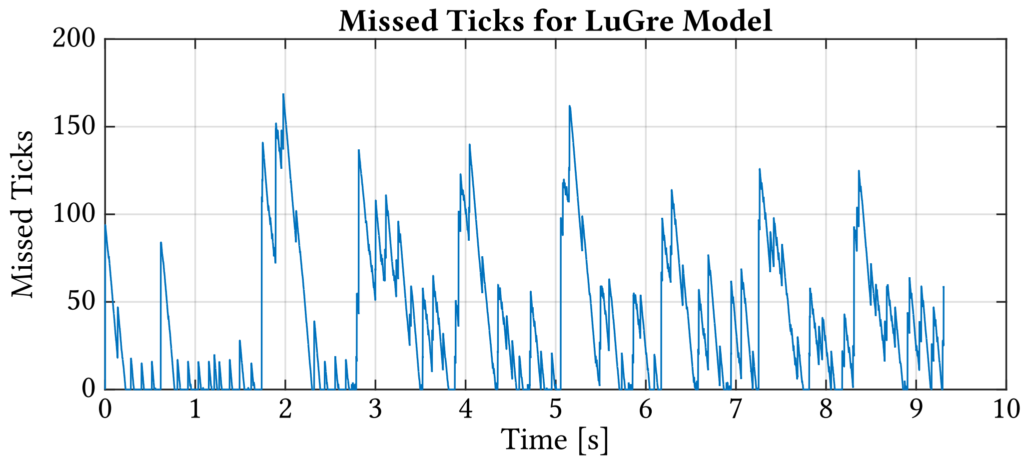

Figure 6.3: Missed ticks for LuGre HiL test

6.2 HiL with SimuLink© Real-Time

This section shows the results following the implementation of HiL simulation of ABS brakes using either a modified Pajecka tyre model in Section 6.2.1 and the LuGre tyre model in Section 6.2.2. The pressure graphs are enlarged along the abscissa since the finer detail is more important in this case. The graphs look similar in the cropped sections and it is deemed more important to visibly see the pump, dump and hold sequences.

The modified Bosch Algorithm parameters used are tabulated in Table 6.1.

Parameter	s_{max}	**A**	α_{max}	α_{min}
Value	0.10	50	10	-30

Table 6.1: Bosch Algorithm Parameters

6.2.1 Modified Pajecka Model

Figure 6.4 shows the velocity curve [1]. Figure 6.5 shows the pressure graph.

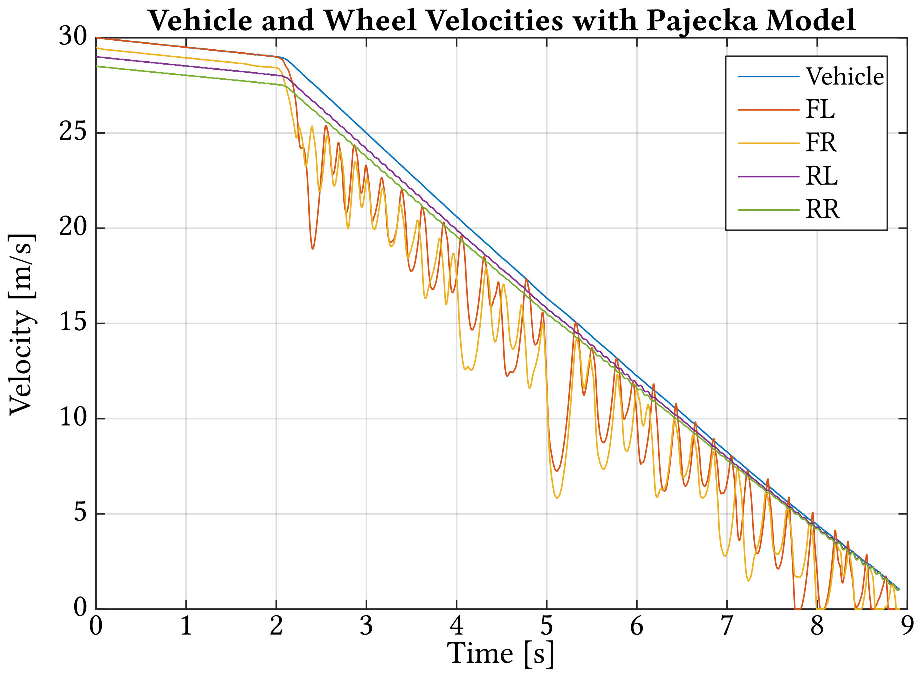

Figure 6.4: Wheel and vehicle velocities for modified Pajecka model

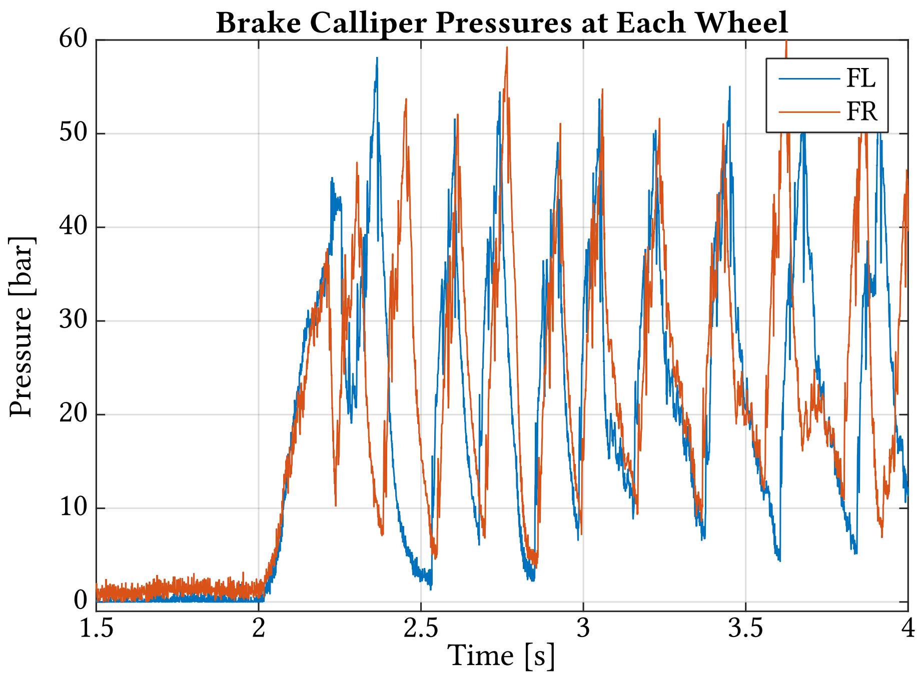

Figure 6.5: Brake pressure for modified Pajecka model

[1] Velocities are offset to ensure clarity

6.2.2 LuGre Tyre Model

Figure 6.6 shows the velocity curve [1]. Figure 6.7 shows the pressure graph.

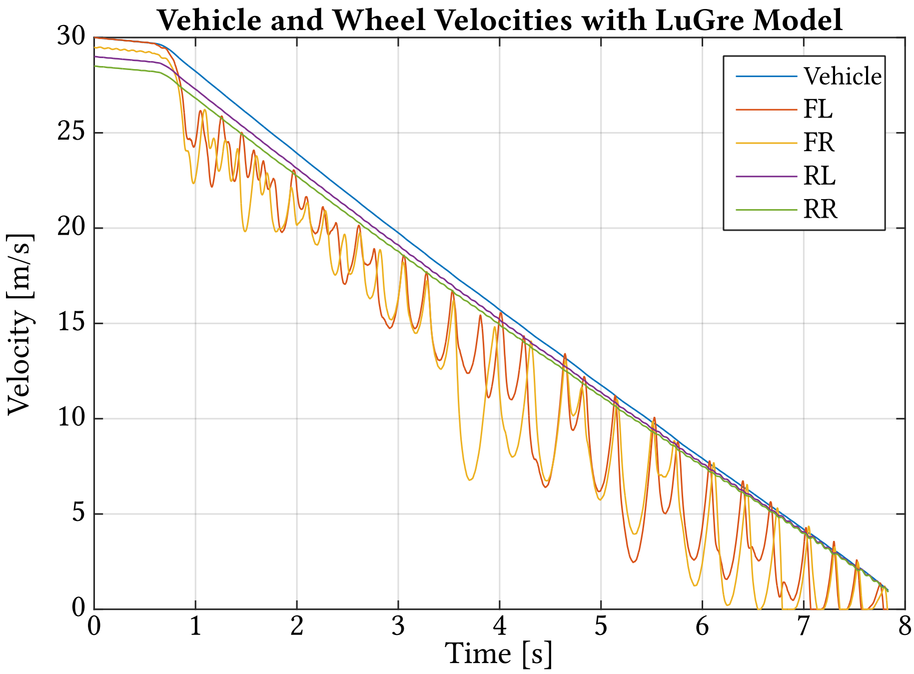

Figure 6.6: Wheel and vehicle velocities for modified LuGre model

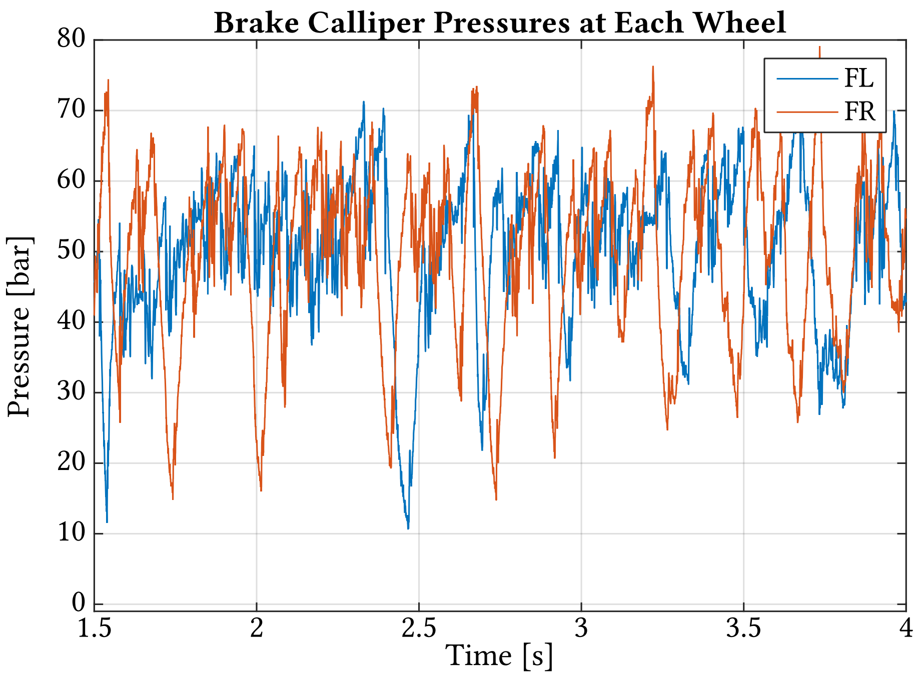

Figure 6.7: Brake pressure for modified LuGre model

[1] Velocities are offset to ensure clarity

6.2.3 Reason for Results of Only Two Wheels

It can be observed that the results show that ABS was only activated for the front two wheels. The reason for this is that the ABS modulator unit and ABS switch box malfunctioned (due to reasons unrelated to this project's testing). HiL simulation was done with all four wheels successfully, however, this data was not recorded and can therefore not be reproduced in this report.

However, the simulation proved to work and, were the modulator unit and switch box working properly, a successful four-wheel HiL test could be conducted.

6.3 Discussion of Results

HiL Confirmation

Figure 6.1 shows the relation between the front left wheel's modulator valve signal and brake calliper pressure. It is clear that pump, dump and hold signals correspond to an increased, decreased and constant pressure, albeit with a small time delay. This data confirms that the HiL setup works as desired. To ensure brevity, the other wheel's data is not shown. However, it was confirmed that all four wheels act in a similar manner.

General

Having established a successful HiL setup, it is possible to perform HiL tests of ABS hardware. Figure 6.4 and 6.6 show successful HiL simulation using ABS. The wheel velocities are centred around 10% slip which is the setting of the Bosch algorithm used in the tests.

In order to ensure a successful HiL test, the amount of missed ticks needs to monitored and possibly minimized to ensure that the simulation is running in real-time. A missed tick refers to a case where the simulation time for that time step exceeds the simulation step size. This causes the simulation to lag behind the hardware which, for HiL testing, can rend the results to be meaningless.

Regarding the modified Pajecka '94 and LuGre models, the host-PC used for this HiL test managed to run both tyre models successfully without the missed ticks exceeding 165. 165 missed ticks refers to a maximum momentous delay of 82.5ms (when run at 2kHz). By comparing the HiL test's velocity graphs shown in Figure 6.4 and 6.6 to the simulation's velocity graphs shown in Figure 3.4, it is clear that this delay does not affect the HiL setup's implementation.

Comparison with Simulation

There is quite a large difference between the initial wheel velocities of the velocity curve of the simulation, shown in Figure 3.4, and the velocity curves of the HiL setup of Figure 6.4 and 6.6. The simulation shows a much larger deceleration of the wheels than the HiL results. The reason for this is attributed to the initial braking force applied.

This is attributed to the following:

- In simulation, the initial braking is simulated as a step function and the brake pressure jumps straight to 70 bar. Because of the immediate pressure rise, the ABS algorithm does not have time to react quick enough and the wheel speeds decrease dramatically.
- In the HiL test, the braking was performed by physically applying full force on the brake pedal. However, this is clearly not a step function but is more similar to a first order response. During this time, the ABS algorithm has enough time to react and decrease the initial wheel deceleration.

Conclusion of Results

In summary, results were obtained by running the developed HiL simulation setup. These results were compared to simulation and found to yield proof of a successful HiL implementation.

Due to external factors, the ABS Modulator Unit and ABS Switch Box were damaged. However, testing was done on the four wheel setup (but just not recorded) and yielded successful results. What this means is that, if the necessary hardware was replaced, the Land Rover's ABS could be tested on all four wheels with the current SimuLink© setup.

In conclusion, the required HiL setup was developed and proven to work successfully, allowing the VDG to perform the required testing of ABS algorithms in the future.

Conclusions and Recommendations

7.1 Summary

The overriding purpose of this study was to implement an HiL simulation of an automobile's ABS. To accomplish this goal it was necessary to perform the following:

- Reach a good understanding of what HiL simulation entails and what requirements need to be adhered to in order to ensure successful HiL simulation. Additionally, there is a need for accurate models of the vehicle, tyres and ABS algorithms. This was performed in Chapter 2.
- Develop a simulation using the chosen vehicle and tyre models which can be used in the HiL setup. Additionally, tests of the simulation must be run to ensure the model performs as required. This was documented in Chapter 3.
- Determine the required changes that need to be made to convert the simulation into an HiL platform. This entails the identification of the hardware-software interface and what data needs to be transferred along this interface. This was specified in Chapter 4.
- Having established the required changes to the simulation, it is necessary to specify exactly how these changes will be implemented. A Helios embedded computer was chosen to interface with the sensors and actuators on the automobile and the UDP communication protocol was used to ensure communication between the host computer and the Helios. This was documented in Chapter 5.
- Finally, validation of the HiL setup is done. Once it is verified that the HiL setup is functional and accurate, braking tests can be performed and compared to the simulated results. This was done in Chapter 6.

7.2 Conclusions

The HiL validation and braking tests performed in Chapter 6 proved to be successful. As discussed in Section 6.3, the obtained results proved the successful implementation of an HiL setup.

In conclusion, the goal of implementing a successful HiL setup for the VDG was achieved. Using the results of this project, ABS algorithms can be validated using physical hardware.

Despite the fact that test data was only available for ABS braking with two wheels, the HiL setup proved capable of working with four wheels. A four wheel braking test was successfully performed numerous times. However, the data was not recorded and could therefore not be reproduced in this report. Due to an unrelated malfunction, both the ABS modulator unit and the ABS switch box were damaged leaving only the circuitry related to the two front wheels intact. If this damage was repaired, a four wheel braking test could be performed and recorded with the developed HiL setup. Due to time constraints, however, this was not possible.

The following is a list of noteworthy observations that were made which could largely contribute to the recreation of this project:

1. The sampling frequency of the Helios and the SimuLink© model must be the same. If this is not done, a buffer overflow will occur at the slower device resulting in a communication lag between the two devices which, ultimately, renders the HiL test useless. Alternatively, an automatic buffer-overflow clearer must be implemented on the slower computer to ensure that the most recent message packet is read.
2. Switching frequency of the ABS algorithm highly affects the capability of the modulator valve's coils to respond as requested by the ABS algorithm. Since the modulator valves consist of solenoids, they have a significant response time. If the ABS algorithm is allowed to be switched too fast, it was found that the coils drained excessive power from the battery causing the whole ABS Switch Box to malfunction. This essentially yields a very poor wheel response where it locks up for one second and then coasts for one second repeatedly until the car reaches standstill. It was found that a 40 ms ABS algorithm step time was the minimum before the system would seize and essentially stop working.
3. The missed ticks of the simulation need to be monitored to ensure the simulation was run in real time. This is of paramount importance regardless of the host computer or RTOS used.

7.3 Recommendations

The HiL setup developed in this report opens up a wide array of potential topics of research in vehicle control and braking. Important considerations for future work are listed:

1. By obtaining more comprehensive data of the target vehicle's tyres, a better parametrisation of the tyre models can be done in order to improve the accuracy of the HiL setup.
2. As discussed in Section 3.2, by obtaining three separate data sets at three different wheel loads for the vehicle's target vehicle, the coefficients for the complete '94 Pajecka tyre model can be determined. This model could then be used as the tyre model for this simulation.
3. Additional aspects such as vehicle yaw control could potentially be tested if a suitable simulation model were to be developed. This could also yield valuable data regarding the success of ABS algorithms since ensuring vehicle control is a primary concern when developing ABS algorithms.

4. By implementing the HiL setup on a real-time platform such as the dSPACE MicroAutobox II [dSPACE.de, 2016], more complex mathematical models could be analysed in real time if necessary. Alternatively, the current models could be run at a higher sampling rate allowing for increased accuracy of current results.

Bibliography

[Anakwa et al., 2001] Anakwa, W., Cohen, E., Naik, A., Carlton, D., Glen, D., and Lopez, J. (2001). Tools for rapid prototyping of embedded control systems.

[Automobile, 2016] Automobile, C. (2016). Land rover defender 110 1997 model.

[Ayman, 2011] Ayman, A. A. (2011). An antilock-braking system (abs) control: A technical review. *Intelligent Control and Automation.*

[Budynas and Nisbett, 2013] Budynas, R. and Nisbett, J. (2013). *Shigley's Mechanical Engineering Design.* McGraw-Hill, ninth edition (si units) edition.

[Day and Roberts, 2002] Day, T. and Roberts, S. (2002). A simulation model for vehicle braking systems fitted with abs. *SAE Internationl.*

[Diamond, 2016] Diamond, S. (2016). Helios pc/104 sbc. online.

[dSPACE.de, 2016] dSPACE.de (2016). dspace webpage. online webpage.

[Ecomodder.com, 2016] Ecomodder.com (2016). Cars compared in wind tunnel. online.

[Fathy et al., 2006] Fathy, H. K., Filipi, Z. S., Hagena, J., and Stein, J. L. (2006). Review of hardware-in-the-loop simulation and its prospects in the automotive area. Technical Report 1, University of Michigan.

[Garcia, 2015] Garcia, A. (2015). Pajecka '94 parameters explained - a comprehensive guide. online.

[Gillespie, 2000] Gillespie, T. D. (2000). Fundamentals of vehicle dynamics. Technical report, Society of Automotive Engineers, Inc., 400 Commonwealth Drive, Warrendale, PA 15096-0001.

[Gomez, 2001] Gomez, M. (2001). Hardware-in-the-loop simulation.

[Halvorsen, 2014] Halvorsen, H. P. (2014). Hardware-in-the-loop simulation.

[Isermann et al., 1999] Isermann, R., Schaffnit, J., and Sinsel, S. (1999). Hardware-in-the-loop simulation for the design and testing of engine control systems. In *Control Engineering Practice*, volume 7, Issue 5, pages 643–653, Landgraf-Georg-Str 4, Darmstadt, Germany. Elsevier.

[Kay, 2012] Kay, J. (2012). Introduction to real-time systems.

[Lee and Suh, 1999] Lee, J. C. and Suh, M. W. (1999). Hardware-in-the loop simulator for abs/tcs. In *Proceedings of the 1999 IEEE International Conference on Control Applications*, volume 1, pages 652–657, Kohala Coast, HI. IEEE.

[Liu, 2000] Liu, J. (2000). Real-time systems.

[Mathworks.com, 2016] Mathworks.com (2016). Simulink real-time. online.

[Mikkola, 2014] Mikkola, A. (2014). Lugre tire model for hmmwv. Technical report.

[Penny and Els, 2016] Penny, W. and Els, P. (2016). The test and simulation of abs on rough, non-deformable terrains. Technical report, University of Pretoria.

[Sun et al., 2016] Sun, W., Cai, X., and Meng, Q. (2016). Testing flight software on the ground: Introducing the hardware-in-the-loop simulation method to the alpha magnetic spectrometer on the international space station.

[Svaricek, 1999] Svaricek, F. (1999). Automatic valuation and verification of abs controllers by using a hardware-in-the-loop simulation.

[Transport.ca, 2016] Transport.ca (2016). Abs braking and stability control. webpage.

[Wabco, 2011] Wabco, A. (2011). *Anti-Lock Braking Systems (ABS) and Anti-Slip Regulation (ASR)*. WABCO Vehicle Control Systems, www.wabco-auto.com, 2 edition.