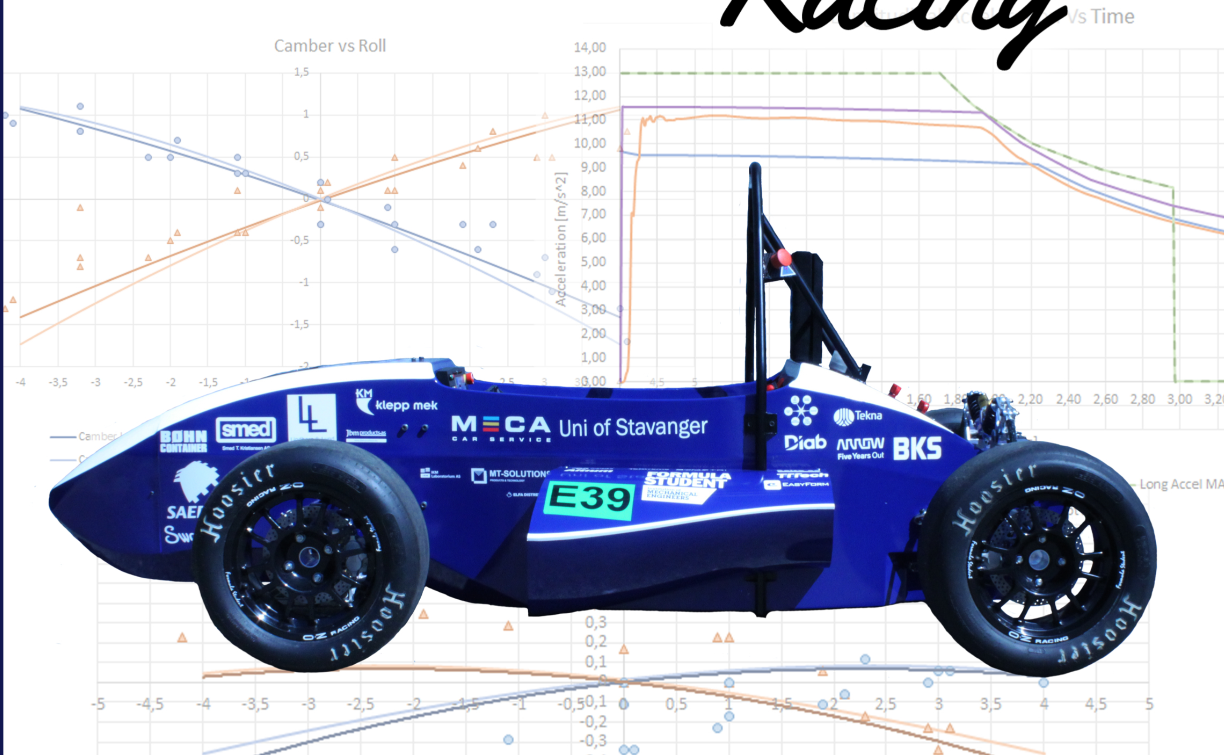
Universitetet
i Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study programme/specialisation: **Engineering structures and Materials**	Spring semester, 2018 Open/~~Confidential~~
Author: **Tor Inge Berge**	Tor Inge Berge (signature of author)
Programme coordinator: **Hirpa G. Lemu** Supervisor(s):	
Title of master's thesis: **Comparative Study and Simulation of Suspension System Using Vehicle Dynamics Simulation Tools**	
Credits: **30 points**	
Keywords: Formula Student, ION Racing, Multi-Body Dynamics, ADAMS/Car, OptimumG, MATLAB, Vehicle Suspension, Vehicle Dynamics, Simulation,	Number of pages: 72 + supplemental material/other: 55 Stavanger, 14/06-2018 date/year

Title page for Master's Thesis
Faculty of Science and Technology



Comparative Study and Simulation of Suspension System Using Vehicly Dynamics Simulation Tools

By Tor Inge Berge

Summary

With the increased availability of computing power, simulation tools has become more available for the average user. This has led to a lot of specialized software for most simulation needs. In Formula Student vehicle dynamics simulation software is used to determine the handling of the car being designed, before it is built. This gives the students a possibility to try different solution on the computer, before choosing which one to use on the real car. This thesis will study the results from three different vehicle dynamics simulation software. These results will be compared with each other and measured values to see if the simulations give realistic results. The software chosen for this study is:

- MATLAB
- OptimumG Software package
- ADAMS/Car

These three software have a different degree of complexity and skill level needed to setup and run simulations. As time is the most valuable commodity in the Formula Student design process, easy and fast setup and simulations is important. The tools will be compared based on the generated results and the time needed to setup and simulate the models. As a reference three simulations will be done:

- Kinematic suspension motion
- Roll Stiffness
- Acceleration event

Kinematic suspension motion and roll stiffness will base their measured results on Bifrost, while the acceleration event will base its reference on results from previous Formula Student competitions. This is because Bifrost have not been tested in acceleration at the time of writing. Each of the software bases their simulation inputs on ION Racing UiS's 2018 car Bifrost. By using the same source of input the results will only vary based on how the software solves the simulation. By comparing the results with measurements collected from the actual vehicle, comparisons on the accuracy of the simulations can be done.

From the kinematic simulations and measured results, the simulations did not deviate more than the uncertainty in the measured results. The toe measurements in heave and roll motion was deviating from the simulations. This is most likely because of the small variations in toe in both heave and roll, and that the test rig was not stiff enough to give consistent measurements at the same resolution as the simulation. The roll stiffness in both calculated and simulated values was close to the measured value. There was some difference in the rear roll stiffness in the simulation, but it was not possible to measure this, as the test rig was only able to measure total roll stiffness. In the acceleration simulation, all of the simulated values was lower than what have been performed by top teams in the competition with a rear wheel drive car. The best times in acceleration at competition is 4.1 sec. The results from the simulations varied from 3.5-3.9 sec. Most likely this is a result of inaccurate inputs, which again gives inaccurate outputs from the software.

With this knowledge of how the simulated results compare to real world tests, better understanding of the results is achieved and better decisions can be made in the future.

Preface

This thesis concludes my time as a student at the University of Stavanger and an active member of ION Racing UiS. It focuses on one of my main interests when it comes to vehicles, namely the dynamics of the vehicle. By being able to simulate and calculate how a vehicle handles before it is even built, is an exciting process. When the car is finished and measurements are done and they show close relation to what is simulated is really satisfying. Everything done in this thesis has been done in the spring semester of 2018. In addition to writing the thesis this semester, most of the time spent at the university have been used to manufacture parts for the suspension on the car, assembling the car, and helping other members of ION Racing with their work. This have been a time consuming process and have affected the amount of time spent on the master thesis. But still, I am happy with the result and hope it will be useful for teams in the future when running vehicle dynamics simulations.

ION Racing have been my main activity besides the time spent in lectures since I started at UiS in the fall of 2012. It have taught me a lot about design and manufacturing which is not possible with just lectures. In addition it have also taught me about team work and how to manage a team of inexperienced students through the process of developing a vehicle capable of competing in Formula Student.

I would like to thank Hirpa G. Lemu firstly for being a driving force behind starting ION Racing, and secondly for being my supervisor for both my bachelor and masters thesis. I would like to thank Audun Brønseth for cooperation with the design of the suspension and damping system. I would also like to thank the team for cooperation and a good working environment through the year. And in the end I would like to thank my family and friends which have supported me, and yet barely seen me the past couple of years.

Table of Contents

List of Figures

List of Tables

Abbreviations

ARB	=	Anti-Roll Bar
CAD	=	Computer Aided Design
CG	=	Center of Gravity
DoF	=	Degrees of Freedom
FEM	=	Finite Element Analysis
FSAE	=	Formula Society of Automotive Engineers
FSAETTC	=	Formula Society of Automotive Engineers Tire Test Consortium
RPM	=	Revolutions pr Minute
SPMM	=	Suspension Parameter Measuring Machine
UiS	=	Universitetet i Stavanger/University of Stavanger

Chapter 1

Introduction

Vehicle suspension and behaviour have developed drastically in the last 30 years. With the possibility to design and evaluate a vehicle on computers, it has been cheaper for car manufacturers to develop new a model [2]. By developing a computer model of the car, simulations and virtual tests can be conducted. This has removed the need to build test prototypes to evaluate the designs, which in turn decreases the cost in the design and development phase. The vehicle model can also be integrated with vehicle handling software to simulate how anti skid systems or other driver aids responds to the car before prototypes have been made.

This thesis will mainly focus on racing car handling and performance of a Formula Student car. This is a light weight vehicle designed to run on tight twisting tracks at speeds up to 100 km/h. As the design and manufacturing time of these cars are about 9 months, time needed to setup and do simulations are critical. By running simulations with different setups, decisions can be done based on what setups give the best performance for the car. Therefore the ease of use and the time it takes to get reliable results from the simulation software is important. The thesis will look at different ways of simulating the performance of the Formula Student car and comparing how much time is spent on the development of a vehicle model and simulation, and how accurate the results are. It is assumed that the readers have some knowledge of terms and expressions used when discussing properties of a vehicle, and few in-depth explanation of terms will be given. It is recommended to read Race Car Vehicle Dynamics [3] for further in-depth explanation terms which are unknown.

1.1 Background

This thesis will be written while manufacturing and testing ION Racing's 2018 car. It was designed in the fall of 2017 and production of the car started in the end of January 2018. As the car is ready to drive when this thesis is complete, the results will not effect this years car in any major way. The results can instead be used as a guide for next years team when deciding on what software to use and how to setup their simulations. In this section some background information about the Formula Student competition and ION Racing will be given.

1.1.1 Formula Student

Formula Student is one of the world's largest engineering competitions for students. The objective for the teams is to design and build a single seater race car for a weekend racer. More than 500 universities around the world competes against each other on different competitions around the world. The official list of competitions are listed in Table 1.1. In addition to this, there are various non-official competitions following the FSAE or Formula Student Germany ruleset around the world where teams can compete.

The first competitions where held in the 1980's and where based in the US [4]. Only a hand full of universities participated and the rules where pretty simple compared to the ones used today. The first European competition was held in UK in 1998. Since 2006 Formula Student Germany have been held on the Hockenheim Ring and in the last 8 years, many more European competitions have emerged. As in other motor sport, the rules are

Table 1.1: Table over official FSAE competitions.

Competition	**Country**
Formula SAE Australasia	Australia
Formula Student Austria	Austria
Frmula SAE Brasil	Brazil
Formula North - Canada	Canada
Formula Student Czech Republic	Czech Republic
Formula Student Germany	Germany
Formula Student East	Hungary
Formula SAE Italy	Italy
Student Formula Japan	Japan
Formula Student Spain	Spain
Formula Student United Kingdom	United Kingdom
Formula SAE Lincoln	United States
Formula SAE Michigan	United States

in a constant evolution. As mentioned, the rules for the first competitions was pretty open, but over the years more rules have been added to mainly increase the safety of the cars. As the rules focus on the safety, there are a lot of possibilities to design a unique car. Both electric or internal combustion engines are allowed, and there are no restrictions on drive type. At the moment the top teams develop electric cars with one motor in each wheel giving a four wheel drive system with full control of the torque on each wheel. In the internal combustion cars both 4 cylinders and single cylinder engines are used, but, with a few exceptions, they are only rear wheel drive. This limits their available traction compared to the electric cars, but with good drivers and a good developed car, these can still compete for the top 10 spots in the competitions.

1.1.2 ION Racing UiS

ION Racing started back in 2011 as Formula Student Team UiS. The first car was a combustion car and competed in the 2012 FSUK competition. For the 2013 competition a new car was built based on the design from 2012, but with improvements in suspension and an aero package. For the 2014 season the choice was made to make the change from combustion cars to electric drive train and also develop a carbon fiber monocoque instead of a steel frame. It was decided to change the name of the organization from Formula Student Team UiS to ION Racing UiS. For the 2015 season the same design with 2 electric motors and carbon fiber monocoque was used but with some changes. The motors and battery pack was mounted inside the monocoque instead of in the wheels and side pods as on the 2014 car. In addition it was decided to add an aerodynamic package to increase the downforce of the car. For the 2016 season some simplifications was done to the driveline as there where trouble getting the two motors to run in a desirable manner. It was chosen to use only one electric motor which would save about 20 kg of weight and would make it easier to control the motor. This was the first electric car to run at the competition, but it did not manage to pass the brake test.

For the 2017 season, it was decided to focus mainly on reliability and simplicity. The main changes was the removal of the aero package and improved drivetrain. The car was done before the competition, but had some electrical bugs which reduced the testing time. At the competition some mechanical problems together with some unresolved electrical bugs, prevented the car to compete in most of the dynamic events except the endurance. It was able to complete one and a half round before the battery management system shut down the battery. This was most probably because of interference from the motor controller and not a battery problem. It also won a prize from Mercedes AMG High Performance Powertrains for best electric drive train. For this year it is planned to use most of the designs from last year, with some tweaks for performance gains, and fix apparent flaws. The suspension and damping systems has been one of the main changes, as last years design had problems with bumpsteer and understeer. The damping system at the front of the car have been moved down to the sides to make space more space for the driver. The goal is to compete in all events at the competition and achieve a top 20 position over all.

1.2 Scope

The goal for this thesis, is to test and compare the results from different simulation tools for vehicle dynamics. This is done to see if there are major differences in results from the different analytic tools. To be able to compare the tools, some realistic simulation cases will be used. These are described in more detail in the Simulations chapter. As a base vehicle for all the simulations, ION Racing's 2018 car Bifrost will be used. To verify the results, measurements on Bifrost and results from previous Formula Student competitions will be used. The tools chosen are available for the team through university licences or after purchase of a license, and have been used by the team. These are:

- MATLAB
- OptimumG
- ADAMS

Each of the software will be discussed in the Methodology chapter, together with a short explanation on how the measurements on the finished car was conducted.

Chapter 2

Theory

This chapter will go through the theory behind Multi-body dynamics. The main focus will be on Lagrangian formulations and how they are used to develop expressions for the movement of bodies. When analyzing multi-body dynamics of a system, it is needed to create a model of the system. This model is a mathematical abstraction of the physical system which can mathematically describe how the system works. This is done by dividing the system into fundamental elements and derive mathematical expressions for these. The 3 basic fundamental elements are:

- Mass/Inertia elements
- Energy storage elements (Springs)
- Energy dissipation elements (Dampers)

As reality is a complex system with many variables that are hard to define, models are usually simplified to ease the amount of calculation needed. By defining what the important aspects in the system is, and what is going to be analyzed, decisions on how complex the model should be, can be made. For a motion system, it should be able to model the overall movement of a system. An example could be to look at the movement of the chassis on a car. Here the model can have a simplified version of the suspension system on each corner, as only the movement of the chassis is important. If it is desirable to know more about the movement of one of the wishbones in the suspension, it would be needed to make a more complex model which can describe the suspension in more detail.

2.1 Equations of motion for mechanical systems

When modeling mechanical systems, the derivation of differential equations is an important part. This is usually based on either Newton-Euler or Lagrangian formulations [1]. Both of these systems use the same starting point with a mechanical system, and give the same results, as long as the same generalized coordinates are chosen. The difference is in how the equations of motion are derived. Figure 2.1 show how the generation of these equations is done with the Newton-Euler and Lagrange method.

In this thesis, the focus will be on the Lagrange method as this is the one mostly used in multi-body simulation software [2].

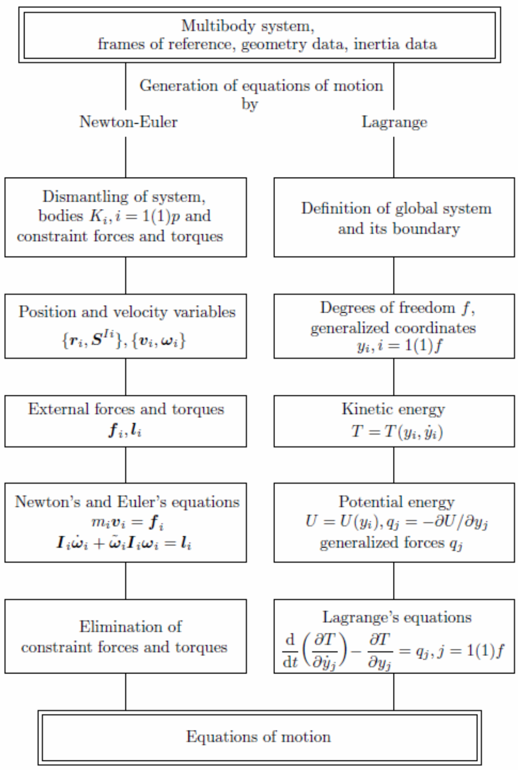


Figure 2.1: Generation of equations of motion by the Newton-Euler and Lagrange method [1]

2.2 Lagrange

Joseph-Louis Lagrange was a mathematician and astronomer born in 1736 Turin, Italy [5]. He started studying mathematics at the age of 16, by the age of 19 he was appointed a professorship at the Royal Artillery School in Turin. He was chosen, after recommendations by Euler, to succeed as Euler as the director of the Berlin Academy. In 1788 Lagrange published his Mecanique Analytique. It presents a general formulae to provide the necessary equations for solving a mechanical problem. This formulation has advantages over using Newton-Euler formulation:

- It directly generates as many equations as there are DoF
- It uses generalized coordinates instead of Cartesian coordinates which Newton-Euler uses.
- It eliminates non-contributing forces, as the generalized forces only include components in the direction of motion.

The formula used for multi-body dynamics is what is called the equation for Lagrangian mechanics as shown in Equation (2.1).

$$\frac{d}{dt}\frac{\delta T}{\delta \dot{q}_j} - \frac{d}{dt}\frac{\delta V}{\delta \dot{q}_j} - \frac{\delta T}{\delta q_j} + \frac{\delta V}{\delta q_j} + \frac{\delta R}{\delta \dot{q}_j} = Q_j \tag{2.1}$$

where:

- T : The systems kinetic energy
- V : The systems potential energy
- R : The systems damping
- q_j : The generalized coordinates
- Q_j : The generalized forces

Generalized coordinates are coordinates which are chosen depending on what is best suited to describe the motion of an object. If an object rotates, angular coordinates are chosen, or if it slides, translational coordinates are more suited. This reduces the amount of DoF needed to define the system, which in turn reduces the amount of equations needed. A pendulum defined with Newton-Euler needs 2 DoF to define the motion of the pendulum, while Lagrange only needs one. The generalized coordinates has a few requirements. They must be independent, meaning if you fix all but one coordinate, the system still have a continuous range of motion in the free coordinate. They must be complete, meaning they are capable of locating all parts at all times. The system must also be holonomic. This means that the number of DoF need to be equal to the number of generalized coordinates. If this is not possible, the Lagrange equation can not be used [6]. The coordinates can be of any kind as long as the above criteria are satisfied. For pendulums or rotating bodies, angular coordinates are more useful, while for planar motion, Cartesian coordinates are useful. For mechanical systems the term $\frac{d}{dt}\frac{\delta V}{\delta \dot{q}_j} = 0$ as the potential energy in the system is not a function of velocity or acceleration. The resulting equation can then be written as 2.2.

$$\frac{d}{dt}\frac{\delta T}{\delta \dot{q}_j} - \frac{\delta T}{\delta q_j} + \frac{\delta V}{\delta q_j} + \frac{\delta R}{\delta \dot{q}_j} = Q_j \tag{2.2}$$

By defining the kinetic and potential energy of the bodies in the system, equations for T and V can be derived. Ususal terms under the potential energy are:

- $m_i g h_i$: Gravitational potential energy
- $\frac{1}{2}k_i x_i^2$: Spring potential energy

where m_i is the mass of a body i, h_i is the height of a body i above a reference plane, g is the gravitational acceleration, k_i is the spring stiffness of spring i and x_i is the compression/extension of spring i. This gives the following potential energy equation:

$$V = \sum_i m_i g h_i + \sum_i \frac{1}{2} k_i x_i^2 \tag{2.3}$$

Terms that usually are a part of the kinetic energy are:

- $\frac{1}{2}I_i\dot{\theta}_i^2$: Rotational kinetic energy
- $\frac{1}{2}m_i v_i^2$: Translational kinetic energy

where I_i is the moment of inertia of a body i, $\dot{\theta}_i$ is the angular velocity of a body i, v_i is the velocity of a body i. This gives the following kinetic energy equation:

$$T = \sum_i \frac{1}{2} I_i \dot{\theta}_i^2 + \sum_i \frac{1}{2} m_i v_i^2 \tag{2.4}$$

In addition to the kinetic and potential energies, damping is also included. Here both friction and viscous damping can be included. For the damping, the energy equation can be written as:

$$R = \frac{1}{2} c \dot{x}^2 \tag{2.5}$$

where c is the damping coefficient.

The above equations have to be defined for each DOF and then combined into Equation 2.2. When solving it manually it is often useful to calculate the left and right hand separately and then combine the results. Reference [6] describe the following approach. Left hand side:

1. Determine number of DOF and choose coordinates q_j to be used

2. Verify that the system is Complete, Independent and Holonomic
3. Compute T+V for all bodies
4. Compute each term on the left had side one by one for all q_j

Right hand side:

1. For each q_j, find Q_j that goes with it
2. Computing virtual work δW associated with the virtual displacement δq_j

$$\delta W_j = Q_j \delta_j \tag{2.6}$$

Chapter 3

Methodology

In this chapter the simulation tools used in the report will be discussed. The tools used are:

- MATLAB
- OptimumG Software Package
- ADAMS/Car

Each software will be presented and some of their features will be discussed.

3.1 MATLAB

MATLAB is a versatile programming software widely used in the industry. It is mainly built as a programming platform for engineers and scientists. It uses a matrix based language which allows for ”natural expressions of computational mathematics” [7]. In MATLAB it is possible to analyze data from large data sets, develop algorithms to validate concepts or designs and create models and applications. In addition to the standard functions following MATLAB there are a wide variety of toolboxes which expand the software with more functionality. There are toolboxes for Finance, Math, Statistics, Physical modeling, Robotics, Biology and a lot more. These toolboxes makes it easier to work on their specific tasks by adding functions tailored for each category.

As MATLAB is a programming software, it is up to the user to develop codes which can be used to run simulations or process the data as needed. This lets the user have full control in how the software do the calculations, but it also requires the user to understand what is needed and how to do the math behind the simulation.

3.2 Optimum Software Package

OptimumG is a consulting company for racing teams and have developed 4 different software which focuses on model and simulation of vehicle behaviour. These are:

- OptimumLap
- OptimumTire
- OptimumKinematics
- OptimumDynamics

The overall focus of the different software, is to have an easy understanding on what to do and how to do it to get results fast. Each of the 4 software focuses on different aspects when simulating vehicle behaviour.

3.2.1 OptimumLap

OptimumLap (OptimumL) is a lap time simulator which brakes down the car into 10 basic parameters [8]. Each of these parameters represents a specific aspect of the car. This makes it easy to evaluate what effect each aspect have on the overall performance of a car. With a car defined by its parameters, lap time simulations can be run on tracks. These tracks can be accessed from the OptimumL database, created with logged data or manually written into the OptimumL Track Maker. The vehicle model used in the simulation is a quasi-steady state mode where the vehicle is assumed to be a point mass. It is a simple, but accurate enough to analyze global performance trends. OptimumG claims that the simulations, by comparing to logged data of total lap time and speeds, are close to 10% and in some cases 5% of the logged data. The last part of the software is the analysis. This is a post processing tool where data from the simulation can be viewed and plotted into tables, plots or visualized on the given track.

3.2.2 OptimumKinematics

OptimumKinematics (OptimumK) is used to design and simulate suspensions based on kinematic [9]. Here the movement of the suspension is most important and forces like springs and dampers are neglected. It bases its suspension designs on templates which the user can choose from, and then position the hard points needed to define the wanted suspension. The available templates are:

- Double A-Arm
- MacPherson
- MacPherson Pivot Arm (Front Only)

- Five Links (Rear Only)
- Live Axle, 2 A-Arms (Rear Only)
- Live Axle, 2 Trailing Arms w/ Panhard Bar (Rear Only)
- Live Axle, 4 Trailing Arms w/ Watts Linkage (Rear Only)

In addition to the suspension layouts there are also different spring actuations, push/pull-rod positions and anti-roll bar systems to choose from to further customize the design the users needs. To simulate the suspension movement a motion cycle can be applied or logged data can be imported and used in the simulation. The motions available are chassis heave, roll and pitch, and steering wheel angle. It is possible to run simulations with multiple iterations where changes have been made on the vehicle setup between each iteration to look at the differences these changes makes. Like in OptimumL there is an analysis tool for OptimumK. Here tables, plots or math channels can be created or the motion can be visualized with animation.

3.2.3 OptimumDynamics

OptimumDynamics (OptimumD) is used to run dynamic simulations on a vehicle [10]. It got mostly the same features as OptimumK, but uses forces instead of motion as the driving factor in the simulations. Before a simulation can be run the user have to define a vehicle setup. This setup have to contain a tire with a stiffness and force model, a chassis, coilover with springs, suspension, brakes and a drivetrain. Each of these parts, have different choices depending on how accurate the results need to be. A tire stiffness can be a linear stiffness or a non-linear, and the force model can be a friction value or a full tire model. The same goes for springs. For the suspension the user can import a suspension from OptimumK or design a new one in OptimumD with the same tools, or choose a simple linear suspension. Brakes only define the brake balance and the drivetrain define the engine torque, gear ratios and if it is a front, rear or all-wheel drive vehicle.

Further components can be added, like ARB's, bump stops and aerodynamics. The simulation forces can be applied in different ways, eader as a single constant force, or varying forces or a Yaw moment diagram. By applying varying forces in the longitudinal and lateral direction, combined with speed or position based on logged data, it is possible to recreate a track map and do visualize the changes in the vehicle behavior throughout the track. OptimumD also got the same visualization tools as OptimumK to visualize the results.

3.2.4 OptimumTire

OptimumTire (OptimumT) differs from the other 3 software as it is not a simulation software, and more a data handling and visualization software. It is designed to use raw tire testing data and derive a full tire model based on this data. It is designed to easily fit a tire model to raw data and verify the results by plotting the data together with the fitted curve. The full tire model can then be used in simulations in OptimumD or converted into ADAMS/Car or other vehicle simulation software to improve the simulations. The most common tire model called Pacejka Magic Formula is included together with a few other less common, but useful tire models. By fitting curves to the data step by step for lateral, longitudinal and moments. When all the fitted curves are done, a friction ellipse can be created to see the longitudinal, lateral and the combined friction coefficient based on normal load. OptimumT will not be discussed further as tire analysis is not in the scope of the thesis.

3.3 ADAMS/Car

ADAMS is a multi-body dynamics simulation software developed to solve large displacement systems. It was developed in the mid 1970s as the ADAMS/Solver. This was a text based application to solve non-linear numerical equations. The models were all written in text format which then was submitted into the ADAMS/Solver. In the early 1990s a graphical user interface was created called ADAMS/View. This lets the user build, simulate and review results in the same environment. Today ADAMS has different specialized packages for different applications, like ADAMS/Machinery, ADAMS/Flex, ADAMS/Rail and ADAMS/Engine.

In this thesis ADAMS/Car will be used, it is a specialized version of the ADAMS/View software focusing on vehicle design and simulation. It allows the user to design vehicle prototypes and run tests on them on the computer. This reduces the cost of creating real life prototypes and testing them [11]. This advancement has made it possible to improve the handling and safety of vehicles enormously for the last 30 years. As standard the software includes different base templates based on the most common suspension and chassis setups. It also includes simulation setups based on real life test cases, like line change on highway, J-turns, constant radius cornering and more. But if more custom suspension, sub systems or simulations are to be used, this is also possible to develop.

ADAMS Car's model hierarchy is compromised of three components, Templates, Subsystems and Assemblies. Templates are parameterized models of a systems where parts and joints are defined. This can also include force members like springs or bushings and dampers. To be able to modify or create templates the user have to have expert user privileges, this is discussed more in depth further down. Subsystems are based on templates and are available for modifications by standard users. Here the geometric position and properties of parts are defined. For suspension rods, this can be its hardpoints, mass and moment of inertia. Hardpoints are the points which define the geometry or position of a part. For a engine there might be less hardpoints to modify, but more parameters like engine torque or RPM limits.

Assemblies combine subsystems into a system. This can for example be a front or rear suspension system or a full-vehicle assembly. As mentioned ADAMS/Car have an

Expert user mode. This mode allows the user to enter the Template Builder to create new or modify old templates. The standard user only have access to the Standard interface which let the user only change subsystems and assemblies and run simulations. This division is done as templates are complex and require good knowledge of the software to create. Because of the way the model hierarchy is built, it is easy to change subsystems and compare different systems. For example to look at two different suspension types, the suspension subsystem can be easily changed with a different one, and simulations can be run at ones.

3.4 Physical Testing

To verify the kinematic simulations, measurements was done the the finished car. Both Toe and Camber was measured in heave, roll and steering. A setup with two beams mounted to the wheels, parallel to the ground, and tape measures was used to measure toe. The setup is shown in Figure 3.1. A digital level gauge was used on the wheel to measure the camber angle. By measuring the distance between a point 1 m behind the wheel on each beam, and a point 1 m in front of the wheel on each beam, it was possible to calculate the angle using Equation 3.1.

$$ToeAngle = \arcsin \frac{\frac{MDR-MDF}{2}}{2000} \tag{3.1}$$

where, MDR and MDF is measured distance rear and front.

In roll, the distance from the point 1 m behind the wheel on each beam and in to the chassis was measured. By comparing the results with the reference distance set at 0 deg roll, it was possible to calculate the toe angle on each wheel. Camber was measured in the same way as in heave, with the digital level on the wheels. The equation used for calculating Toe in roll is shown in Equation 3.2.

$$ToeAngle = \arcsin \frac{RL - ML}{1000} \tag{3.2}$$

where RL is the reference length measured at 0 deg roll, and ML is the measured length at any given roll angle.

The measurements on the steering angle and camber was done in the same way as with toe and camber in heave and roll, but only on the front side of the wheels. At angles over 5 deg, the beams mounted to the wheels would hit the chassis if they reached 1 m behind the wheel. Instead the setup shown in Figure 3.2 was used. Here the beams reach 2 m in front of the wheel and measurements are done on the third beam parallel to the two mounted to the wheels. By measuring the distance between the tip of the beams mounted to the wheels and the parallel beam at zero steer angle, a reference length is set. For each steering wheel angle, the same distance is measured and the change in length can be used to calculate the Toe or wheel steer angle using Equation 3.2. The only change is to divide by 2000 instead of 1000 as the length of the beams are 2 m. The Camber angle was done as on the Heave and Roll motion, with a digital level gauge at the wheel center.

To measure the roll stiffness, a rig with a vertical pole was mounted inside the cockpit of the car at the approximate center of mass. By applying a load on the top of the pole, and using a level on the chassis to measure the angle, the roll stiffness can be measured. The pole has a ring 1900 mm above the ground which is used to fasten a ratcheting strap which is connected to a weight secured in a rigid object. By tightening the ratcheting strap the car will roll and the force needed can be read on the weight. By subtracting the height of the roll center from the height where the ring on the pole is located, the moment arm around the roll center can be calculated. By comparing the force needed to roll the car at different degrees, the total roll stiffness of the car can be determined. The whole calculation is shown in Equation 3.3.

$$RollStiffness = \frac{\frac{W*g*(H-RCH)}{1000}}{deg} \tag{3.3}$$

where W is the mass measured at the weight, g is the gravitational constant, H is the height of the ring on the pole, RCH is the roll center height and deg is the angle of roll the car is experiencing at the given load.

If the front or rear roll stiffness was to be determined, a more advanced rig need to be built. It would need some way of letting one end of the car roll free, while the other is fixed, and the loads should be applied over each axle. As in depth testing of the suspension system is not part of this thesis, only the total roll stiffness have been measured.

Figure 3.1: The measurement rig used to measure toe in roll and heave. The steel beams are mounted to the wheels with a rod through the hub, and supported at wheel center and parallel to the floor by the wooden blocks at each end.

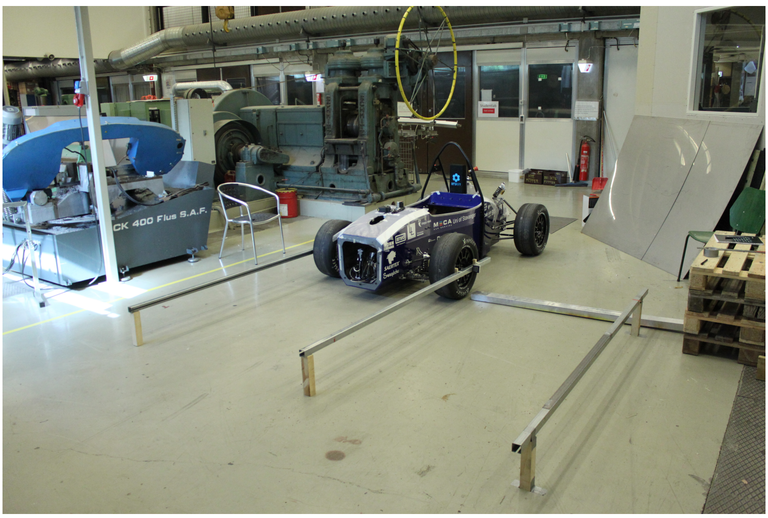


Figure 3.2: The measurement rig used to measure the toe angle in the steering case. As with the heave and roll, two beams are mounted on the wheels, while the last one is a reference beam.

Chapter 4

Simulation

The main purpose of this thesis is the simulation and evaluation of the results from different simulation software. Depending on how the software defines the problem mathematically and what equations it uses it might give different results on the same setup. The three different simulations software used are MATLAB/Simulink, the OptimumG simulation package and ADAMS/Car. The simulations that will be run here are:

- Suspension kinematics
- Body roll in cornering
- Acceleration event

Suspension kinematics look at the movement in the suspension without the dynamic effects of damping or springs. Usually the kinematics are evaluated based on a set steering wheel angle or body motion. This can either be a displacement or an angle set to the body or the wheel. In OptimumK the motion is applied on the body and/or steering wheel, while in ADAMS/Car the motions is applied to the wheel and/or the steering wheel. The different applications of motion gives the same resulting suspension motion, as it depends on the reference frame of the observer. The simulations will study the camber change and toe/steering angle in different motions to compare the results from the two software. Suspension kinematics will not be simulated in MATLAB/Simulink as it would involve a lot of trigonometric equations which will be too time consuming to derive. Therefor this simulation will only be compared between OptimumK and ADAMS/Car.

Body roll in cornering will look at the suspension roll stiffness and compare the results with a calculated value from MATLAB based on a procedure by David Gould from the book Competition car suspension by Allan Staniforth [12]. Acceleration event will be based on the 75m acceleration in the Formula Student competition. The car will start at 0 km/h and accelerate down a 75m straight. The time needed to cover this distance will be compared for the different setups. Here a few different MATLAB scripts will be used to look at the difference in results depending on what inputs are used.

For most commercial software, the equations used in the solver are a mystery to the user. What is going on in the solver is hidden and is known only to the software developer. This makes it difficult to assess where differences in results originate. It might be because of different inputs into the simulation, or a different definition of directions, or there could be a calculation in the solver that is generating different results than another software. Figure 4.1 illustrate how most commercial software operate, with the user defining inputs and let the software handle the calculations inside its black box, and generating outputs which the user can read and analyze.

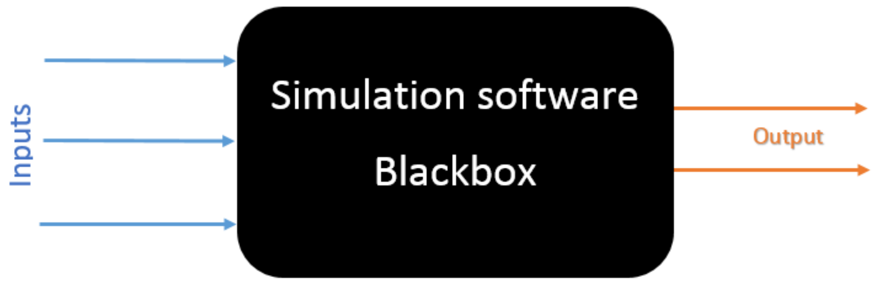


Figure 4.1: Illustration of how most commercial software handles simulations, with inputs from the user and outputs to the user.

4.1 MATLAB

MATLAB is a versatile software capable of running advanced simulations. Here MATLAB will be used for simple simulations to get approximations of the results based on general formulations. Different formulations will be used for the different simulations cases, and for some of the cases multiple formulations with increasing complexity will be used. One of the benefits with MATLAB simulations is that the user is in control of the whole process and is able to look at what is going on in the simulation. But it also require a lot of time to develop or understand the simulation and derive all the formulations used. This can be a time consuming process and requires knowledge of the mathematics behind the case being studied. To help with developing simulations or any calculation or data treatment in MATLAB, there is a huge community available to help. This is one of the benefits with MATLAB. Someone has most likely done something similar before and often they have published their code for open use. This code can then be modified or combined with other codes to generate the required script for the simulation. MATLAB will be used for body roll siffness calculation and to simulate the acceleration event. Both the body roll stiffness code and the acceleration event code will be available in the Appendix.

4.1.1 Body Roll Stiffness

The roll stiffness is the suspensions ability to resist rolling when turning a corner. Race cars are designed to prevent as much rolling as possible, to keep the weight distributed evenly on all wheels. Roll stiffness is defined as the moment needed to roll the chassis 1 deg. This is done around the roll center of the car by the mass center of the vehicle. By positioning the mass center close to the roll center, the roll stiffness is increased, as the moment arm between the mass and roll center is reduced. If the mass center is placed at the same height as the roll center, the car will not roll. As the mass center of most cars are not in the same position as the roll center, rolling will occur. To prevent this, stiffer springs and ARB systems can be implemented. To be able to determine the roll stiffness of a car, the force causing the roll, and the forces acting against the roll have to be determined.

In MATLAB the body roll stiffness is calculated based on calculations described by David Gould in the book Competition Car Suspension [12]. This have been used in the development period of Bifrost to determine how stiff the ARB system need to be. It have also been used to decide how stiff the chassis of the car need to be. It is a common practice to have the stiffness of the chassis 3-5 times stiffer than the total roll stiffness of the suspension. The MATLAB code will use the same functions as presented by David Gould with the values representing Bifrost.

4.1.2 Acceleration

To simulate the acceleration time for the event, a few different simulations have been done in MATLAB. The first calculation is based on the kinematic equation for distance (Equation 4.1). By solving for time, and using Newtons second law to replace the acceleration with $\frac{F}{m}$, Equation 4.2 is derived. *S* is the distance of the acceleration, *m* is mass of the car and *F* is the tractive force in the contact patch of the wheel. To define the tractive force equation 4.3 is used, where m_r is the rear weight of the car, g is the gravitational acceleration and μ is the coefficient of friction of the tire.

$$S = v_i * t + \frac{1}{2} * a * t^2 \tag{4.1}$$

$$t = \sqrt{\frac{2 * S * m}{F}} \tag{4.2}$$

$$F = m_r * g * \mu \tag{4.3}$$

This calculation assumes that the car accelerate at a constant value through the whole run. As both the tractive force of the tire and the motor output will change over the duration of the run, the acceleration of the car will change. The acceleration event can be broken down in to 3 sections. The first section is the tire tractive limiting section, where the tractive force in the tire limits the acceleration. The second section is the motor limiting section, where the motor torque limits the acceleration. The last sections is the top speed section, where the motor has reached its max rpm and the car can not accelerate any more. To take this into account a second calculation were set up. By incorporating a loop to calculate distance traveled per time, the velocity and acceleration, a better approximation can be made. The script bases its engine torque map on the same used in OptimumL and ADAMS/Car, and linearly interpolate the torque between the points based on RPM. By running a loop which calculates incremental changes in the acceleration and speed based on a fixed time step and the previous speed and acceleration, it is possible to calculate the time needed to transverse 75m. Two scrips have been written with this same basic principle. One assumes a constant weight on the rear axle resulting in a constant tire friction, while the other calculates a load transfer based on the acceleration and calculates a new tire friction force each step. The last script is based on a calculation done in reference [1] p.267, which include rolling and air resistance of the vehicle. As the two previous scripts do not take into account resistances, they will most likely produce a faster acceleration time, and might not be that accurate.

4.2 OptimumG

The different software in the OptimumG package is used for different simulations based on what they are suited for. OptimumL will be used for simple lap time simulation and OptimumK for kinematic analysis of the car. The car has been designed in OptimumK, which makes it easy to setup and run simulations as the suspension layout is the most time consuming part of the simulation setup.

4.2.1 Kinematic Analysis

For the kinematic analysis of the suspension OptimumK is used. The suspension layout used in the design process of the car is imported and new motion cases are created. The suspension can be shown in Figure 4.2 and list of coordinates for each point in the suspension system is available in Appendix E. The motion cases are created by defining a start position and a end position in a plot, with linear motion in between the points. Figure 4.3 shows the layout of the motion creation window in OptimumK. By adding and changing the values in the table on the left advanced motions can be created simulating real body behaviour on track. As mentioned it is also possible to add logged motions into the simulations. In the case of the heave, roll and steering simulations done for this report, only point to point linear motions are needed.

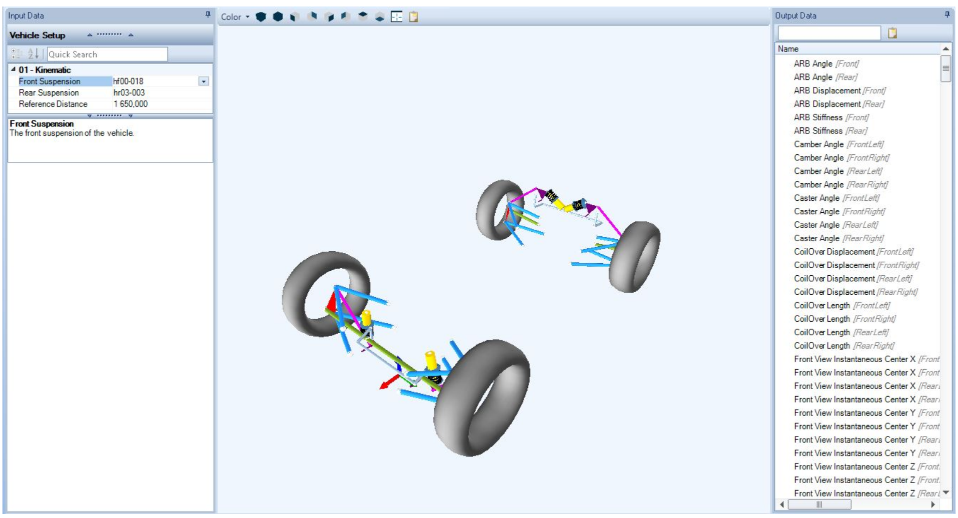


Figure 4.2: The full vehicle suspension used in OptimumK simulations.

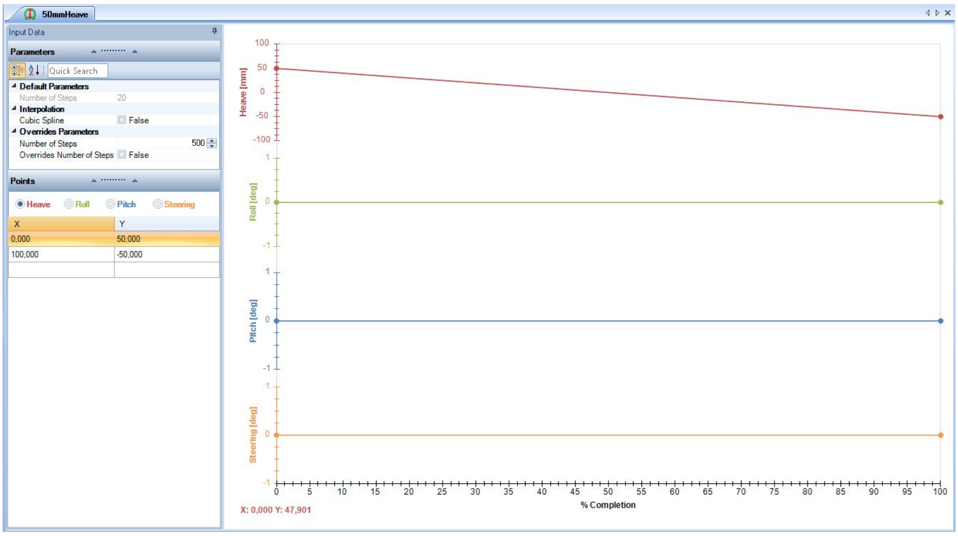


Figure 4.3: The heave motion definition in OptimumK. Roll and steering motion have the same layout and can also be combined to create extreme cases like full roll at max steering input in a bump.

4.2.2 Lap event analysis

For the lap time event for acceleration, OptimumL is used. This software assumes the car as a single point with 10 parameters to define its performance. These are Weight, Drivetype, Aerodynamic properties, Tire data, Engine Data and Transmission data. The Aerodynamic data is mostly left untouched as this is not available for the 2018 car, but the values for a non-aero FSAE car template is used except for the frontal area which is measured in CAD software. The tire properties are gathered from tire data in OptimumT. The engine torque map is calculated based on the electric motors data sheet and allowable power consumption in the competition. As the car uses a fixed gear ratio, a single gear with 1 in gear ratio is added in the transmission and the final drive ratio is set as the transmission gear ratio. Figure 4.5 show the values used when configuring the vehicle. The weight of the car is defined with a driver weighting 60 kg.

With the vehicle done, a track has to be made. This is possible in OptimumL track configurator. Here straights and corners can be created and are set in sequence to generate a track. For the acceleration event, a single straight is created with a length of 75m, shown in Figure 4.4

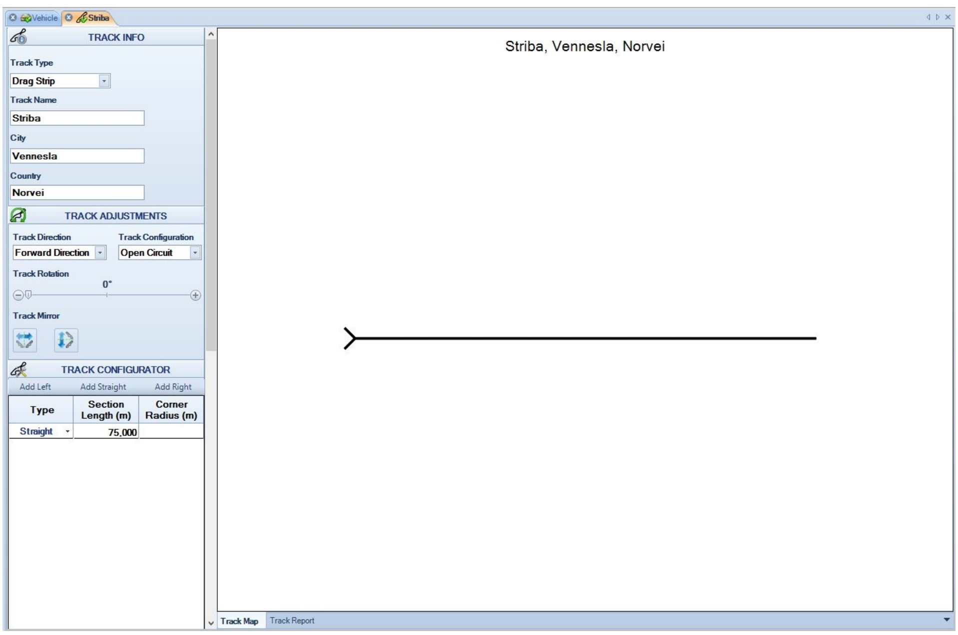


Figure 4.4: The acceleration track defined in the track configurator in OptimumL

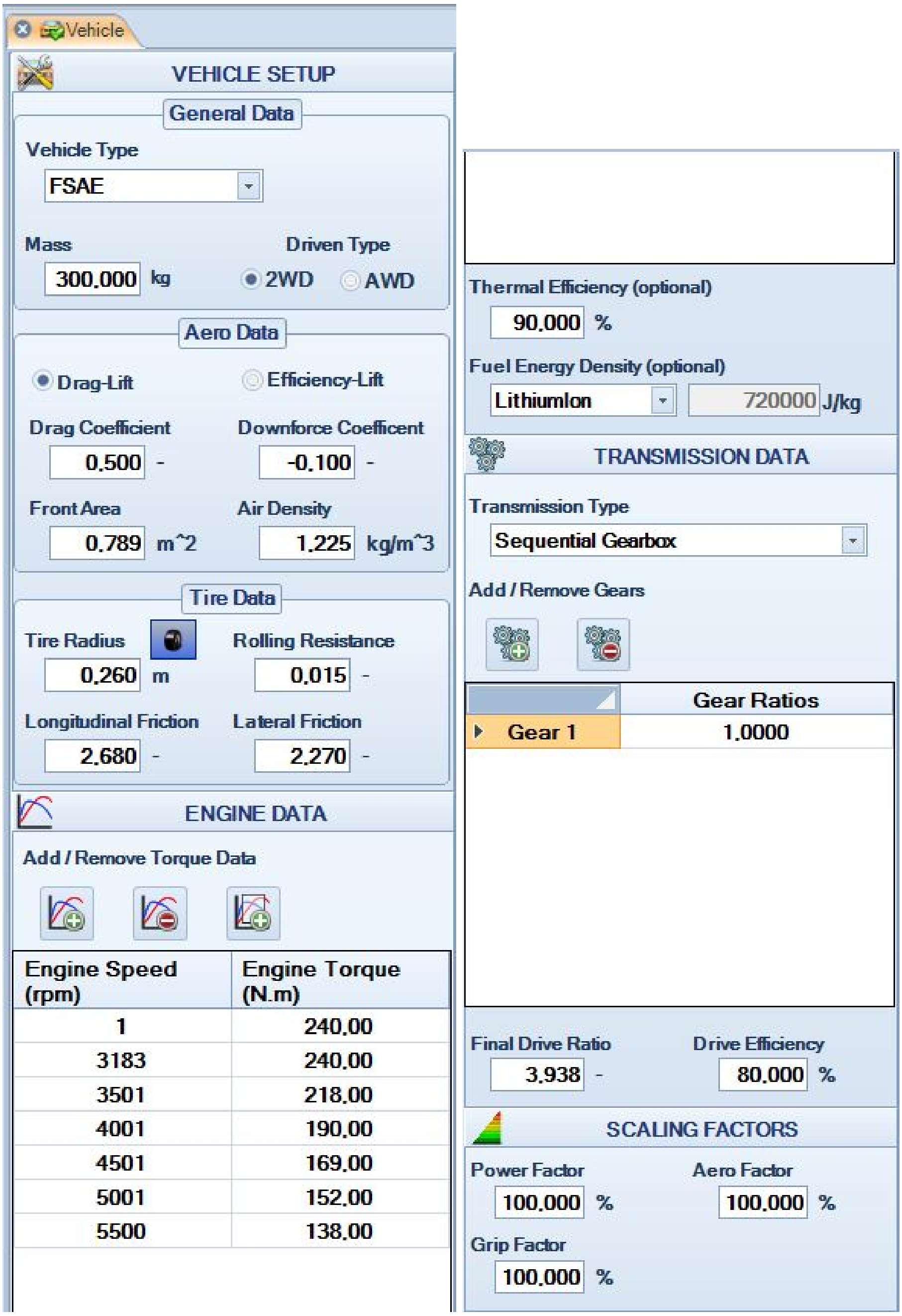


Figure 4.5: The vehicle data for Bifrost represented in OptimumL.

4.3 ADAMS Car

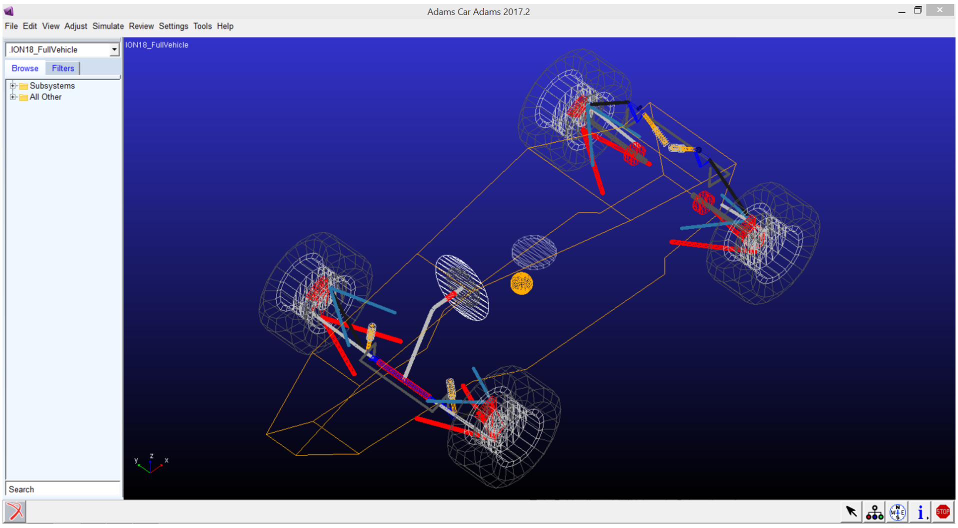


Figure 4.6: An overview of the main view of the full vehicle assembly in ADAMS/Car.

To create the model of the vehicle in ADAMS/Car some modifications have been done to the FSAE 2012 template available from MSC's website [13]. The main changes to the template, is the change from pushrod to pullrod in the front suspension and the modification of the powertrain to fit an electric vehicle. By modifying the coordinates of the hardpoints, the geometry of the suspension and pick up points can be defined. When all geometry has been defined, the properties of all the subsystems has to be applied. Most systems have properties file, which can be modified with regular text editors, like Notepad. To get a good representation of Bifrost, a new engine file was generated, describing the EMRAX228 motor, and changes to the Powertrain subsystem was done to represent the single speed gearbox used in the car. Springs and damper properties was also changed to represent the springs and the hlins TTX25 dampers used on the car.

To get a rough estimate of the outline of the chassis of the car, the frame nodes in the FSAE 2012 Chassis template was modified to match the contour of the monocoque used on ION2018. The result of all the modifications is shown in Figure 4.6. The whole process is described in more detail in Appendix A. This full vehicle model is used to do the dynamic simulations of the car, like acceleration.

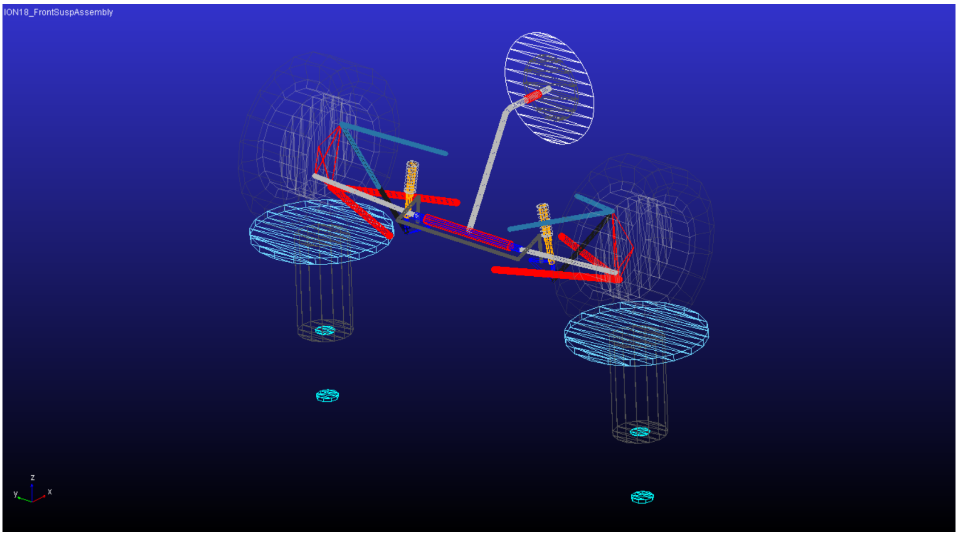


Figure 4.7: The front suspension assembly with ARB and steering. The wheels are supported on a motion test rig, which applies motion to the wheels when running simulations.

For the kinematic simulation to verify the motion of the suspension, a separate assembly have been created for each end of the car. These assemblies only contain the suspension geometry, ARB and steering for the front, and drive axles for the rear. Figure 4.7 show the front suspension assembly with the suspension test rig. This front assembly contain the suspension subsystem, the ARB subsystem and the steering subsystem. The tires are a part of the test rig and are defined by test rig parameters, like radius and width. To define the different parameter for the test rig, the Setup Parameter has to be defined. These can be found under Simulation → Suspension Analysis → Set Suspension Parameters. The parameters used in the simulation can be seen in Figure 4.8.

The tire unloaded radius is found from the tire test data, or measured on the tire directly. The tire stiffness can also be found in the tire test data, as one of the warm up sequences. Here the tire testing machine pushes the tire down in the road and measure the force needed to deform the tire a specified amount. By comparing the unloaded radius with the loaded radius from the test, and the force applied at the defined loaded radius, a tire stiffness can be determined. Mass of the wheel is the mass of the tire and wheel together. Sprung mass is the mass of the car which is supported by the suspension system. For Bifrost this is approximately 190 kg. CG height is the height of the mass center in the car. This have been determined by an Excel spread sheet where all part in the car has been given a position and mass. These positions have been determined in CAD software and the masses is a mix of CAD values and weighted parts.

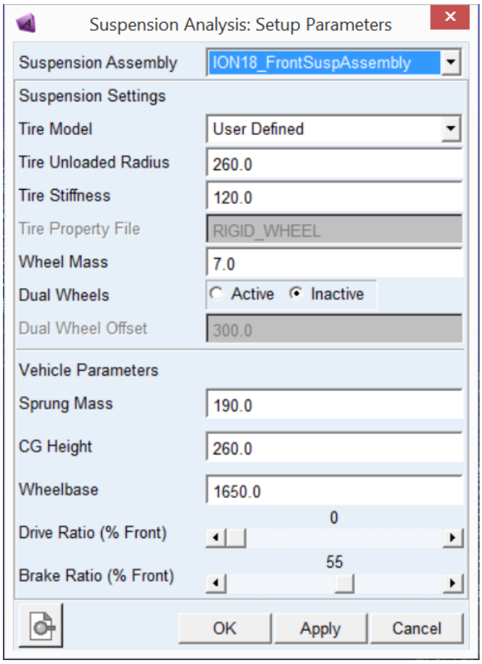


Figure 4.8: Setup parameters used for the kinematic analysis on the front suspension in ADAMS/Car

Wheelbase is set to 1650 mm as one of the design values to work with from the beginning of the concept phase. This is a distance we have used for most of our cars and is a compromise between maneuverability and packaging. The drive ratio is the amount of tractive force going to the front wheels. As this is a rear wheel drive car, this value is set to 0. The Brake ratio is the brake force difference front and rear. The calculations done on the brake system have estimated the front brakes to have 55% of the brake force.

ARB systems can be modeled in a few different ways. It can be modeled as a torsional spring with a fixed stiffness value, or it can be modeled as a flexible model with varying thicknesses and stiffness. In this simulation the ARB is modeled as a torsional spring with a fixed value.

When running simulations, the motion is applied to the test rig and transferred to the wheels through contact joints. The hard points connecting the suspension to the chassis are fixed. As OptimumG applies the motion to the chassis and lock the contact point of the wheels the results need some post processing in order to be compared directly. This post process have been done in Excel.

Figure 4.9 show the setup for the heave simulation. Jounce and Rebound is upward and downward motion of the wheel.

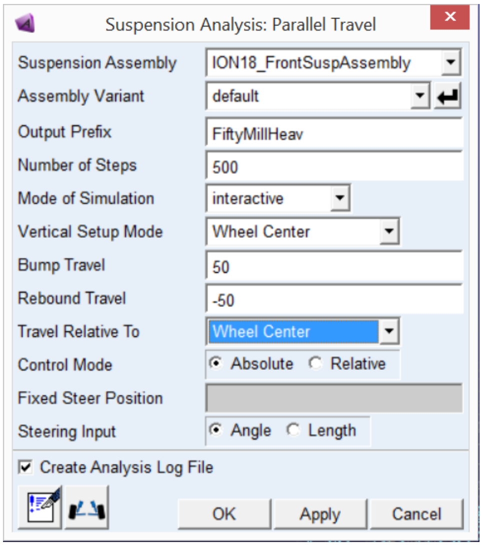


Figure 4.9: The setup parameters for +-50 mm heave simulation

Figure 4.10 show the simulation setup for the roll simulation. This simulation also need a vertical force. This might affect the results compared to OptimumK. The chosen vertical force is based on the normal load on the front axle to reduce the effect of this variable.

Figure 4.11 show the setup for the steering simulation. +-90 deg is applied to the steering wheel and through the gearing in the steering column, turns the wheels. For all the simulations, 500 steps are chosen to get more steps in the animation of the simulation. This animation was used to check for motions that might be wrong. When doing steering simulations, this post simulation check with the animation helped discover faults in the steering subsystem. The column was set to invert the steering motion resulting in steering right with the steering wheel rotated the wheels to the left. Another fault discovered was the steering ratio, where 1 deg of steering wheel motion resulted in 22 deg at the wheels. This gave an unstable car when running acceleration simulations. By addressing these faults the acceleration simulation was more stable and did not result in the car running off track.

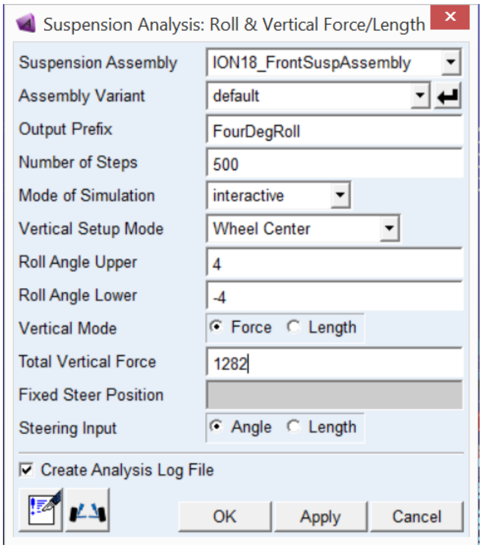


Figure 4.10: The setup parameters for +-4 deg roll simulation

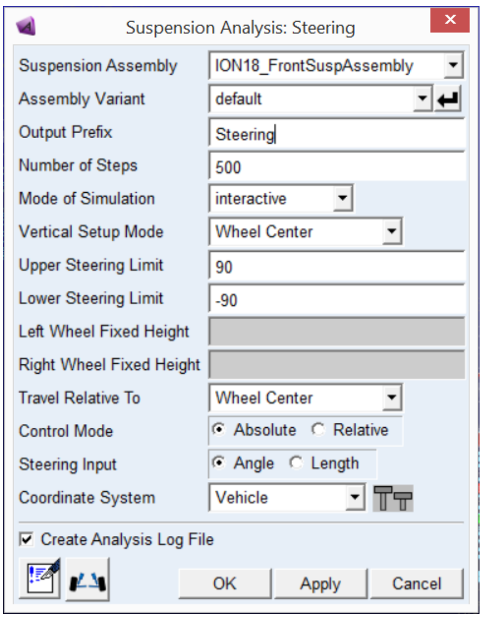


Figure 4.11: The setup parameters for +-90 deg steering wheel rotation simulation

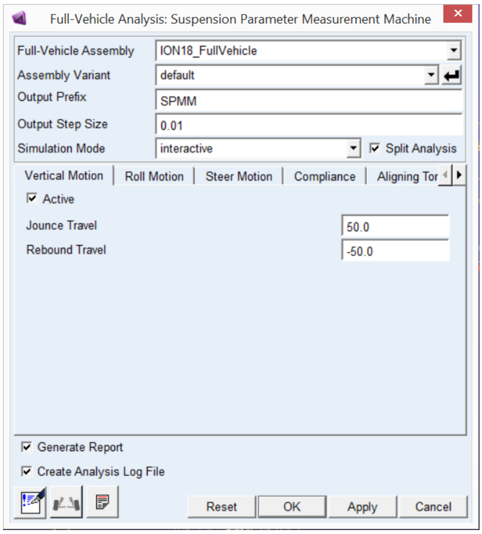


Figure 4.12: The setup parameters for the SPMM simulation. Most values was left as default, only the steering angle was changed.

The last simulation is the vehicle parameters, focusing on the roll stiffness. This is done using the Suspension Parameter Measurement Machine (SPMM). This is a simulation in ADAMS/Car where the vehicle is put on a virtual test machine. The simulation is located under Simulation→Full-Vehicle Analysis→Kinematics and Compliance→Suspension Parameter Measurement Machine. This simulation moves the suspension through predefined movements and applies forces to determine compliance in the parts. The compliance in our case is not accurate, as the suspension members and parts are not accurately modeled. Figure 4.12 show the setup window for the SPMM simulation. The first tab show the vertical motion, under roll and steering motion the values are set to 4 deg roll and +-90 deg steering. When the simulation runs, it puts the vehicle through various motions, both kinematic and dynamic, which determines the suspension parameters of the vehicle. Here the total weight and distribution, inertia, Spring rates, roll stiffness, wheel rates and a lot more can be determined for the suspension.

Chapter 5

Results

This chapter will present the results from the simulations conducted using the simulation tools discussed in Chapter 3. Comparisons between the tools used and notable points of interest will be discussed for front suspension kinematics, roll stiffness and the acceleration event simulations.

5.1 Front Suspension Kinematics

To study the results of a kinematic simulation between ADAMS/Car and OptimumK, the front suspension has been analyzed. The motion cases considered in the simulation are the following:

- Heave motion
- Roll motion
- Steering wheel motion

As mentioned earlier in the report, there are some differences in the reference frame of the two software. As heave in OptimumK is defined as vertical chassis motion and in ADAMS/Car it is defined as vertical wheel motion, negative heave in OptimumK is positive heave in ADAMS/Car. To make the result plots more clear, they have been combined into one plot for each case and the signs have been edited to account for differences in reference frame.

5.1.1 Heave

The heave motion for the front suspension is run from -50 to +50 mm, as this is the full range of motion the front suspension is capable of. The results from the analysis which will be discussed here is:

- Toe vs Heave
- Camber, Caster and Kingpin vs Heave
- Roll Center height vs Heave

These three plots show some of the effects heave motion has on the chassis. The toe and camber change will in addition be compared to measurements done on the car.

The plot in Figure 5.1 shows what is called bump steer, which is how much the wheels turn when going over a bump. To much of this will make the car unpredictable when running over rough roads. Positive heave is defined as lifting the wheels or moving the chassis downwards. Positive toe angle is defined as pointing the tire inward against the chassis. We can see from the plot that both OptimumK and ADAMS/Car have the same amount of toe change on both wheels, but the two software have a small difference in the amount of toe. The results from ADAMS/Car seem to be over all about 0.012 deg higher than in OptimumK. The blue dots are the measured results on the actual car. The results are spread out, but roughly follow the line from the simulations. As the toe angles are so small, and the measurement rig is not completely stiff and stable, these variations in the results can be expected.

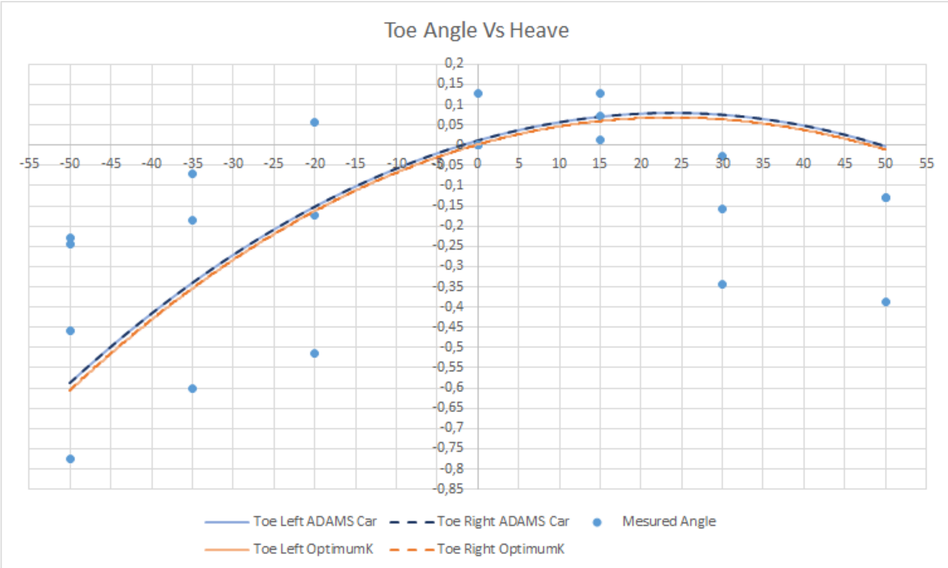


Figure 5.1: Plot of toe vs heave motion from both ADAMS/Car and OptimumK

The plot in Figure 5.2 shows the change in Camber, Caster and Kingpin angle for the two software. Here both share the same values. This is most likely because these three angles are defined directly by the position of the pickup point on the upright, and as the hardpoints in ADAMS/Car is based on the same hardpoints in OptimumK, these will be the same. What is most interesting of these three results is the change in camber. As camber changes with wheel rise or fall, the size of the contact patch of the wheel change. In an breaking event, the front of the car will dive. This is essentially the same as moving the chassis downward, or in our case, the wheels upwards. A motion like this will generate negative camber on the front wheels resulting in reduced grip and weaker breaking force. To prevent this dive, stiffer springs can be added, an anti-pitch system can be incorporated or the suspension geometry have to be changed.

On Bifrost the camber change in dive seems to be significant, based on the results. When the suspension was designed, this was one of the considerations taken. If the tires was going to be kept close to upright in corners, they will gain excessive camber in acceleration and breaking. It should be taken into consideration to stiffen up the front springs to prevent the nose of the car to dive in breaking events. The blue dots and triangles show the measured camber angles for both the left and right wheel. The deviation from the simulated results are not to high.

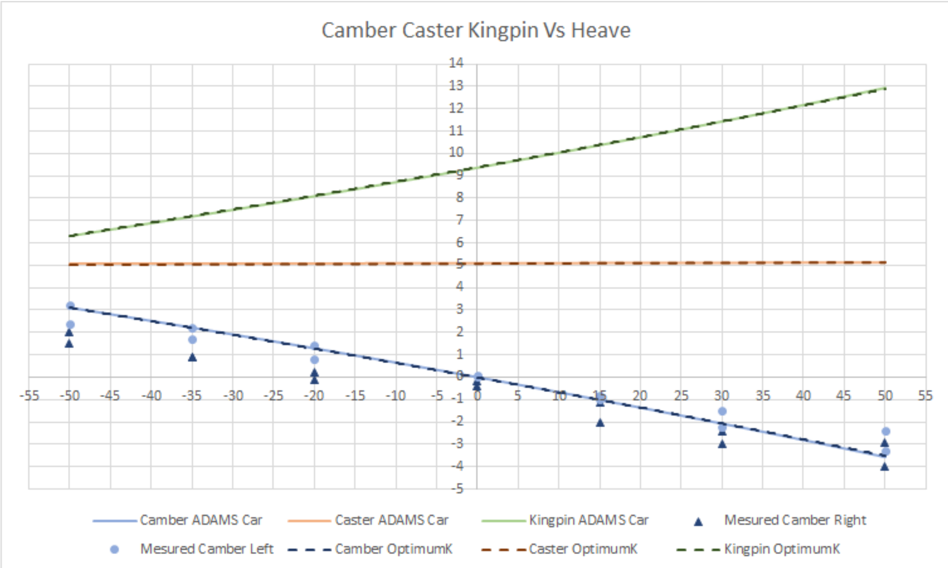


Figure 5.2: Camber, Caster and Kingpin angle plotted against Heave motion

Figure 5.3 show the vertical movement of the roll center in heave. In the neutral position with zero heave OptimumK sets the roll center at about 87mm above ground, while ADAMS/Car sets it at 85mm. At 50mm chassis heave both OptimumK and ADAMS/Car puts the roll center at about 155mm, but in -50mm chassis heave ADAMS/Car put the roll center at about 17mm while OptimumK put it at 24mm. There are a difference in how the software calculates the position of the roll center. In ADAMS/Car it is done by force vectors, projected from the contact patches of the tires [11]. OptimumK the method is unknown, but most likely it is based on node to node vector calculation, which is the most common method. Because of the way ADAMS/Car calculates the roll center, it is important to have the right tire model to get an accurate result from the simulation. As the tire model will define the displacement of the contact patches in the tires. In addition the contact patch will move depending on the camber of the wheels, resulting in even more roll center movement.

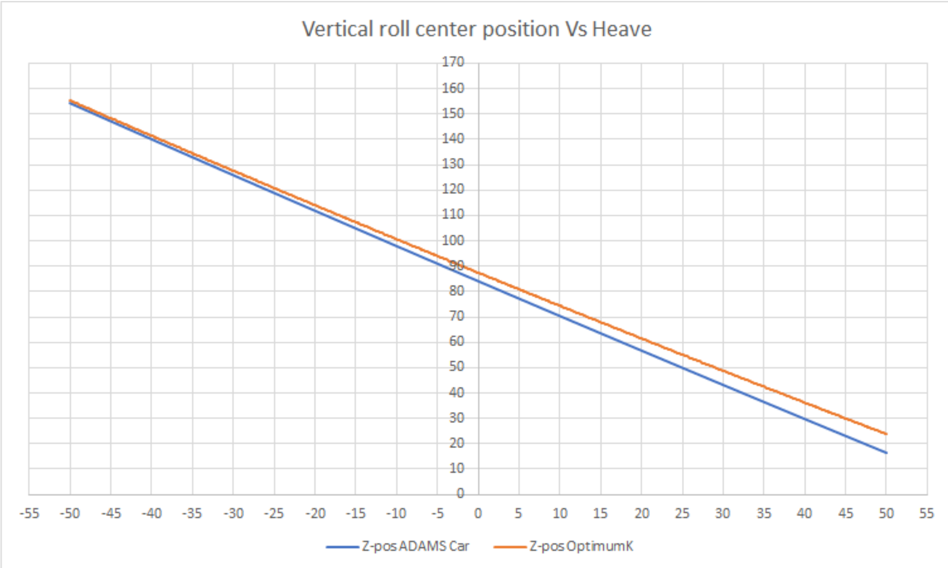


Figure 5.3: Roll center height in Heave motion

5.1.2 Roll

When running the roll simulations a roll from -4 to 4 deg have been used in both software. Positive roll is defined as rolling over to the right side. The roll simulation in ADAMS/Car include a vertical load, which can have affected the results in a minor way. Plots of the roll center movement and toe change are presented. The first plot in Figure 5.4 show the vertical movement of the roll center in body roll. Here we can see the same as in the heave case, that the static position of the roll center in the two software is different with ADAMS/Car at 84mm and OptimumK at 87,5mm above ground. There is also a significant difference in amount of vertical movement in the roll center. In OptimumK the roll center is moving about 1mm lower with 4deg roll, while in ADAMS/Car the roll center moves about 4,5mm with 4deg roll. Comparing the roll center position in roll at 4 deg with the roll center position in heave, we can see that it compares to about 2.5 mm wheel movement. The values for the roll center height is equal for both of the simulations. The difference from the tools should be examined closer as it can affect the handling of the car if the difference was larger. But as the roll center is an imaginary point in space it is difficult to measure directly on the car, it is therefor not easy to validate this analysis. Here both roll centers are relatively close to each other and do not differ more than 10mm, which is not a affecting the performance much. But if the difference in roll center movement is larger for another setup, there might be a reason to look into which of these simulations produce the most accurate and real results.

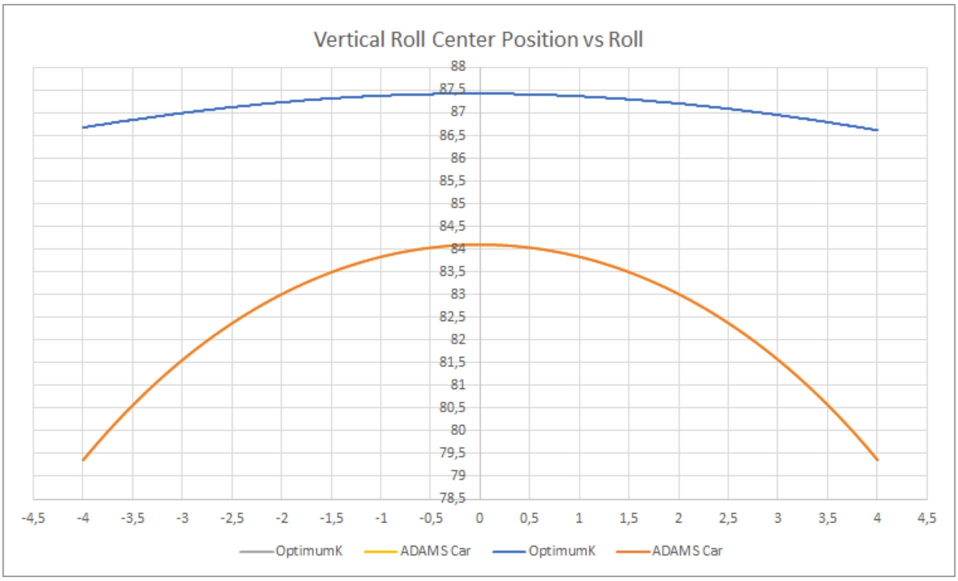


Figure 5.4: Roll center movement in Z axis in chassis roll

Figure 5.5 show a plot where the toe angle of the front wheels are plotted against the roll angle. Positive toe is, as defined above, pointing the tire towards the vehicle. Here the curve have a similar profile, but the results from ADAMS/Car have a more flat curve on the negative Toe side. When comparing the simulation results from the heave and roll simulation, -4 deg roll is about the same as -41.6 deg heave for the left side wheel. For OptimumK the results are close for both simulations, but ADAMS/Car have a lower value on the negative toe angle in roll then in heave. It might have something to do with ADAMS/Car's definition of toe as the the angle between the X-axis of the car with an line defined by the intersection of the ground plane and the center plane of the wheel. As the wheel center plane is also affected by camber, any change in camber will move the intersection line [11]. In OptimumK toe is defined as the angle between the wheel center line and the vehicles center line [9]. This discovery led to more investigation into the camber change in roll and the effect this had on the toe for both ADAMS/Car and OptimumK.

The measured angles do not help much when determining which analysis have the most real results, as the measured values are to varied for any accurate definition. But it shows how little the difference in the two simulations mean compared to measurable values.

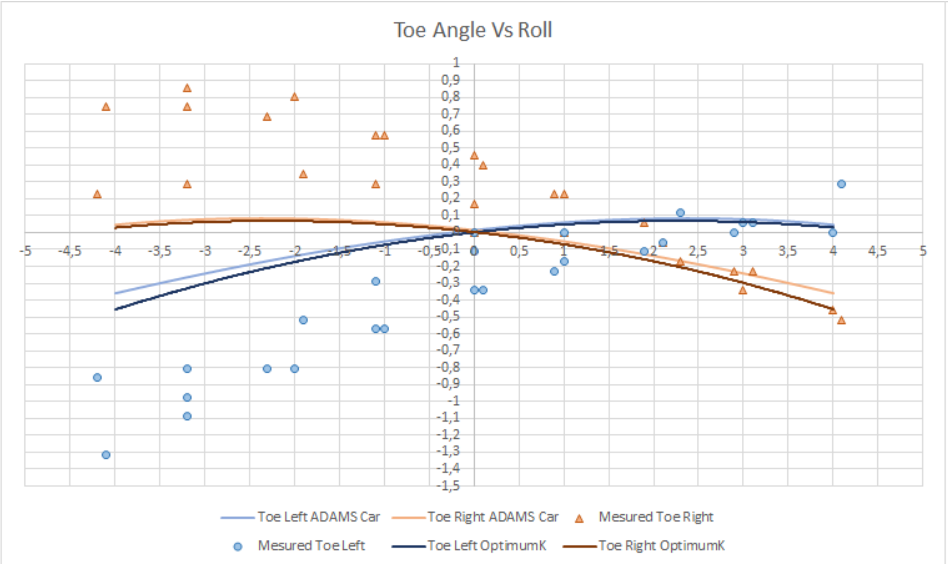


Figure 5.5: Wheel Toe angle change in roll

Camber in ADAMS/Car is defined as the angle between the vehicle's vertical plane and the wheel plane. In OptimumK, camber is defined as the angle between a plane perpendicular to the ground and the wheel vertical axis. This is the same as the inclination angle in ADAMS/Car, which is equal to the camber angle subtracted from the roll angle. To make an easier comparison, inclination angle is used instead of the camber in ADAMS/Car. Figure 5.6 show the difference of Camber in roll. When comparing the toe and the camber plot, negative camber seem to generate positive toe in ADAMS/Car. ADAMS/Car's definition of toe seems therefor to be affected by the camber of the wheel, and therefor this is most likely the reason for the negative toe difference at +-4 deg roll between the two software. But the reason for the difference in the amount of camber is still not completely understood. From the measured data, the OptimumK simulation seem to be a bit closer than ADAMS/Car's results. This is mostly likely because the way camber was measured is the same as OptimumK uses to define camber. The variation is still large enough that it includes the simulated results from ADAMS/Car.

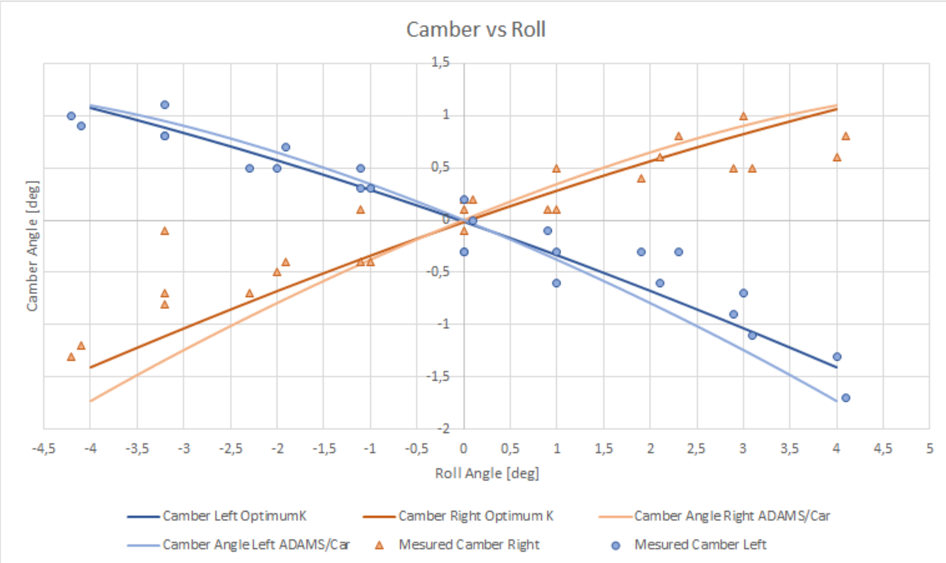


Figure 5.6: Camber change in chassis roll

5.1.3 Steering

In the steering simulation, the steering wheel is turned from -90 deg to 90 deg. Positive steering wheel angle is defined as turning to the left. Here the camber change and the toe angle will be compared. The camber will change depending on the amount of steering angle because of the kingpin and caster angle on the wheel. For most cases it is desirable to have a positive camber angle on the inner wheel and negative on the outer wheel in a turn. This increases the contact patch of the tire compared to an upright wheel in a corner. camber is not desirable in straight line maneuvers as it reduces the contact patch of the tire. This explanation is a simplification, but hold true for most cases.

Here toe describe how much the wheel turns when turning the steering wheel. It can be used to determine how narrow corners the car can maneuver through and how much Ackermann the car got. If a car got 100% Ackermann, the wheels follow perfect arches around the center of a turning circle when going around corners. For slow speeds with a stiff wheel this will be valid, but as the speeds increases slipping in the wheels will occur and the car will not follow the perfect Ackermann lines through a corner. Therefor it is often necessary to have an Ackermann higher than 100% to account for the slippage.

The plot in Figure 5.7 show how the camber change when turning the wheels. Camber is affected by the caster and kingpin angles and is usually desired to be close to 0 in straight lines and the top of the wheel angled towards the center of a corner radius when driving around a corner. This is why the curves are opposite each other for the left and right wheel, resulting in both wheels cambering inwards against the corner center. On the negative camber side of the curve, both OptimumK and ADAMS/Car have a close to identical change in camber. But on the positive camber side, they split and OptimumK gives 4 deg of camber at +-90 deg, while ADAMS/Car is 0,5 deg lower at 3,5 deg. This difference is actually significant and can affect the handling of the car.

By looking at the measured data points in the plot, the simulation and measured data do not vary to much on the positive camber side. The left side have some more difference than the right side. The right side stay close to the simulated results. But when looking at the negative camber side, both plots of the measured angles are not showing as much negative camber as the analytic results. The measurements do not go any lower than -0.6 deg, while the simulated results reach -1 deg. It is not known why this is the case, but it could be caused by some differences in the produced parts compared to the designed. There are some adjustability in the system, but all are tuned to their designed position at 0 steer angle. There might have been a change in the caster angle in the assembly process. This could have been done by mounting the lower wishbone upside down. This would cause a change in caster and cause the camber to be lower than the simulated values, but it would also affect the positive camber values, which is not the case here. There could also be some errors in the measurements. The digital level gauge could have been calibrated on an angled surface, but this would also affect the positive values. The most likely reason is that there are some differences in the designed system and what was produced. Some small adjustments can change the whole kinematics of the system. One turn on a rod-end or a loose nut can affect the end result.

The next plot in Figure 5.8 show the toe or wheel angle with respect to steering wheel angle. As mentioned, this plot can tell how sharp corner the vehicle can handle, and by looking at the difference of the left and right wheel, the amount of Ackermann the steering system have. As shown in the plot, ADAMS/Car shows a bit lower steering angles than OptimumK. This is most likely a result from the change in camber, as discussed in the roll section earlier. But comparing the simulated and measured results, the deviation in the simulated results do not seem to be to large. The overall measured angles stay close to the simulated results. As in the toe measurements in heave and roll, there are bending in the rig used to measure the angles which can cause differences. For these measurements this bending is not affecting the overall results as much as in heave and roll, as the values are greater and 0.3 deg difference on 20 deg is not as much as 0.3 deg on 0.3 deg.

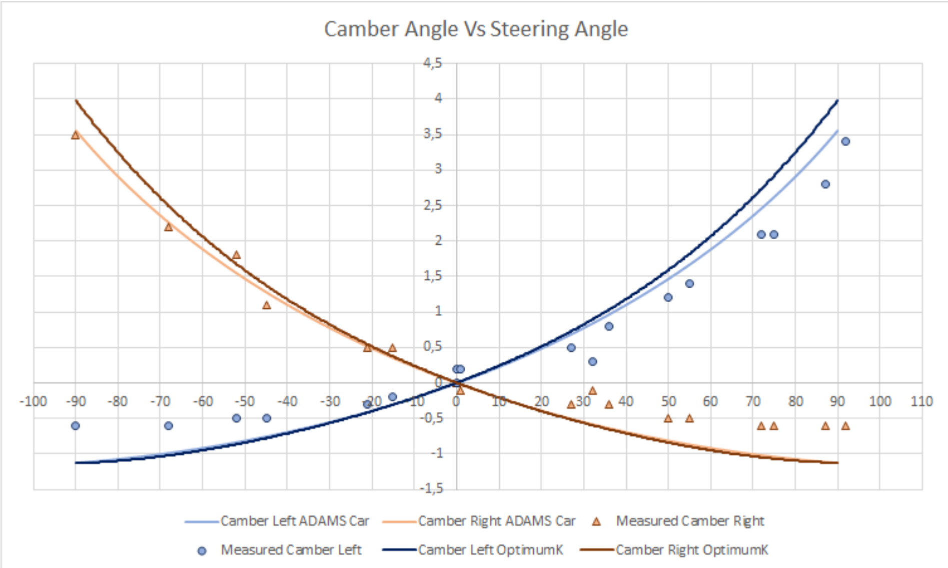


Figure 5.7: Camber change as a result of steering

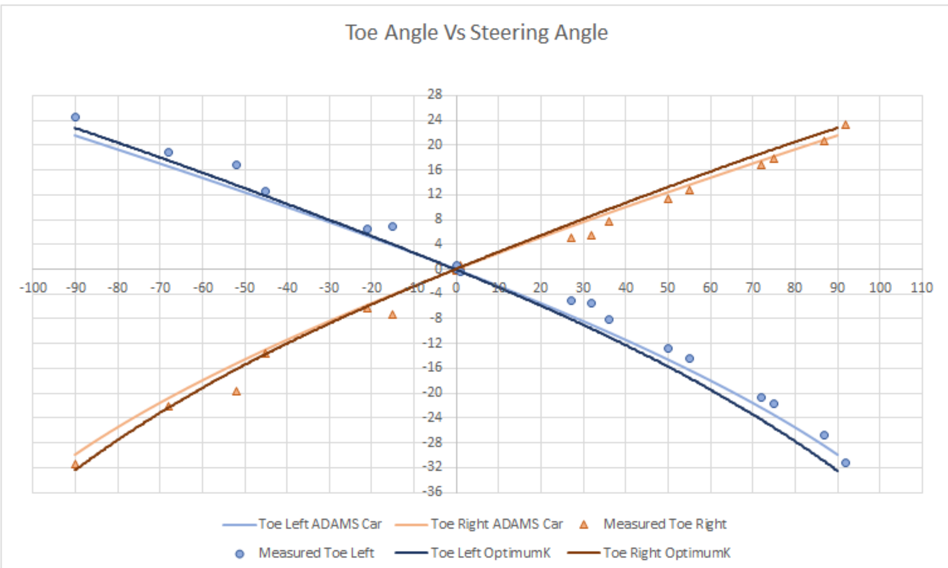


Figure 5.8: Wheel Angle vs Steering wheel angle

5.2 Roll Stiffness

The roll stiffness from ADAMS/Car is taken from the SPMM simulation described in Chapter 3.3. The simulation gave the following results:

- Roll Stiffness Front: 214 Nm/deg
- Roll Stiffness Rear: 345 Nm/deg
- Total Roll Stiffness: 559 Nm/deg

The MATLAB calculation gave the following results:

- Roll Stiffness Fromt: 212.3 Nm/deg
- Roll Stiffness Rear: 383.2 Nm/deg
- Total Roll Stiffness: 595.5 Nm/deg

From the results we can see that both gave about the same stiffness in the front, with a difference of 1.7 Nm/deg. The rear is a bit higher in the calculations, with 40.2 Nm/deg. A possible reason for this difference could be that the MATLAB calculation do not take into account the bending of the arms connecting the torsion rod to the rockers in the ARB system. ADAMS/Car do not take this directly into account either, but instead uses a torsional constant. This is derived from a table in Excel. This table calculates the stiffness of the torsion bar, and uses stiffness's of the arms connecting the bar to the rockers from FEM analysis. These are combined by the formula for springs in series to derive the final ARB stiffness used in ADAMS/Car. The front ARB system uses much shorter arms than the rear system, and therefor the difference is larger in the rear than the front.

Table 5.1: Data collected in the roll stiffness test. The first of the 1 deg roll cases were without the full mass in the vehicle. This led to wheel lift at more then 1 deg roll. The rest of the measurements was done with the total vehicle weight with driver.

Deg	kg	N	Nm	Nm/deg
1	29	284,49	516,6338	516,6338
1	32	313,92	570,0787	570,0787
2,1	58	568,98	1033,268	492,0322
3,2	82	804,42	1460,827	456,5084
4	90	882,9	1603,346	400,8366

The measurements on the car was done with the rig described in Chapter 3.4. By reading the results for roll at 1 deg increments from 1-4 deg roll, and calculating based on the length of the moment arm around the roll center, the total roll stiffness was derived. Table 5.1 show the calculated results based on the measurements.

It can be seen that with increased roll the roll stiffness is not a fixed value, it is reduced by 170 Nm/deg over 3 deg. The calculations done in MATLAB uses the initial value which is needed to start rolling from 0 deg. The ADAMS/Car manual do not specify at what roll angle the roll stiffness is based on, but from the calculations and measurements, it seems to be around 1 deg. It is therefor safe to assume the value of 570 Nm/deg is a good measurement for the total roll stiffness of the car.

5.3 Acceleration

For the acceleration event case, 3 scripts have been written in MATLAB to simulate the acceleration event. The first two scripts do not take into account rolling and air resistance and the first script do not take into account weight transfer when accelerating. These simplifications have been done to see what effect neglecting these forces will have on the over all result, and how much it will effect the accuracy of the result. In addition to the simulations, a quick calculation based on the formulas of constant acceleration was done. This resulted in a acceleration event time of 3.32 seconds. Based on results from the Formula Student event in 2016, a 3.32 seconds acceleration time is to low for a real life case [14]. The team with the fastest lap on the acceleration event managed a time of 3.78 seconds. This car also had all wheel drive increasing the available traction from the tires, and control systems to reduce the amount of wheel slippage. Based on the results from the acceleration event in 2016, times around 4.1 seconds is realistic for a rear wheel drive car, and will be used as the reference value in this case.

Figure 5.9 show the distance covered with respect to time. By looking at where each line ends, it is possible to determine what the finishing time of each simulation would be. The plot shows that all of the simulations estimates lower event times than what have been done by a rear wheel drive car in the competition. OptimumL is the one closest to the reference value with a time of 3.96 seconds. Next is ADAMS/Car with a time of 3.78 seconds and the 3rd MATLAB script at 3,73. The 1st and 2nd MATLAB script actually do not deviate at all and finish at a time of 3.54 seconds. This will be looked into further later in this chapter.

The next plot in Figure 5.10 show the longitudinal acceleration depending on time for the 5 simulations. There is a reoccurring trend for each of the simulations with a relatively constant acceleration in the beginning before it drops. This flat section can be caused by two factors, the motor reach its max torque or the tires are slipping. There are also differences in the amount of longitudinal acceleration each simulation sets as its max. All have the same motor torque inputs, so the max values are limited by the resistances acting against the acceleration. This is the rolling resistance in the tires, the air resistance and the efficiency of the drivetrain. It can be seen that Script 1 and 2 reaches the highest acceleration value, and hold this value constant for the entire section. For the other 3 simulations, the acceleration is reduces as the speed is increased. This is most likely due to aerodynamic drag which increases as the velocity increases. The other main reason for the lower acceleration is the tire rolling resistance, this is not affected by speed, but keeps constant over the speed region the car is operating in. The flat acceleration section lasts until the RPM of the motor reaches around 3200 RPM. If the speed is increased over this RPM limit, the torque in the motor has to be reduced to not consume more than the allowed 80 kW in the competition [15]. In the plot of the acceleration from the MATLAB Scripts, the acceleration drops to zero right before they reaches the end. This is the RPM limited zone, where the motor has reached its rated RPM and is not allowed to spin any faster.

The plot in Figure 5.11 show the acceleration with lower tire friction. Here it is possible to differentiate the effect of load transfer to the rear wheels in Script 2. Where Script 1 max out at about $9.4m/s^2$, which is the maximum traction force on the wheels with no weight transfer, Script 2 reach $13m/s^2$.

Figure 5.9: Distance vs Time plot for the acceleration event

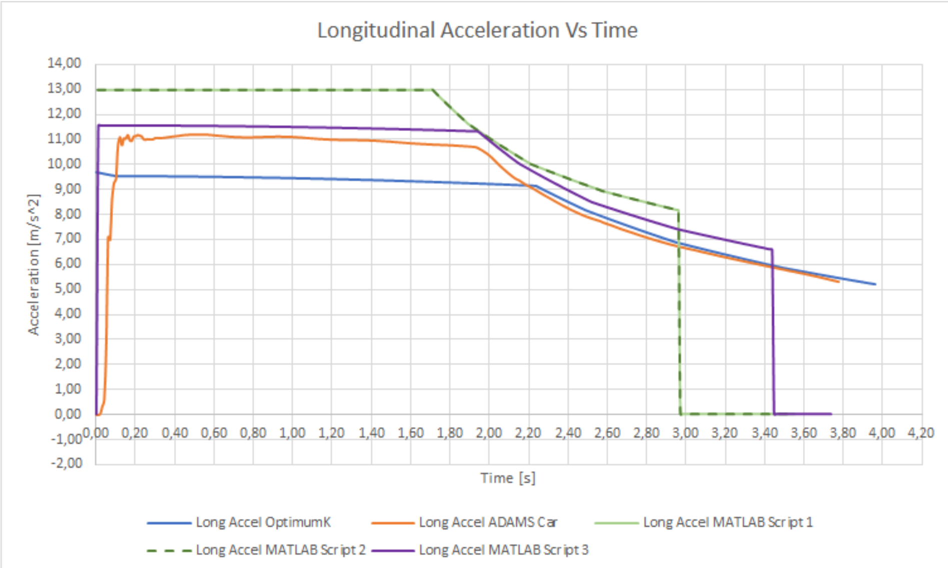


Figure 5.10: Longitudinal acceleration vs time for the acceleration event

The last plot in Figure 5.12 show the speeds in the different simulations. The most notable in this plot is the max speed limit which is shown on all the MATLAB script plots. Here the speed stops to increase and flats out at 125 km/h, which is the max speed of the car when the motor is running at max RPM. It can be seen that the ADAMS/Car plot runs a bit over this speed limit, even though it is defined in the powertrain subsystem. As it is not possible to see what ADAMS/Car do in the simulation, it is not possible to say for certain what the reason for this is. As ADAMS/Car include the inertia of the drivetrain and motor in its simulation, it might be possible the inertia gives a overshoot of the max RPM value before it drops down and settles.

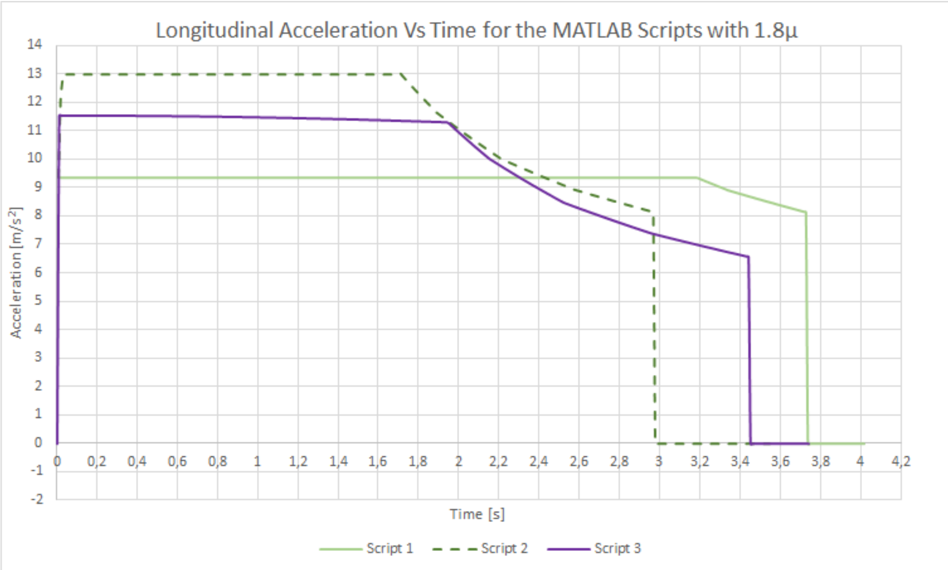


Figure 5.11: Longitudinal Acceleration vs time for the 3 MATLAB Scripts with $\mu = 1.8$

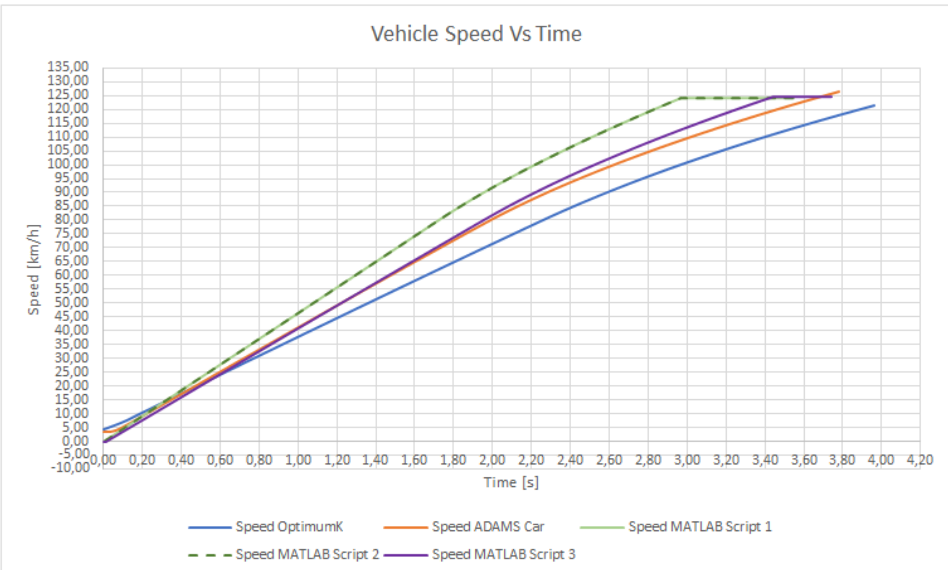


Figure 5.12: Speed vs time plot for the acceleration event

Chapter 6

Discussion

With all the results discussed in the previous chapter, this chapter will be used to discuss errors and possible changes in the simulations. Here the major causes for errors and differences will be mentioned for the Kinematic analysis and the Acceleration event simulation.

6.1 Kinematics

The kinematic simulations highlighted some differences in the way the two simulation tools defined a few of the values. Especially toe and camber was different as for ADAMS/-Car the camber affected the toe angle, which was not expected at first. After some reading through the ADAMS/Car manual it answered most of the questions concerning the toe and camber, and helped get an understanding on what caused the differences. With the way ADAMS/Car define camber, with referencing the vehicle vertical vs the wheel vertical, is harder to measure in real life than the way OptimumK defines it. With OptimumK's definition, it is possible to replicate and measure directly on the wheels of the car and compare the results. To compare real life measurements with results from ADAMS/Car, the roll angle of the chassis need to be known and the camber angle of the wheels have to be subtract from the roll angle before the angle which is measurable is defined.

When comparing the measured toe and camber angles, the most notable is the difference in accuracy of the data. Toe varies a lot compared to the simulated data, but the values are so small it is difficult to measure it accurately. There was some bending in the beams which have affected the measurement, and in addition there was some slop in the steering on the left wheel. This gave about 6 mm movement at the measuring point, which results in 0.34 deg of movement. Compared to the simulated results this is over half of the total movement from the simulations. Based on the tool available to measure the toe at the moment, the difference in toe in the simulation is not large enough to be of any concern.

The difference on the negative toe from the simulations, is about 0.1 deg. This is the same as 1.74 mm on the measuring rig. It is possible to measure, but with the slop in the left wheel and some bending in the beam, this small variation will not be detectable.

The camber measurements seem a lot closer to the simulated values. As camber was measured directly on the wheels, there are less causes of error which give more accurate results. In addition, the values are higher than for the toe, which reduces the over all effect of 0.1 deg in variation.

The roll center position was another difference in the two results. As mentioned ADAMS/Car uses what is called the force method. It is defined in Ref [3] as ”That point in the transverse vertical plane through any pair of wheel centers and equidistant from them, at which lateral forces may be applied to the sprung mass without producing an angular (roll) displacement of the sprung mass.”. This differs from the geometric definition, which uses the intersection point between projections from the instant centers of the wishbones to the center of the tire contact patch, to define the roll center position. This is most likely what OptimumK uses, and is the most common way of locating the roll center. The geometric definition assumes that all members in the suspension system and chassis are rigid bodies with no compliance. As this is not possible in real life, the force method will give a more accurate position, but in simulations, it requires a accurately modeled system with the correct stiffness for the linkages and parts. The force method is what is used in the industry to determine the roll center of cars on test rigs.

6.2 Roll Stiffness

The roll stiffness calculations, simulations and measurements gave results which did not differ a lot. There was a difference in the rear roll stiffness of the calculated and the simulated values. This can be a result of the assumption of a stiff lever arm as discussed in the results chapter. As a test rig for the suspension is not available for the team, it was not possible to measure the front and rear roll stiffness's individually. This could have helped determining which of the two results gave the best answer. It might be an idea for a bachelor thesis to develop and build a test rig for the suspension to do real life testing and be able to compare it with simulated results.

6.3 Acceleration

As mentioned in the acceleration results in chapter 5.3, the simulations give a much lower finishing time on the acceleration event then what have been done by top rear wheel drive teams in the competition. The first, and probably the main reason for these optimistic results, is the tire data. The tire data used to model the tire is collected from a rolling road. A rolling road is a steel belt with a surface coating that mimic a road surface. The wheel with the tire being tested, is mounted on a rig which can apply driving forces, turn the wheel, tilt the wheel and measure reaction forces from these movements [16]. As the wheel is put through cycles on the tire testing machine, rubber from the tire is worn of, and get stuck to the rolling road. The rubber that gets stuck on the rolling road increases the traction between the tire and the road, as rubber sticks better to hot rubber than steel.

The increase in traction increases the reaction forces in the rig and results in friction values higher then is achievable on roads or even clean race tracks. In addition, for the MATLAB scripts and OptimumL the tire friction coefficient is assumed to be a constant, but in reality it depends on a lot of factors, but mainly the normal load on the tire. As the load is increased the friction coefficient is reduced . The tire data can, unfortunately, not be discussed openly in this thesis as it is only available for Formula Student teams which have payed to gain access to the FSAE Tire Test Consortium (FSAETTC). For the ADAMS/Car simulation, the tire model was sourced directly from the FSAETTC website and have been developed, using the data from the testing, to make a model for ADAMS/Car. This model still have the flaws with higher friction coefficient then a real road can manage. Another possible error in the ADAMS/Car simulation is the aerodynamic properties of the car. These have not been modified and are based on the FSAE2012 template and can therefor give the wrong amount of drag and lift.

Ref p dette, mulig Audun si bachelor oppg

Another aspect which will reduce the overall acceleration is the definition of the 80 kW rule. 80 kW is the max amount of power allowable to be drained from the accumulator at any times in an event. The calculation of the motor torque have not taken into account power loss in the motor controller and all the different losses inside the electric motor. Therefor the real torque output from the motor would be somewhat lower than used in the simulations. The overall efficiency of both the motor and the motor controller is unknown. It is therefor hard to tell how much power is lost in the system. Based on purely heat generation in the motor controller and motor, there is a loss of about 10%. In addition, there are a small loss in the cables, but this can be neglected as the cable stretches are short. To get a realistic result, a safe approximation could be around 13% loss in the electric side of the drive train. There are also some losses in the gearbox and differential, but these are in the 3-5% region and are taken into account in the simulation by the templates.

Chapter 7

Conclusion

From the simulation and measured results we can see that for kinematic analysis both OptimumK and ADAMS/Car give closely the same results. The difference between the software comes mainly in the form of how long time is needed to set up the simulation. OptimumK can be set up by an inexperienced user quite easily, while ADAMS/Car requires a deal of knowledge of the software to be able to run simulations. For fast simulations used to decide on a design choice OptimumK gives acceptable results which can be trusted. ADAMS/Car would be more suited for advanced in-depth analysis of ride, comfort and handling of a car where dynamic and transient response of the vehicle is analyzed. In the roll stiffness case, both the calculation and the simulation gave accurate results. If only roll stiffness is needed, the time spent setting up and running the simulation in ADAMS/Car would be better spent somewhere else on the car. The calculation will give an acceptable answer. But if a full vehicle assembly is already set up, all the other parameters available through the SPMM simulation will give most of the car parameters needed in documentation of the vehicle in the design event of the competition.

For the acceleration event all of the simulations underestimated the time needed to run the acceleration event. This is most likely caused by the inputs of the motor torque being higher than what is achievable, and the tire friction coefficient being higher than achievable on a track. But it can be concluded that air and rolling resistance need to be a part of the calculations in order to determine a realistic acceleration time. This will be even more important if an aerodynamic package would be mounted on the car. In the concept and design phase of a new car OptimumL would give good answers on how a vehicle will perform in dynamic events. As it simplify the car, the overall design and layout is not that important, and is therefor useful before the car is even designed. When a car have been designed, it would be useful to run tests in ADAMS/Car to test different springs, dampers and ARB stiffness's before the car have been produced. This requires someone on the team to start experimenting with ADAMS/Car from the beginning and learn the software.

Overall the OptimumG software package supplies what Formula Student teams needs. A easy to use software which gives reasonably accurate results in a short time, giving more time to fine tune the car. It can also run dynamic simulations using OptimumD, but as there was some problems with the software while working on this thesis, it was not possible to run simulations. ADAMS/Car is a complex software, which can do everything the software in the OptimumG package can do, and even more. It can give the user information on how the car handles and reacts to inputs before it is built. But it requires accurate input information to give accurate results.

This is probably the most important conclusion from this thesis. The kinematics and roll stiffness simulations ran on inputs generated in a software and the car was manufactured using the coordinates and the same components as was defined in the software. This resulted in the measured values to be close to the simulated values. For the acceleration event, a lot of assumptions was made. The motor torque assumed no loss in the HV-control system, the tires used data from a higher friction surface, and some of the simulations did not take into account resistances acting against the car. This led to simulation which did not give realistic results. Figure 4.1 showed the software black box where all the calculations are done. If the inputs are wrong, the black box do not fix these and gives right outputs. It requires accurate input to produce accurate output.

For further work it is recommended to gather accurate inputs to work with. This could be done with a more comprehensive processing of the tire and motor data. Collecting test data from the car directly by driving it or running it on a test bench. And keep on comparing simulated results with what is available of test data in the team, from other teams, or from competition results.

Bibliography

[1] K. Popp and W. Schiehlen, *Ground Vehicle Dynamics*. Springer, 2010.

[2] D. Schramm, M. Hiller, and R. Bardini, *Vehicle Dynamics Modeling and Simulation*. Springer, 2014.

[3] D. L. Milliken and W. Milliken, *Race Car Vehicle Dynamics*. SAE International, 1995.

[4] "History formula sae," SAE International. [Online]. Available: https://www.fsaeonline.com/page.aspx?pageid=c4c5195a-60c0-46aa-acbf-2958ef545b72

[5] S. Rao, *Mechanical Vibrations, Fifth Edition in SI Units*. Pearson, 2011.

[6] J. K. Vandiver, "Engineering dynamics, lecture 15: Introduction to lagrangian mechanics," 2011. [Online]. Available: https://www.youtube.com/watch?v=zhk9xLjrmi4

[7] MathWorks. [Online]. Available: https://se.mathworks.com/discovery/what-is-matlab.html

[8] "Optimumlap, vehicle dynamics simulation, simplified," OptimumG. [Online]. Available: http://www.optimumg.com/software/optimumlap/

[9] *OptimumKinematics - Help File*, OptimumG. [Online]. Available: http://downloads.optimumg.com/Software/OptimumKinematics/website/Help_File.pdf

[10] *OptimumDynamics Computational Vehicle Dynamics - Help File*, OptimumG. [Online]. Available: http://downloads.optimumg.com/Software/OptimumDynamics/website/Help_File.pdf

[11] *Getting Started with ADAMS/Car*, MSC Software. [Online]. Available: https://simcompanion.mscsoftware.com/infocenter/index?page=content&id=DOC11408&cat=2017.2_ADAMS_DOCS&actp=LIST

[12] A. Staniforth, *Competition Car Suspension, Fourth Edition*. Haynes Publishing, 2006.

[13] "Fsae2012 template." [Online]. Available: https://simcompanion.mscsoftware.com/infocenter/index?page=content&id=KB8020723

[14] IMechE, "Formula student uk results 2016." [Online]. Available: http://www.imeche.org/docs/default-source/1-oscar/formula-student/fs2016_results.zip?sfvrsn=2

[15] "Formula student rules 2018," Formula Student Germany. [Online]. Available: http://www.imeche.org/docs/default-source/1-oscar/formula-student/fs-rules_2018_v1-1.pdf?sfvrsn=2

[16] E. M. Kasprzak and D. Gentz, "The formula sae tire test consortium - tire testing and data handling," 2006. [Online]. Available: http://www.millikenresearch.com/TTC_SAE_paper.pdf

Appendix A

ADAMS Car Tutorial

This appendix will explain the process of developing the templates and full vehicle setup for the ADAMS/Car simulation. It is based on the FSAE2012 template available from the MSC website [13], with modifications to model ION Racings 2018 car Bifrost. It is recommended to read through and do the tutorials for basic ADAMS/Car setup and simulation before starting this tutorial. The main modifications are the powertrain to change it from internal combustion with a transmission to a electric motor drive train with a single speed gearbox. Other modifications are changing from a push rod to pull rod in the front suspension, and changing the chassis to visualize the car better. Other changes have been moving the hard points to their respective positions for the 2018 car.

A.1 Initial setup

When ADAMS/Car is installed and ready to use, some modifications to the *.acar.cfg* file have to be done. This is to get access to the expert user mode and the Template Builder. The *.acar.cfg* file can be found in your local user folder and can be edited with Notepad. To access the expert mode, change the line

```
ENVIRONMENT MDI_ACAR_USERMODE standard
```

as shown in figure A.1

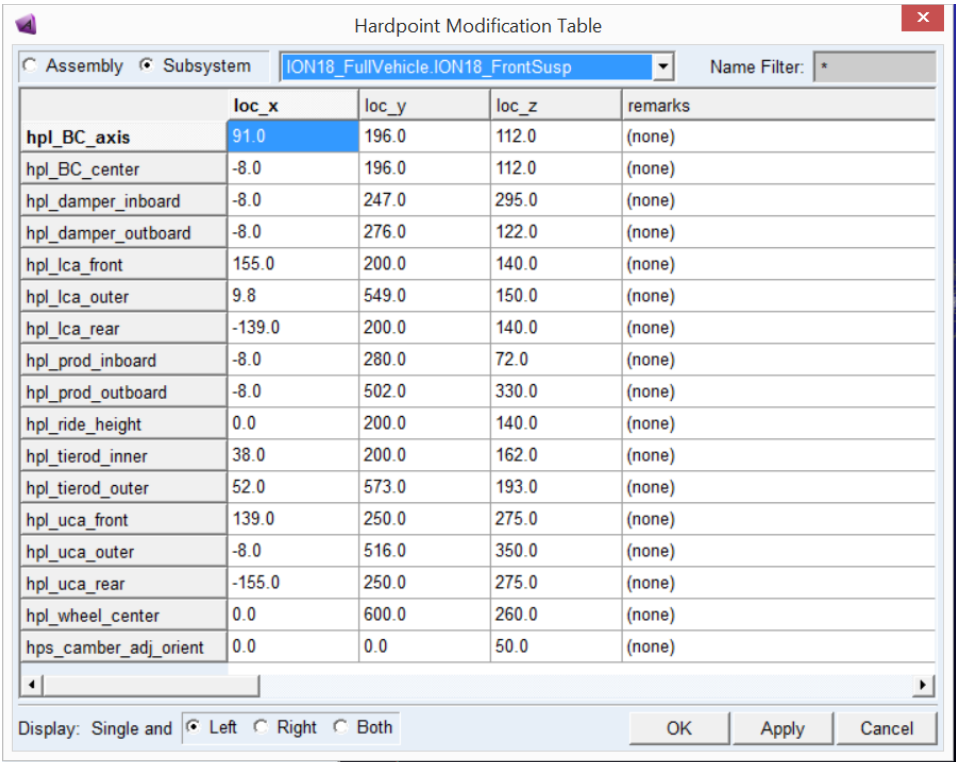
```
!
!-----------------------------------------------------------------------!
! - List of personal environment variables
!-----------------------------------------------------------------------!
! Desired user mode (standard/expert)
ENVIRONMENT   MDI_ACAR_USERMODE        expert
ENVIRONMENT   MDI_ACAR_VEHICLE_REAR    -1,0,0
ENVIRONMENT   MDI_ACAR_VEHICLE_LEFT    0,1,0
!
!-----------------------------------------------------------------------!
! - List of personal database directories
!            Database name     Path of Database
!-----------------------------------------------------------------------!
```

Figure A.1: Modified version of the .acar.cfg file to access the Expert Mode

Next, make a work folder where you would like to save all files related to the project. Start by making a folder in the file explorer, and then open ADAMS/Car. Open Tools→Database Management→Create Database. Here a new database can be created and added to the ADAMS/Car database set. To change the default database to the one just created, we go to the Open Tools→Database Management→Set Default Writable and choose the database just created. Extract the FSAE2012 files into the folder

A.2 Template Modification

Start with changing from push to pull rod in the front suspension. To do this, you have to open the *_fsae_frontsusp.tpl* from the *templates.tbl* folder. Here find and open *jorsph_prod_to_LCA* under Attachement → Joints. The I Part should be changed from the lower control arm to the upper control arm. Now it is possible to start changing the geometry by modifying the hardpoints. Figure A.2 show the hard points for the front suspension. Do the same for the rest of the suspension templates. Hardpoints are shown in the following figures.



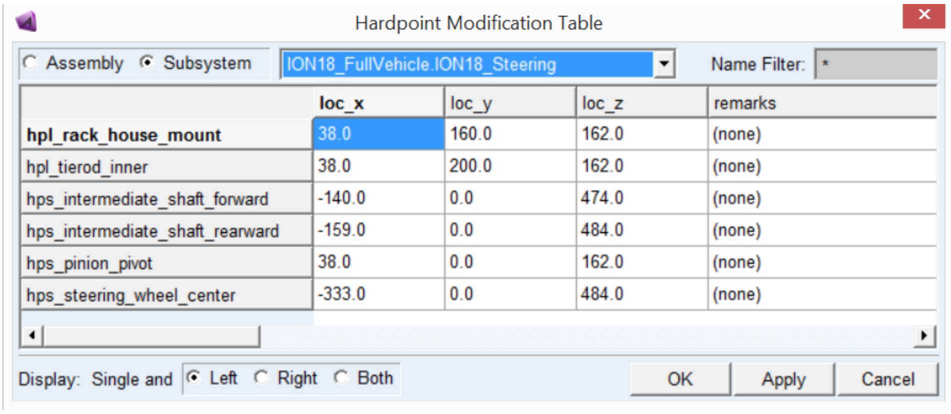
	loc_x	loc_y	loc_z	remarks
hpl_BC_axis	91.0	196.0	112.0	(none)
hpl_BC_center	-8.0	196.0	112.0	(none)
hpl_damper_inboard	-8.0	247.0	295.0	(none)
hpl_damper_outboard	-8.0	276.0	122.0	(none)
hpl_lca_front	155.0	200.0	140.0	(none)
hpl_lca_outer	9.8	549.0	150.0	(none)
hpl_lca_rear	-139.0	200.0	140.0	(none)
hpl_prod_inboard	-8.0	280.0	72.0	(none)
hpl_prod_outboard	-8.0	502.0	330.0	(none)
hpl_ride_height	0.0	200.0	140.0	(none)
hpl_tierod_inner	38.0	200.0	162.0	(none)
hpl_tierod_outer	52.0	573.0	193.0	(none)
hpl_uca_front	139.0	250.0	275.0	(none)
hpl_uca_outer	-8.0	516.0	350.0	(none)
hpl_uca_rear	-155.0	250.0	275.0	(none)
hpl_wheel_center	0.0	600.0	260.0	(none)
hps_camber_adj_orient	0.0	0.0	50.0	(none)

Figure A.2: Hardpoint coordinates for the front suspension on Bifrost

The origin in X-axis is assumed to be in the center of the front axle. The default ADAMS/Car coordinate system define X positive towards the rear axle. In the OptimumG software package X-axis is positive towards the front of the car. This makes the Y-axis point toward the right compared to Optimum where it points to the left.

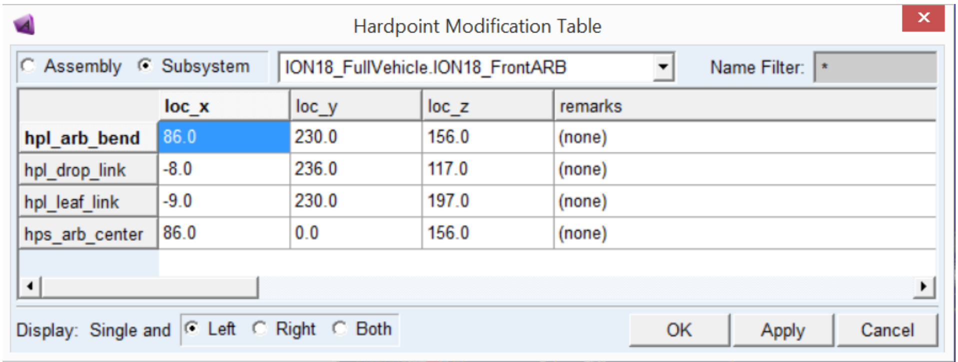
A.3 Subsystem generation

With all the different templates ready, we can start with creating subsystems. To create a subsystem, go to File→New→Subsystem. Here you define the name of the subsystem you are creating, choose the template the system should be based on and if it is a front, rear or trailer system. All subsystems on the front axle should have the front system chosen and all the rear subsystems should have the rear system position. Subsystems with different positions cannot be combined into an assembly. The trailer system can be ignored for our case. Do this for all the subsystems in the suspension. As the hardpoints have been defined in the templates, no modification to the subsystem hardpoints have to be done.



	loc_x	loc_y	loc_z	remarks
hpl_rack_house_mount	38.0	160.0	162.0	(none)
hpl_tierod_inner	38.0	200.0	162.0	(none)
hps_intermediate_shaft_forward	-140.0	0.0	474.0	(none)
hps_intermediate_shaft_rearward	-159.0	0.0	484.0	(none)
hps_pinion_pivot	38.0	0.0	162.0	(none)
hps_steering_wheel_center	-333.0	0.0	484.0	(none)

Figure A.3: Hardpoint coordinates for the steering system on Bifrost



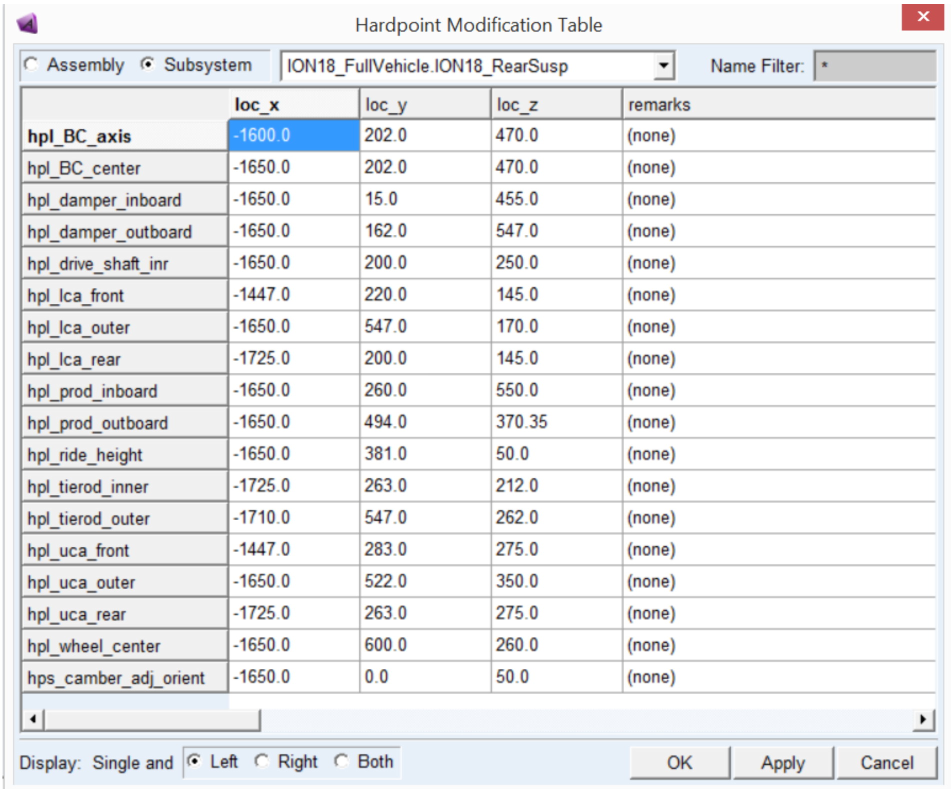
	loc_x	loc_y	loc_z	remarks
hpl_arb_bend	86.0	230.0	156.0	(none)
hpl_drop_link	-8.0	236.0	117.0	(none)
hpl_leaf_link	-9.0	230.0	197.0	(none)
hps_arb_center	86.0	0.0	156.0	(none)

Figure A.4: Hardpoint coordinates for the front ARB system on Bifrost

For the Powertrain, Chassis and Tire subsystems the ones from the FSAE2012 can be used as a start. No modifications to the templates need to be done, but some changes need to be done in the Powertrain subsystem to make it act as an electric motor instead of an internal combustion engine. Under Parameter Variables changes need to be made to the following variables:

- pvs_max_gears = 1
- pvs_engine_rev_limit = 5500
- pvs_engine_idle_speed = 1
- pvs_final_drive = 1
- pvs_gear_1 = 3.93

The max gears is changed to prevent the powertrain to use more then the first gear, as Bifrost only got one gear. The rev limit is set to 5500 RPM as this is the limit the motor is rated for. Engine idle speed is set to 1 as it have to have a positive value, but

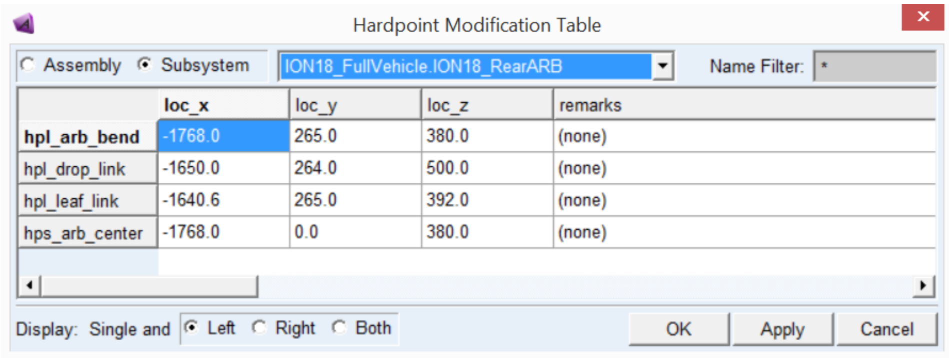


	loc_x	loc_y	loc_z	remarks
hpl_BC_axis	-1600.0	202.0	470.0	(none)
hpl_BC_center	-1650.0	202.0	470.0	(none)
hpl_damper_inboard	-1650.0	15.0	455.0	(none)
hpl_damper_outboard	-1650.0	162.0	547.0	(none)
hpl_drive_shaft_inr	-1650.0	200.0	250.0	(none)
hpl_lca_front	-1447.0	220.0	145.0	(none)
hpl_lca_outer	-1650.0	547.0	170.0	(none)
hpl_lca_rear	-1725.0	200.0	145.0	(none)
hpl_prod_inboard	-1650.0	260.0	550.0	(none)
hpl_prod_outboard	-1650.0	494.0	370.35	(none)
hpl_ride_height	-1650.0	381.0	50.0	(none)
hpl_tierod_inner	-1725.0	263.0	212.0	(none)
hpl_tierod_outer	-1710.0	547.0	262.0	(none)
hpl_uca_front	-1447.0	283.0	275.0	(none)
hpl_uca_outer	-1650.0	522.0	350.0	(none)
hpl_uca_rear	-1725.0	263.0	275.0	(none)
hpl_wheel_center	-1650.0	600.0	260.0	(none)
hps_camber_adj_orient	-1650.0	0.0	50.0	(none)

Figure A.5: Hardpoint coordinates for the rear suspension on Bifrost

electric motors do not have an idle speed. 1 is Ok for most cases as the car will not run at speeds low enough for this to take effect, except for a few cases discussed later. The final drive is set to 1 and gear 1 is set to 3.93. These two could be changed around or two values could be set, as long as the final gear ratio of the two is 3.93. To easily modify these values, open the Parameter Variable Modification Table under Adjust→Parameter Variable→Table. Here all parameters in a subsystem can be changed in one table, figure A.7 show the table for the Bifrost Powertrain.

With the engine speeds and gearing set, the engine moment table have to be set. The easiest way to modify the table is to open the property file in Notepad and change it there. The file can be found in the working folder generated in the beginning, under powertrains.tbl. in the FSAE2012 template it is called 600cc_engine_map.pwr. This should be copied and the new file should be called ElectricMotor or something similar. Open it and change the data under Engine to match as shown in figure A.8. Save the file and it is updated in the powertrain subsystem.



	loc_x	loc_y	loc_z	remarks
hpl_arb_bend	-1768.0	265.0	380.0	(none)
hpl_drop_link	-1650.0	264.0	500.0	(none)
hpl_leaf_link	-1640.6	265.0	392.0	(none)
hps_arb_center	-1768.0	0.0	380.0	(none)

Figure A.6: Hardpoint coordinates for the rear ARB system on Bifrost

To change the chassis shape requires a bit more work. The nodes for the chassis points are not accessible through ADAMS/Car after the chassis have been created. There for the _fsae_chassis.tpl file have to be modified manually. This file is found in the templates.tbl folder. A bit down in the file the lines in figure A.9 is shown. All the G175, G382 and so on are the markers which define the location of the points which define the outline of the chassis. Table A.1 show the coordinates for each point to define the chassis on Bifrost.

With these modifications, the subsystems overal layout is ready. What is left it some changes to the steering system, defining the spring stiffnesses and the damping. Under the steering systems Parameter Variables rack displacement should be set to 36mm and max steer angle to 100. Under Gears and Reduction Gear the pinion to rack reduction ratio should be set to 8.33E-02. and neither the pinion to rack or steering wheel column should have invert output direction. In addition when the steering subsystem is added to an assembly the steering ratio and rack ratio has to be set. Figure A.10 show the full-vehicle analysis setup parameters. If this is not changed from the standard values in the template, the steering is to responsive resulting in the simulations having problems keeping the steering straight.

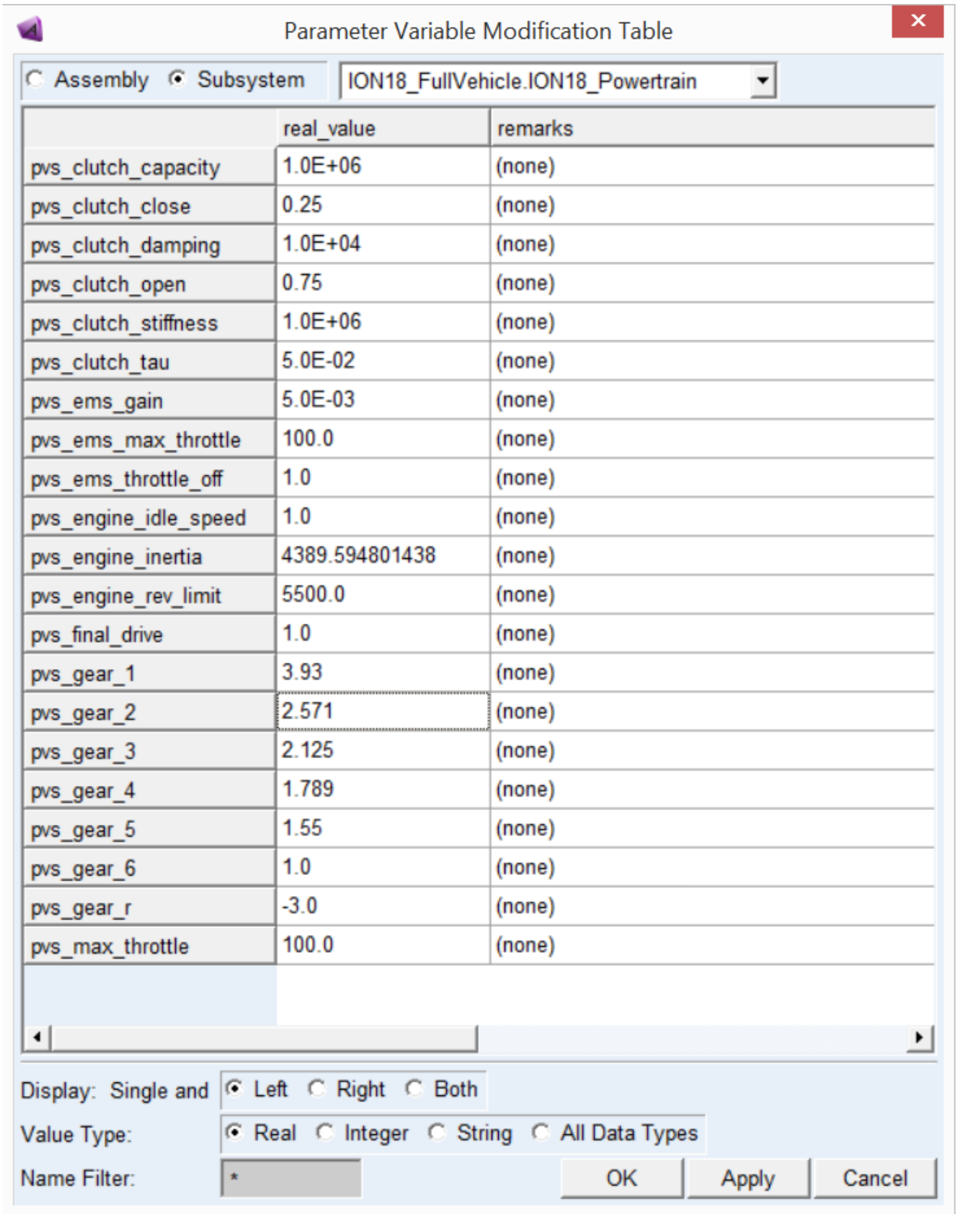
Next up is the spring and damper modification. These are property files which can be modified like the engine torque file. The spring file is found under the springs.tbl folder. For Bifrost CaneCreek 350 springs are used. This is springs with a stiffness of 350 lb./in which is 61.3 N/mm. Copy the msc_0001 file and rename it to Ohlins350.spr. Open it and change the values in the file to match figure A.11.

For the dampers the files can be found in the dampers.tbl folder in the database. Do the same as with the springs and copy the original file, rename it and modify it according to figure A.12.

Table A.1: Table over the chassis outline points in the chassis template file

Point	X	Y	Z
G175	450	-400	258
G382	817	-685	251
G522	1186	-579.43	1282.43
G1676	623	-707	906
G2160	-593	-700	669
G2213	-689	-400	259.5
G2610	1932.27	-685	358.5
G3164	1563.78	-685	247
G3273	2411	-620.85	434.37
G3454	3520.1	-627	505
G5741	3506.17	-600	989.1
G5815	3066.52	-624.57	1000.27
G5869	2435	-561.2	1286.5
G10175	450	400	258
G10382	817	685	251
G10522	1186	579.43	1282.43
G11676	623	707	906
G12160	-593	700	669
G12213	-689	400	259.5
G12610	1932.27	685	358.5
G13164	1563.78	685	247
G13273	2411	620.85	434.37
G13454	3520.1	627	505
G15741	3506.17	600	989.1
G15815	3066.52	624.57	1000.27
G15869	2435	561.2	1286.5

With all these modifications done, the subsystems can be joined into a suspension assembly or a full vehicle assembly. With the assemblies it is possible to run simulations just like in the simulation tutorials supplied by MSC Software for ADAMS/Car. To run the kinematic simulations a front suspension assembly was created with the front suspension, steering and front ARB subsystems. The Acceleration simulations need a full vehicle assembly. The acceleration event was created with a modified acceleration event using the event builder.



	real_value	remarks
pvs_clutch_capacity	1.0E+06	(none)
pvs_clutch_close	0.25	(none)
pvs_clutch_damping	1.0E+04	(none)
pvs_clutch_open	0.75	(none)
pvs_clutch_stiffness	1.0E+06	(none)
pvs_clutch_tau	5.0E-02	(none)
pvs_ems_gain	5.0E-03	(none)
pvs_ems_max_throttle	100.0	(none)
pvs_ems_throttle_off	1.0	(none)
pvs_engine_idle_speed	1.0	(none)
pvs_engine_inertia	4389.594801438	(none)
pvs_engine_rev_limit	5500.0	(none)
pvs_final_drive	1.0	(none)
pvs_gear_1	3.93	(none)
pvs_gear_2	2.571	(none)
pvs_gear_3	2.125	(none)
pvs_gear_4	1.789	(none)
pvs_gear_5	1.55	(none)
pvs_gear_6	1.0	(none)
pvs_gear_r	-3.0	(none)
pvs_max_throttle	100.0	(none)

Figure A.7: The Parameter Variable Modification Table for the ION_Powertrain subsystem

File Edit Format View Help

```
$---------------------------------------------------------------MDI_HEADER
[MDI_HEADER]
 FILE_TYPE     =  'pwr'
 FILE_VERSION  =  1.0
 FILE_FORMAT   =  'ASCII'
$--------------------------------------------------------------------UNITS
[UNITS]
(BASE)
{length  force      angle       mass     time}
'mm'    'newton'   'degrees'   'kg'     'sec'
(USER)
{unit_type    length  force  angle  mass  time  conversion}
'Nmm'           1       1      0      0     0     1.0
'rpm'           0       0      1      0    -1     6.0
$
$     6.0 deg./s/rpm = (360 deg/revolution) * (1 min./ 60s )
$-------------------------------------------------------------------ENGINE
[ENGINE]
(Z_DATA)
{throttle <no_units>}
0.0
1.00
(XY_DATA)
{engine_speed <rpm>  torque <Nmm>}
0       0       0
1       -240    240000
500     -240    240000
1000    -240    240000
1500    -240    240000
2000    -240    240000
2500    -240    240000
3000    -240    240000
3183    -240    240000
3500    -215    215000
4000    -185    185000
4500    -165    165000
5000    -150    150000
5500    -130    130000
6000    -125    125000
```

Figure A.8: The motor torque file used in the powertrain subsystem

```
    }
}
DEF ges_chassis PART {
    ATTRIBUTES          ._fsae_chassis.ges_chassis.Attr
    REMARKS             ["Chassis Structure"]
    MASS                90.7184740000000005
    CM                  ._fsae_chassis.ges_chassis.cm
    USE_CALCULATED_MASS false
    IM                  ._fsae_chassis.ges_chassis.inertia_frame
    IP                  [2.20445299999315113E+08, 1.06721599999431062E+09, 1.18107800000194287E+09, 0.0, 0.0, 0.0]
    DEF cm MARKER {
       ATTRIBUTES       ._fsae_chassis.ges_chassis.cm.Attr
       LOC              [819.14999999999986358, 0.0, 279.39999999999997726]
       DEF Attr ATTRIBUTES {
          VISIBILITY    off
          NAME_VISIBILITY off
       }
    }
    DEF G175 MARKER {
       ATTRIBUTES       ._fsae_chassis.ges_chassis.G175.Attr
       LOC              [450.0, -400.0, 258.0]
       DEF Attr ATTRIBUTES {
          VISIBILITY    off
       }
    }
    DEF G382 MARKER {
       ATTRIBUTES       ._fsae_chassis.ges_chassis.G382.Attr
       LOC              [817.0, -685.0, 251.0]
       DEF Attr ATTRIBUTES {
          VISIBILITY    off
```

Figure A.9: The figure show Where the modification to the position of the chassis outline points are in the fsae_chassis template.

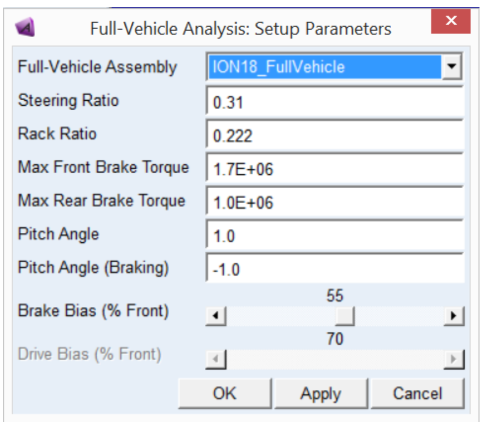
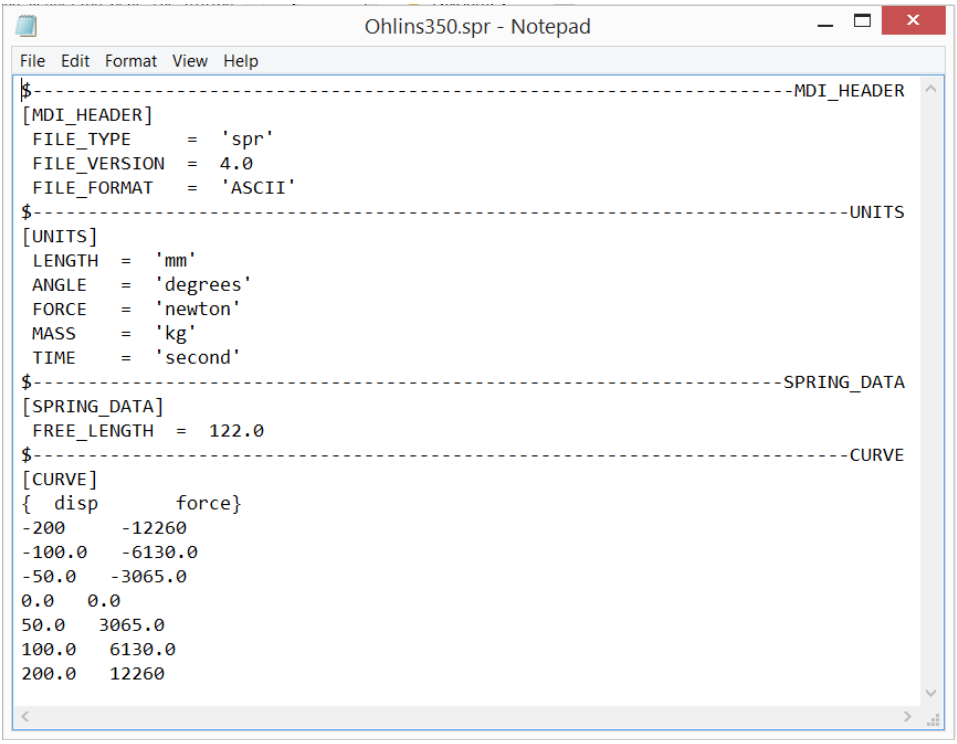


Figure A.10: Full-Vehicle Analysis Setup Parameters for the Bifrost full vehicle assembly.

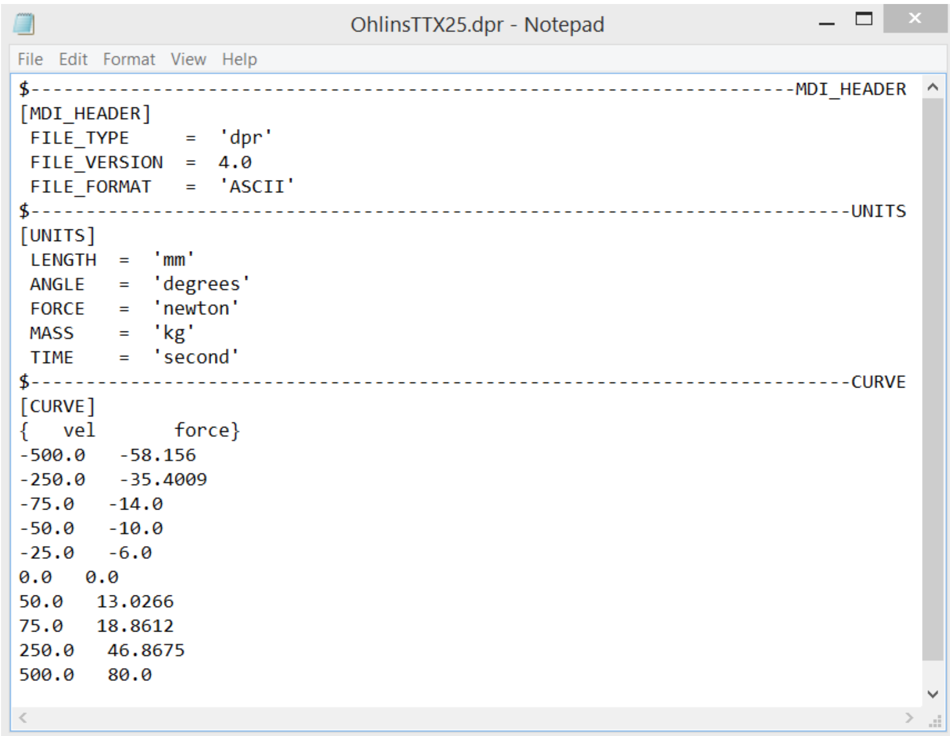


Ohlins350.spr - Notepad

File Edit Format View Help

```
$---------------------------------------------------------------------MDI_HEADER
[MDI_HEADER]
 FILE_TYPE     =  'spr'
 FILE_VERSION  =  4.0
 FILE_FORMAT   =  'ASCII'
$--------------------------------------------------------------------------UNITS
[UNITS]
 LENGTH   =  'mm'
 ANGLE    =  'degrees'
 FORCE    =  'newton'
 MASS     =  'kg'
 TIME     =  'second'
$--------------------------------------------------------------------SPRING_DATA
[SPRING_DATA]
 FREE_LENGTH  =  122.0
$--------------------------------------------------------------------------CURVE
[CURVE]
{  disp        force}
-200      -12260
-100.0    -6130.0
-50.0    -3065.0
0.0    0.0
50.0    3065.0
100.0    6130.0
200.0    12260
```

Figure A.11: The properties of the 350 springs used on the car.



OhlinsTTX25.dpr - Notepad

File Edit Format View Help

```
$-------------------------------------------------------------------MDI_HEADER
[MDI_HEADER]
 FILE_TYPE     =  'dpr'
 FILE_VERSION  =  4.0
 FILE_FORMAT   =  'ASCII'
$------------------------------------------------------------------------UNITS
[UNITS]
 LENGTH  =  'mm'
 ANGLE   =  'degrees'
 FORCE   =  'newton'
 MASS    =  'kg'
 TIME    =  'second'
$------------------------------------------------------------------------CURVE
[CURVE]
{   vel        force}
-500.0   -58.156
-250.0   -35.4009
-75.0   -14.0
-50.0   -10.0
-25.0   -6.0
0.0   0.0
50.0   13.0266
75.0   18.8612
250.0   46.8675
500.0   80.0
```

Figure A.12: The properties of the Ohlins TTX25 dampers used on the car.

Appendix B

Raw Results from ADAMS

B.1 Heave

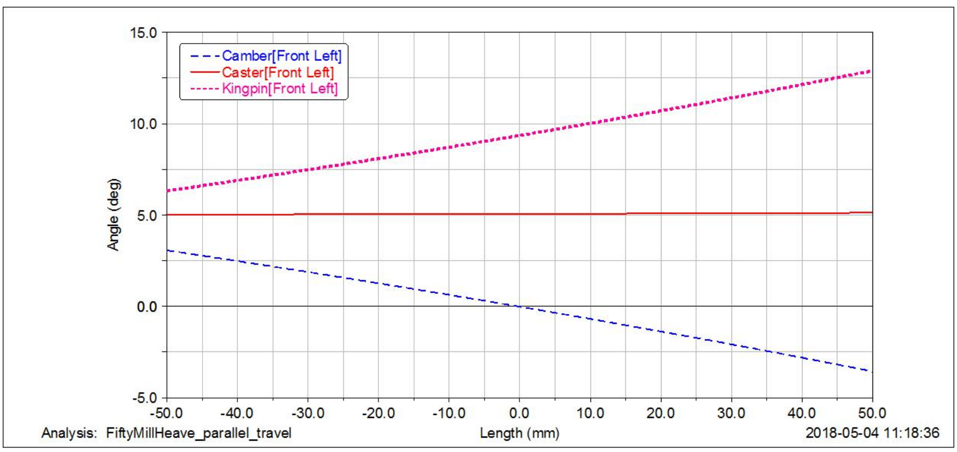


Figure B.1: Camber, Caster and Kingpin vs Wheel travel in ADAMS/Car

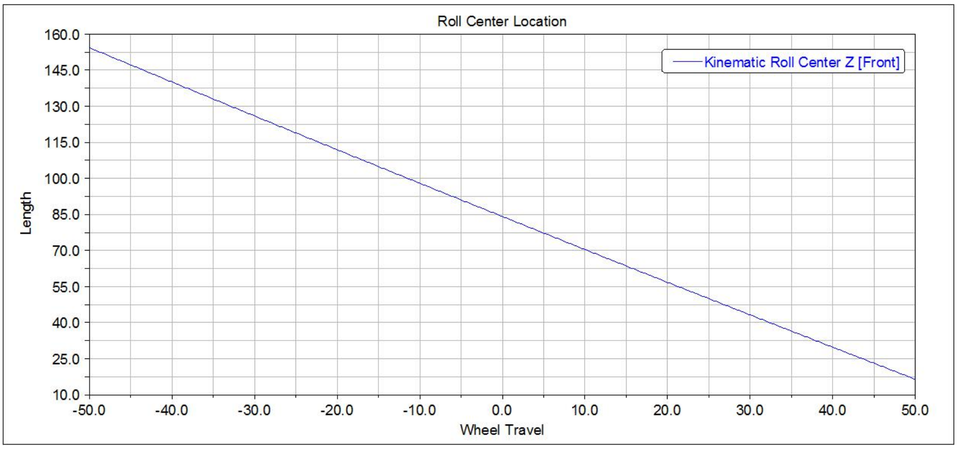


Figure B.2: Roll center vertical location vs wheel travel in ADAMS/Car

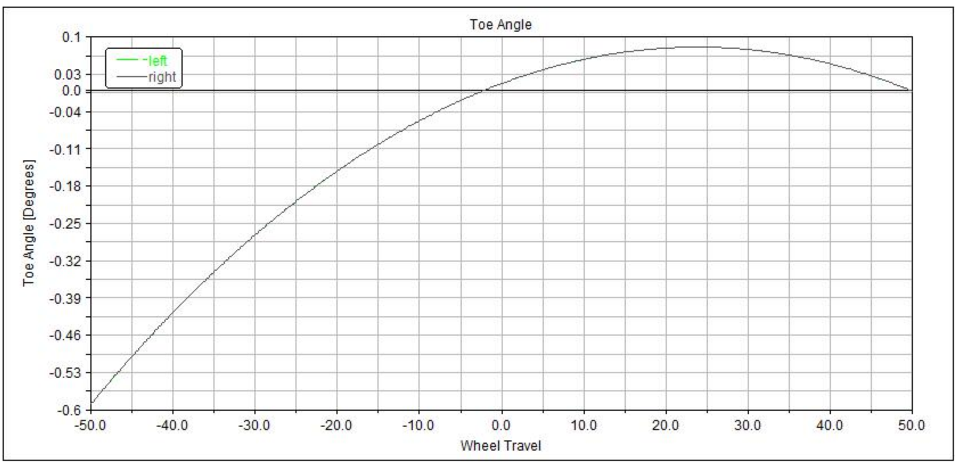


Figure B.3: Toe Change vs Wheel Travel in ADAMS/Car

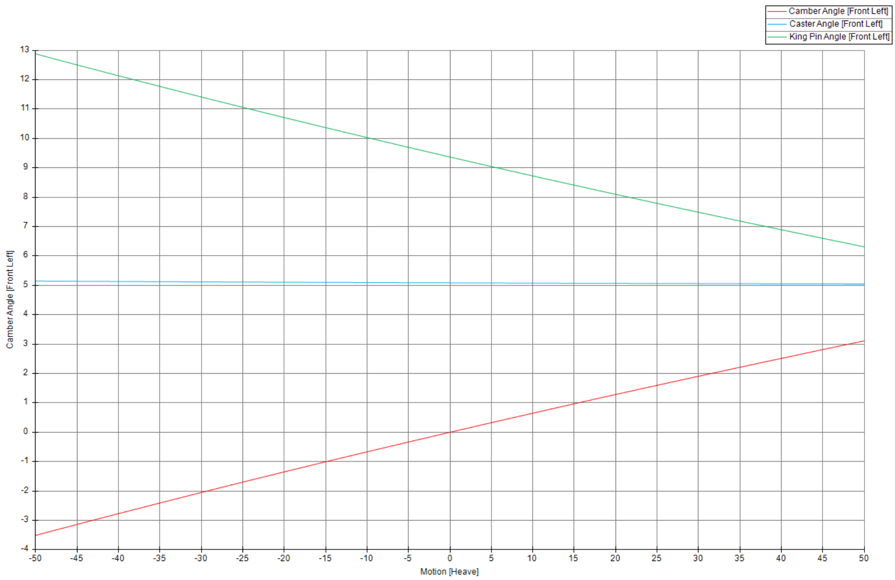


Figure B.4: Camber, Caster and Kingpin vs wheel travel in OptimumK

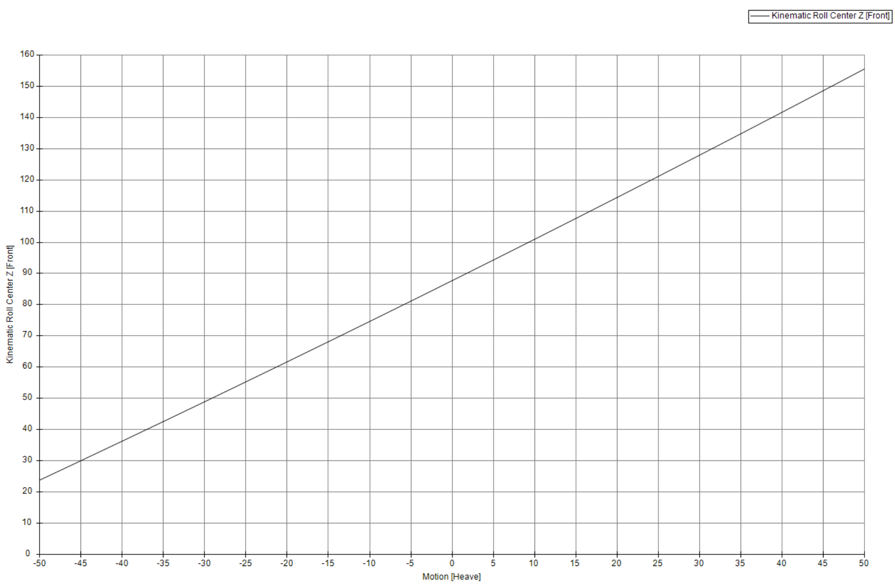


Figure B.5: Roll center vertial location vs wheel travel in OptimumK

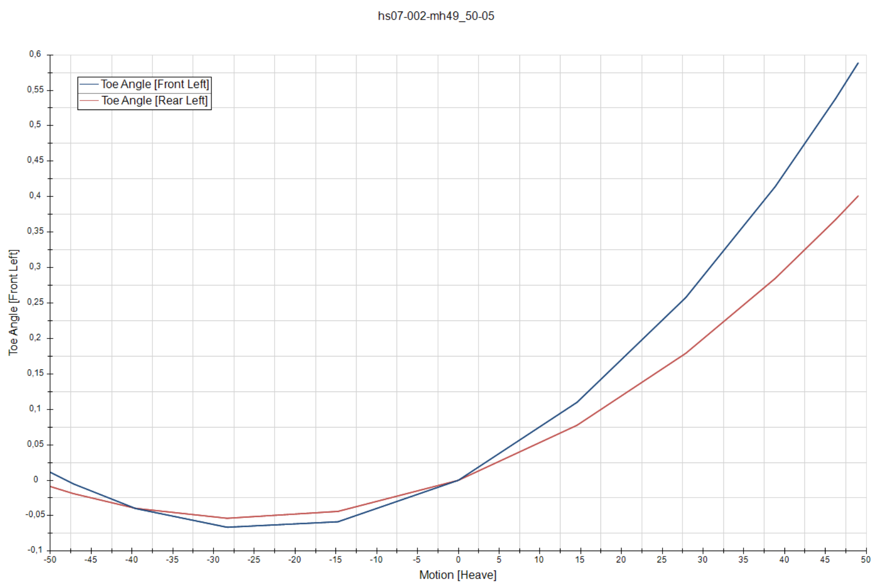


Figure B.6: Toe change vs wheel travel in OptimumK

B.2 Roll

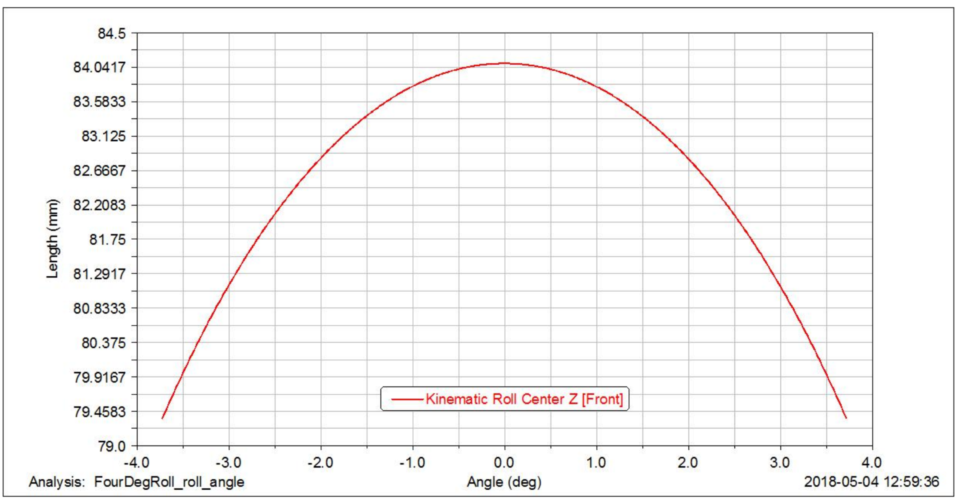


Figure B.7: Roll center vertical location vs chassis roll in ADAMS/Car

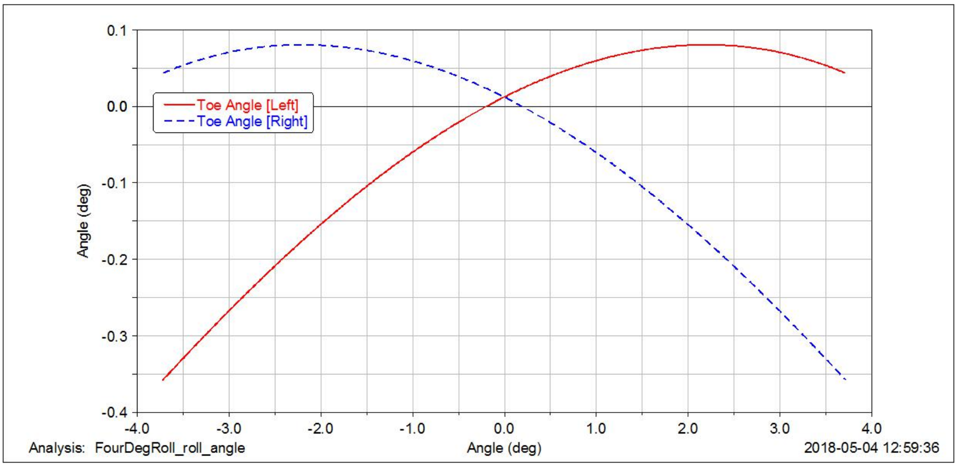


Figure B.8: Toe change vs chassis roll in ADAMS/Car

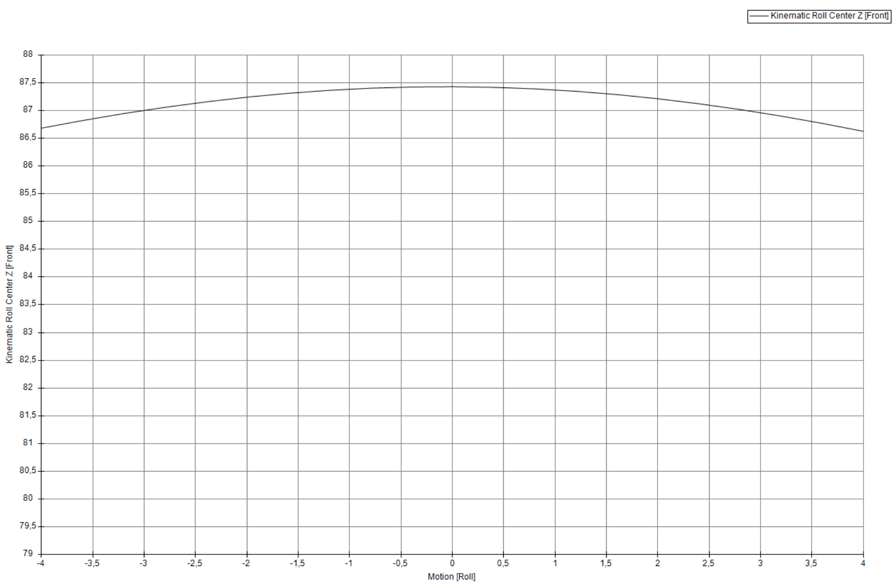


Figure B.9: Roll center vertical location vs chassis roll in OptimumK

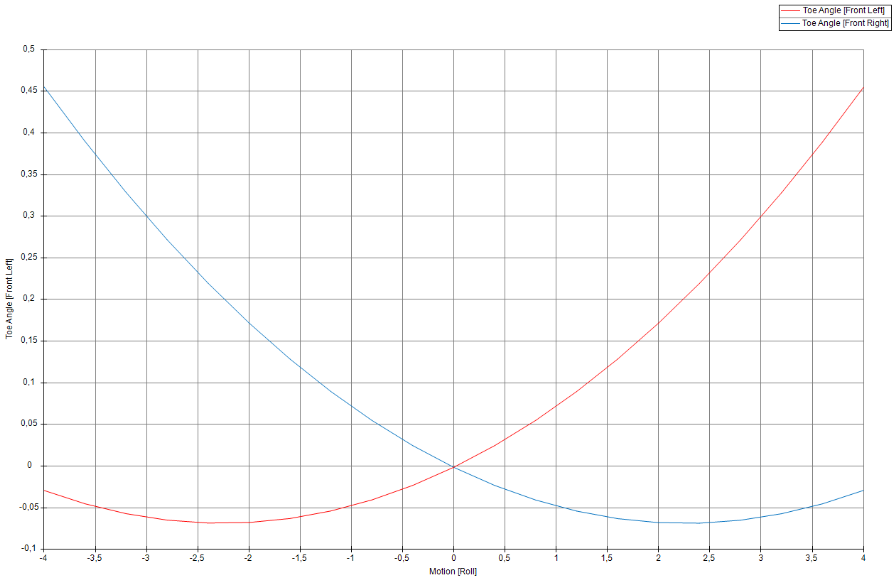


Figure B.10: Toe change vs chassis roll in OptimumK

B.3 Steering

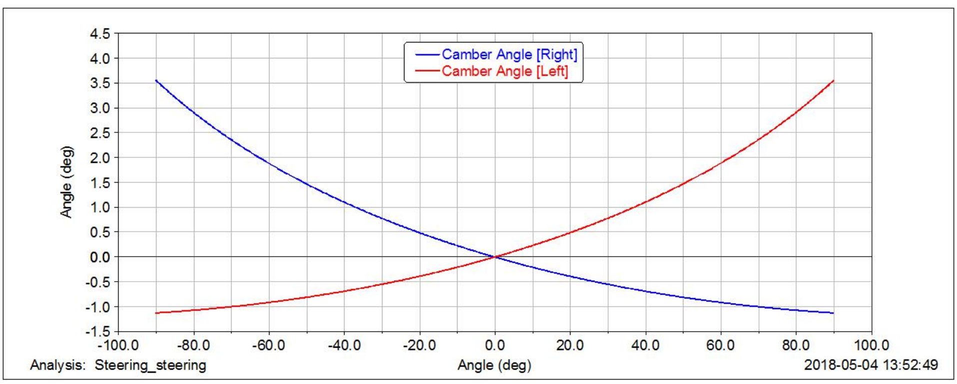


Figure B.11: Camber change vs steering angle in ADAMS/Car

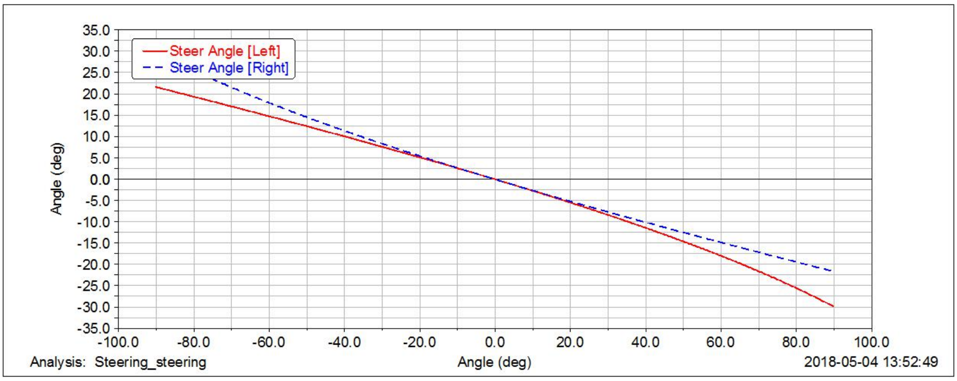


Figure B.12: Wheel angle vs steering wheel angle in ADAMS/Car

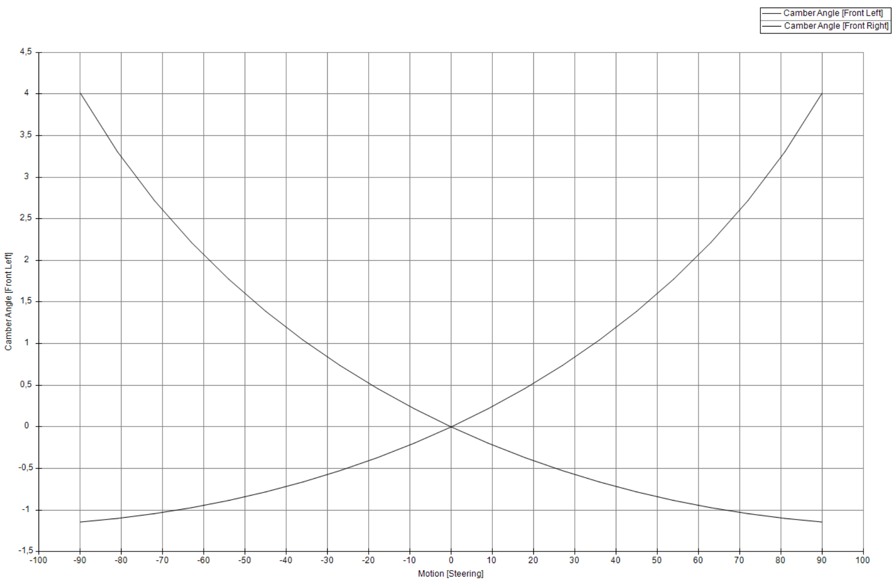


Figure B.13: Camber change vs steering angle in OptimumK

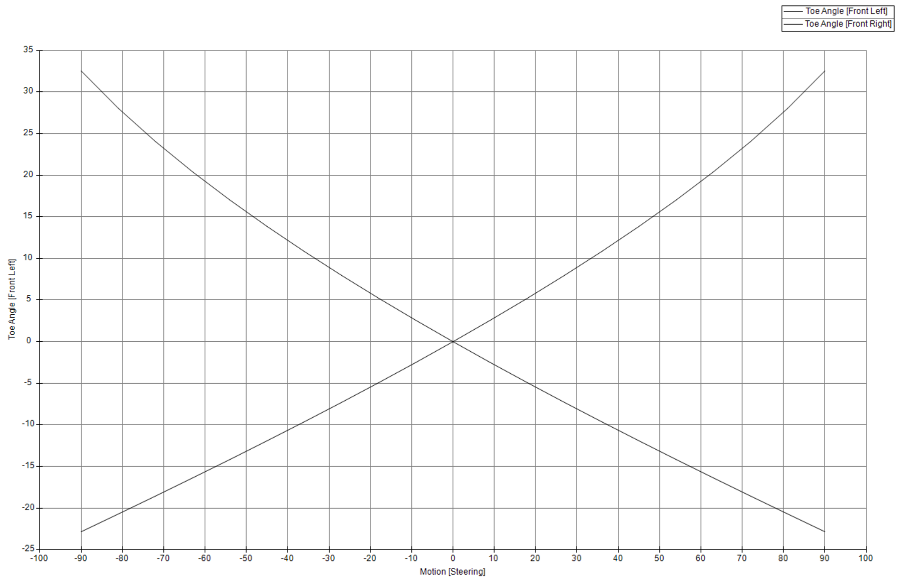


Figure B.14: Wheel angle vs steering wheel angle in OptimumK

B.4 Acceleration

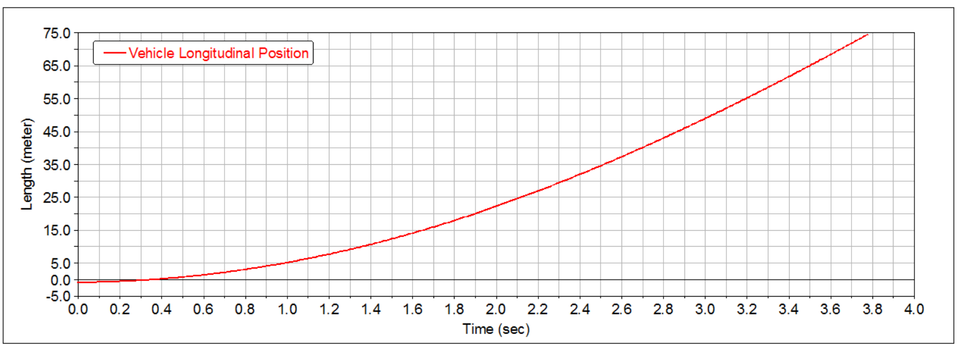


Figure B.15: Distance vs time in ADAMS/Car

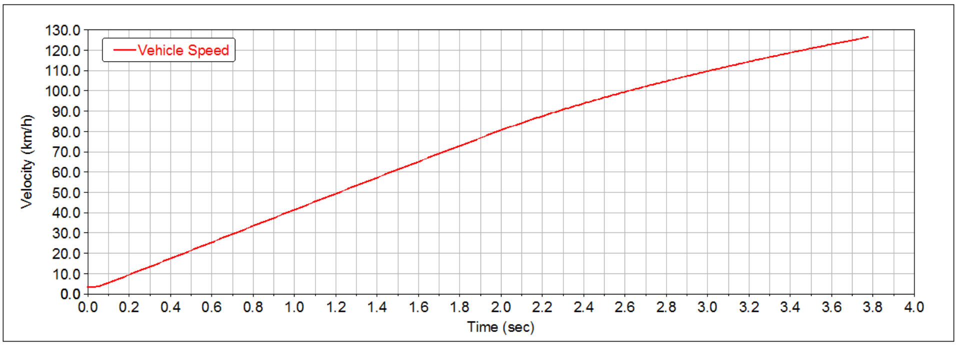


Figure B.16: Speed vs time in ADAMS/Car

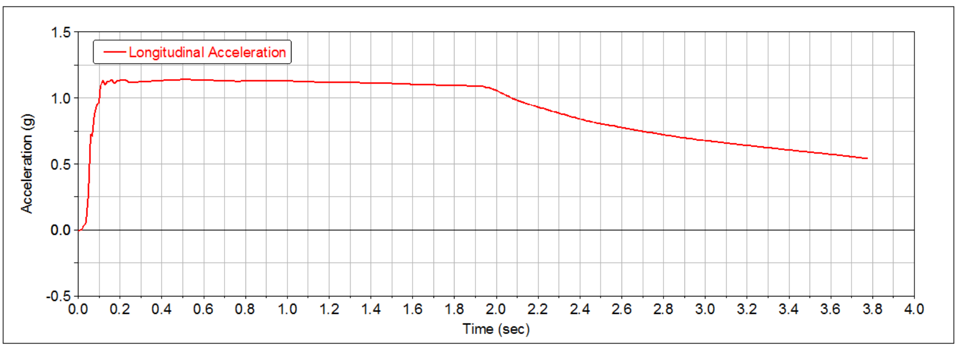


Figure B.17: Longitudinal acceleration vs time in ADAMS/Car

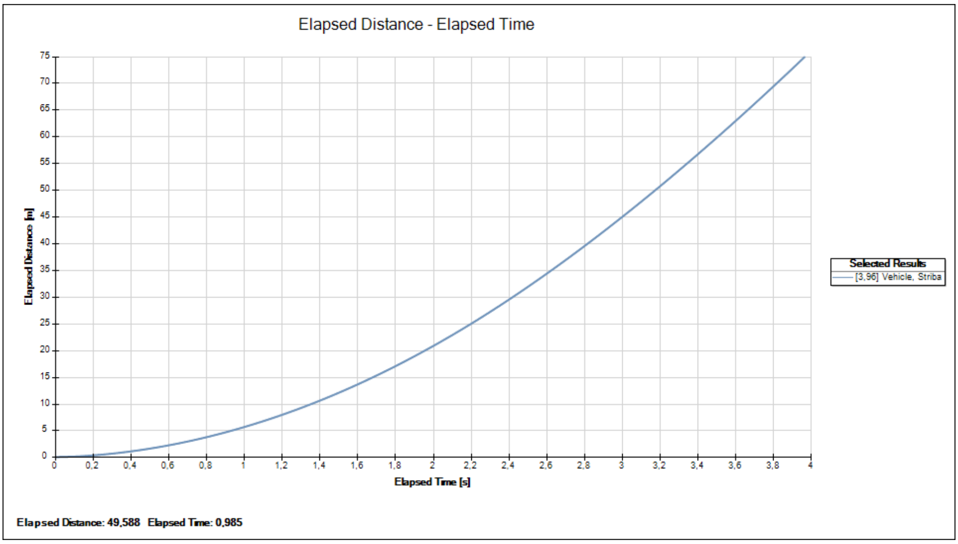


Figure B.18: Distance vs time in OptimumL

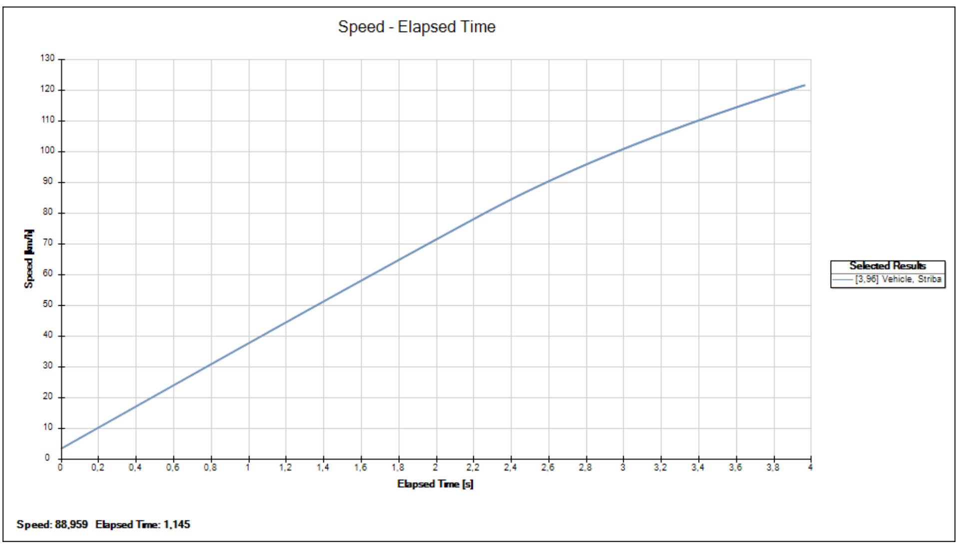


Figure B.19: Speed vs time in OptimumL

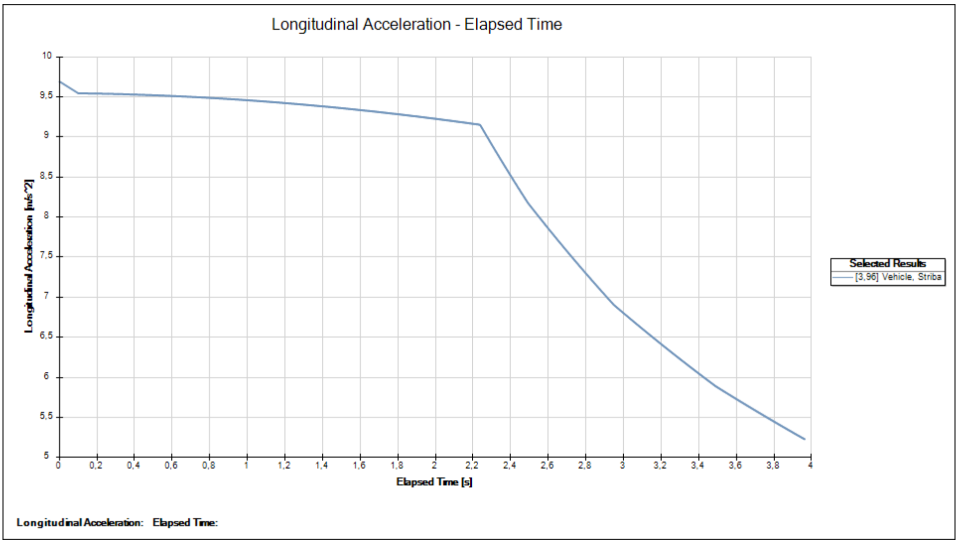


Figure B.20: Longitudinal acceleration vs time in OptimumL

Appendix C

MATLAB Acceleration Script

```
% Acceleration event simulation
% By Tor Inge Berge
clear all
g=9.81;             % Gravitational acceleration
D=75;               % Length of the Acceleration distance
m=280;              % Total mass of the car with driver
mdist=0.47;         % Static mass distribution on the front axle
mf=m*mdist;         % Static mass on the front axle
mr=m-mf;            % Static mass on the rear axle
mu=2.63;            % Coefficient of friction

%% Single point mass calculation with constant acceleration and no tire slippage
F1=mr*g*mu;              % Friction force on the rear wheels
t1=sqrt(2*D*m/F1);   % Time needed to complete the acceleration event

%% Single point mass calculation with varying acceleration and no tire slippage
EngTrq=[1 240
        3183 240
        3501 215
        4001 185
        4501 165
        5001 150];                          % Engine speed vs engine torque
GR=3.93;                                    % Gear ratio
```

```
WheeRad=0.26;                                   % Wheel radius
v2(1)=0;                                        % Initial Speed
s2(1)=0;                                        % Initial Position
s=0;
Fm2=EngTrq(1,2)*GR/WheeRad;                     % Motor
    acceleration force
Ft2=mr*g*mu;                                    % Tire friction
    force
tstep=0.01;                                      % Time step
i=1;                                            % Counter
while s < D ;                               % As long as the
    if Fm2 > Ft2;
        a2(i)=Ft2/m;
    else
        if (v2(i)>=0) && (v2(i)<EngTrq(2,1)*0.0069);
            a2(i)=EngTrq(1,2)*GR/WheeRad/m;
        elseif (v2(i)>=EngTrq(2,1)*0.0069) && (v2(i)<EngTrq
            (3,1)*0.0069);
            a2(i)=EngTrq(3,2)*GR/WheeRad/m;
        elseif (v2(i)>=EngTrq(3,1)*0.0069) && (v2(i)<EngTrq
            (4,1)*0.0069);
            a2(i)=EngTrq(4,2)*GR/WheeRad/m;
        elseif (v2(i)>=EngTrq(4,1)*0.0069) && (v2(i)<EngTrq
            (5,1)*0.0069);
            a2(i)=EngTrq(5,1)*GR/WheeRad/m;
        else (v2(i)>EngTrq(5,1)*0.0069);
            a2(i)=EngTrq(6,2)*GR/WheeRad/m;
        end
    end
    t2=i*tstep;
    time2(i)=tstep*i;
    s2(i+1)=v2(i)*tstep+0.5*a2(i)*tstep^2;
    if v2(i)>EngTrq(6,1)*0.0069;
        v2(i+1)=EngTrq(6,1)*0.0069;
    else
        v2(i+1)=v2(i)+a2(i)*tstep;
    end
    Fm2=a2(i)*m;
    s=s+s2(i+1);
    i=i+1;
end
%% Single point acceleration with load transfer and
    constant friction
WB=1.650;                 % Wheel Base
CGH=0.260;                % Center of gravity height
```

```
EngTrq=[1  240
        3183  240
        3501  215
        4001  185
        4501  165
        5001  150];                              % Engine speed vs engine torque
GR=3.93;                                         % Gear ratio
WheeRad=0.26;                                    % Wheel radius
v3(1)=0;                                         % Initial speed
s3(1)=0;                                         % Initial position
s=0;                                             % Initial distance covered
Fm3=EngTrq(1,2)*GR/WheeRad;                      % Motor acceleration force
Ft3=mr*g*mu;                                     % Tire friction force
mr3(1)=mr;                                       % Initial rear wheigt
tstep=0.01;                                       % Time step
i=1;                                             % Counter
while s < D ;                                    % As long as the have not covered 75m continiue the sim.
    if Fm3 > Ft3;                                % If the tractive force is larger then the tire friction.
        a3(i)=mr3(i)*g*mu/m;                     % Tire limited acceleration
    else
        if (v3(i)>=0) && (v3(i)<EngTrq(2,1)*0.0069);
            a3(i)=EngTrq(1,2)*GR/WheeRad/m;
        elseif (v3(i)>=EngTrq(2,1)*0.0069) && (v3(i)<EngTrq(3,1)*0.0069);
            a3(i)=EngTrq(3,2)*GR/WheeRad/m;
        elseif (v3(i)>=EngTrq(3,1)*0.0069) && (v3(i)<EngTrq(4,1)*0.0069);
            a3(i)=EngTrq(4,2)*GR/WheeRad/m;
        elseif (v3(i)>=EngTrq(4,1)*0.0069) && (v3(i)<EngTrq(5,1)*0.0069);
            a3(i)=EngTrq(5,1)*GR/WheeRad/m;
        else (v3(i)>EngTrq(5,1)*0.0069);
            a3(i)=EngTrq(6,2)*GR/WheeRad/m;
        end
    end
    t3=i*tstep;
    time3(i)=i*tstep;
```

```
    s3(i+1)=v3(i)*tstep+0.5*a3(i)*tstep^2;
    if v3(i)>EngTrq(6,1)*0.0069;
        v3(i+1)=EngTrq(6,1)*0.0069;
    else
        v3(i+1)=v3(i)+a3(i)*tstep;
    end
    LonLT(i+1)=a3(i)*m*CGH/WB/g;
    mr3(i+1)=LonLT(i+1)+mr;
    Fm3=a3(i)*m;
    s=s+s3(i+1);
    i=i+1;
end

%% Acceleration with air and road resistance (Ground
    Vehicle Dynamics)
% This simulation takes into account rolling and air
    resistance while
% accelerating. Taken from Ground Vehicle Dynamics (1993),
    Popp, K. and
% Schiehlen, W. p.
Ia=0.39;          % Mass moment of inertia of axle
Ig=0;             % Mass moment of inertia of gearbox
Im=0.01;          % Mass moment of inertia of the motor
id=1;             % Final drive/diffratio
GR=3.93;          % Gear ratio
WheeRad=0.26;     % Wheel radius
Mm=[240 215 185 165 150 130];              % Motor moment
wm=2*pi/60*[3183 3501 4001 4501 5001 5501];      % Motor
    angular velocity
nu=0.95;          % Gear efficiency
B=0;              % Breaking force
alpha=0;          % Inclination angle of the slope
A=0.79931;        % Frontal area of the car
rho=1.2;          % Air density
Cw=0.3;           % Air resistance coefficient
phir=0.02;        % Rolling resitance of the wheels
S4=0;             % Initial position
v4(1)=0;          % Initial velocity
t4=0;             % Initial time
i=1;              % Initial Step
time4(1)=0;
while S4(i) < D
    if i==1
        v4(i)=0*WheeRad/GR;
        Mm4(i)=240;
```

```
    else
        if v4(i)<=wm(1)*WheeRad/GR
            Mm4(i)=Mm(1);
        elseif v4(i)<=wm(2)*WheeRad/GR
            Mm4(i)=Mm(2);
        elseif v4(i-1)<=wm(3)*WheeRad/GR
            Mm4(i)=Mm(3);
        elseif v4(i-1)<=wm(4)*WheeRad/GR
            Mm4(i)=Mm(4);
        else
            Mm4(i)=Mm(5);
        end
    end
    Wl(i)=0.5*Cw*A*rho*v4(i)^2;
    Wr(i)=phir*m*g;
    Af(i)=1/WheeRad*nu*GR*Mm4(i);
    Mi(i)=m+(Ia*2)/WheeRad^2+id^2*Ig/WheeRad^2+GR^2*Im/
        WheeRad^2;
    a4(i+1)=1/Mi(i)*(Af(i)-Wl(i)-Wr(i));
    v4(i+1)=v4(i)+a4(i)*tstep;
    if v4(i+1)>wm(5)/WheeRad*GR
        v4(i+1)=wm(5)/WheeRad*GR;
    end
    s4(i+1)=v4(i)*tstep+0.5*a4(i)*tstep^2;
    S4(i+1)=S4(i)+s4(i+1);
    t4=t4+tstep;
    time4(i+1)=time4(i)+tstep;
    i=i+1;
end
```

Appendix D

Roll Stiffness Calculation

```
% Roll stiffness calculation
% Based on David Gould's example in Competition Car Suspension
clear all
% Inputs
W=2800.74;          % Total weight of the car in N
WF=1291.41;         % Weight on the front axle
WR=1509.33;         % Weight on the rear axle
UWF=159.41;         % Unsprung weight on the front axle
UWR=182.66;         % Unsprung weight on the rear axle
UGF=260.35;         % Mass center of the unsprung mass on the front axle
UGR=260.35;         % Mass center of the unsprung mass on the rear axle
TF=1200;            % Track at the front axle
TR=1200;            % Track at the rear axle
RCFH=87;            % Roll center height front
RCRH=72;            % Roll center height rear
SWF=1132;           % Sprung mass on the front axle
SWR=1326.67;        % Sprung mass on the rear axle
SW=2462.31;         % Total sprung mass
GM=248;             % Height of center of mass
Garb=79300;         % Shear modulus of ARB metal
OD=15.8;            % ARB outer diameter
ID=14.2;            % ARB inner diameter
ARBarmF=106;        % ARB arm length front
ARBarmR=156;        % ARB arm length rear
ARBFL=460;          % ARB rod length front
ARBRL=530;          % ARB rod length rear
```

```
% Unsprung weight transfer
UtF=UWF*UGF/TF;  % Weight transfer of unsprung mass on the front axle
UtR=UWR*UGR/TR;  % Weight transfer of unsprung mass on the rear axle

% Weight transfer via roll center
CtF=SWF*RCFH/TF; % Weight transfer of sprung mass on the front axle
CtR=SWR*RCRH/TR; % Weight transfer of sprung mass on the rear axle

% Weight transfer via the sprung mass
WDR=SWR/SW;                  % Weight distribution rear
TM=((TR-TF)*WDR)+TF;         % Average track width
CM=((RCRH-RCFH)*WDR)+RCFH;   % Average roll center height
LM=GM-CM;                    % Distance between CM and GM
St=SW*LM/TM;                 % Total sprung weight transfer

% Total weight transfer
Wt=UtF+UtR+CtF+CtR+St;       % Total weight transfer

% Springs
SF=61.3;    % Spring stiffness front
SR=61.3;    % Spring stiffness rear
MrF=2;      % Spring motion ratio front
MrR=1.5;    % Spring motion ratio rear
ArF=(SF/MrF^2)*TF^2/2*pi/180/1000;  % Anti-roll in front springs
ArR=(SR/MrR^2)*TR^2/2*pi/180/1000;  % Anti-roll in rear springs

% Roll Bars
MrArbF=3.638;   % ARB motio ratio front
MrArbR=1.6;     % ARB motio ratio rear
AngRF=(pi*Garb*(OD^4-ID^4))/(32*ARBFL*180*1000); % Torsional stiffness front
AngRR=(pi*Garb*(OD^4-ID^4))/(32*ARBRL*180*1000); % Torsional stiffness rear
BF=AngRF/(ARBarmF^2*pi/180);     % ARB stiffness front
BR=AngRR/(ARBarmR^2*pi/180);     % ARB stiffness rear
BrF=BF*(1/MrArbF)^2*TF^2*pi/180;    % Anti-roll in ARB front
BrR=BR*(1/MrArbR)^2*TR^2*pi/180;    % Anti-roll in ARB rear
```

```
% Distribution
Fr=ArF+BrF      % Total front roll stiffness
Rr=ArR+BrR      % Total rear roll stiffness
```

Appendix E

OptimumKinematics Suspension Coordinates

Table E.1: Coordinates used in OptimumK to define the front wishbones.

	Point Name	Left			Right		
		X	Y	Z	X	Y	Z
Double A-Arm	CHAS_LowFor	155,0	200,0	140,0	155,0	200,0	140,0
	CHAS_LowAft	139,0	200,0	140,0	139,0	200,0	140,0
	CHAS_UppFor	139,0	250,0	275,0	139,0	250,0	275,0
	CHAS_UppAft	155,0	250,0	275,0	155,0	250,0	275,0
	UPRI_LowPnt	9,8	549,0	150,0	9,8	549,0	150,0
	UPRI_UppPnt	8,0	516,0	350,0	8,0	516,0	350,0
	CHAS_TiePnt	38,0	200,0	162,0	38,0	200,0	162,0
	UPRI_TiePnt	52,0	573,0	193,0	52,0	573,0	193,0

Table E.2: Coordinates used in OptimumK to define the front pull-rod and damper system.

	Point Name	Left			Right		
		X	Y	Z	X	Y	Z
Push Pull	NSMA_PPAttPnt_L	8,0	502,0	330,0	8,0	502,0	330,0
	CHAS_AttPnt_L	8,0	247,0	295,0	8,0	247,0	295,0
	CHAS_RocAxi_L	91,0	196,0	112,0	91,0	196,0	112,0
	CHAS_RocPiv_L	8,0	196,0	112,0	8,0	196,0	112,0
	ROCK_RodPnt_L	8,0	251,0	47,0	8,0	251,0	47,0
	ROCK_CoiPnt_L	8,0	276,0	122,0	8,0	276,0	122,0

Table E.3: Coordinates used in OptimumK to define the U-bar and setup parameters for the wheels and steering.

	Point Name	Left			Right		
		X	Y	Z	X	Y	Z
U-Bar	NSMA_UBarAttPnt_L	8,0	236,0	117,0	8,0	236,0	117,0
	UBAR_AttPnt_L	9,0	230,0	197,0	9,0	230,0	197,0
	CHAS_PivPnt_L	86,0	230,0	156,0	86,0	230,0	156,0
Rack Pinion	Steering Ratio	80					

Point Name	Left	Right
Half Track	600,0	600,0
Longitudinal Offset	0,0	0,0
Lateral Offset	0,0	0,0
Vertical Offset	0,0	0,0
Static Camber	0,0	0,0
Static Toe	0,0	0,0
Rim Diameter	350,0	350,0
Tire Diameter	520,7	520,7
Tire Width	177,8	177,8

Appendix F

SPMM Results

SPMM TEST REPORT

======================

VEHICLE ASSEMBLY: Date : 12 Jun 2018 15:43

<ION18>/assemblies,tbl/ION18_FullVehicle,asy

(PARAMETER)	(UNITS)	(TOTAL)	(LEFT)	(RIGHT)
Total weight	N	2800,74	1400,37	1400,37
	%		50	50
Front ground reaction	N [%]	1291,41	645,7	645,7
Rear ground reaction	N [%]	1509,33	754,67	754,67
Total roll inertia	kg-mm**2	1,32E+07		
Total pitch inertia	kg-mm**2	3,57E+07		
Total yaw inertia	kg-mm**2	4,78E+07		
Total product Ixy	kg-mm**2	5,49E-05		
Total product Ixz	kg-mm**2	1,56E+06		
Total product Iyz	kg-mm**2	-2,69E-05		
Global C,G location (X/Y/Z)	mm	890,26	0	248,72
Ground plane elevation	mm	0		
C,G, height	mm	248,72		
Body yaw angle	deg	0		
Body pitch angle	deg	-0,04		
Body roll angle	deg	0		
Wheelbase	mm	1650,37	1650,37	1650,37
Average track width	mm	0,49		
Static stability factor	-	0,001		

FRONT AXLE:
<ION18>/subsystems,tbl/ION18_FrontSusp,sub
<ION18>/tires,tbl/fsae_tire_front,tir
<ION18>/tires,tbl/fsae_tire_front,tir

(PARAMETER)	(UNITS)	(AVERAGE)	(LEFT)	(RIGHT)
STATIC:				
Static toe	deg	-0,11	-0,11	-0,11
Static camber	deg	-1,02	-1,02	-1,02
Static caster	deg	-5,03	-5,03	-5,03
Kingpin angle	deg	-8,36	-8,36	-8,36
Caster trail	mm	-22,56	-22,56	-22,56
Scrub radius	mm	-27,12	-27,12	-27,12
Track width	mm	-1195,02		
Roll center height	mm	104,95		
Roll center lateral shift	mm	0		
Wheel center rise	mm	-15,88	-15,88	-15,88
Tire loaded radius	mm	254,26	254,26	254,26
BOUNCE TEST:				
Wheel rate	N/mm	13,65	13,65	13,65
Spring rate	N/mm	61,3	61,3	61,3
Spring ratio	mm/mm	0,48	0,48	0,48
Damper ratio	mm/mm	0,48	0,48	0,48
Spring rate at wheel	N/mm	13,83	13,83	13,83
Bump steer	deg/mm	0,00974	0,00974	0,00974
Bump caster	deg/mm	-0,00081	-0,00081	-0,00081
Bump camber	deg/mm	0,06314	0,06314	0,06314
Longitudinal wheel center recession	mm/mm	-0,01	-0,01	-0,01
Lateral wheel center migration	mm/mm	-0,11	-0,11	-0,11
ROLL TEST:				
Wheel rate in roll	N/mm	17,2	17,2	17,2
Roll stiffness	Nmm/deg	2,14E+05		
Roll steer (per degree)	deg/deg	0,0875	0,0875	0,0875
Roll steer	deg/mm	0	0,00965	-0,00965
Roll camber w,r,t, chassis (per degree)	deg/deg	-0,56762	-0,56762	-0,56762
Roll camber (per degree)	deg/deg	0,43245	0,43245	0,43245
Roll camber	deg/mm	0	0,0634	-0,0634
Steering wheel torque vs roll angle	Nmm/deg	434,95697		

STEER TEST:

Rack travel @ steering input	mm	18,86		
Max steer angle @steering input (left turn)	deg	-25,51	-21,76	-29,27
Max steer angle @steering input (right turn)	deg	25,51	29,27	21,76
Full lock angle @pvs_max_rack_disp (left turn)	deg	N/A	N/A	N/A
Full lock angle @pvs_max_rack_disp (right turn)	deg	N/A	N/A	N/A

LONGITUDINAL COMPLIANCE TEST:

Brake Steer	deg/kN	N/A	N/A	N/A
Traction Steer	deg/kN	N/A	N/A	N/A
Contact patch braking stiffness	N/mm	N/A	N/A	N/A
Wheel center longitudinal stiffness	N/mm	N/A	N/A	N/A
Caster compliance - braking	deg/kN	N/A	N/A	N/A
Caster compliance - traction	deg/kN	N/A	N/A	N/A

LATERAL COMPLIANCE TEST - PARALLEL:

Lateral compliance steer	deg/kN	N/A	N/A	N/A
Lateral compliance camber	deg/kN	N/A	N/A	N/A
Wheel center lateral stiffness	N/mm	N/A	N/A	N/A

LATERAL COMPLIANCE TEST - OPPOSITE:

Lateral compliance steer	deg/kN	N/A	N/A	N/A
Lateral compliance camber	deg/kN	N/A	N/A	N/A
Wheel center lateral stiffness	N/mm	N/A	N/A	N/A

ALIGNING TORQUE TEST - PARALLEL:

Aligning torque - steer	deg/kNmm	N/A	N/A	N/A
Aligning torque - camber	deg/kNmm	N/A	N/A	N/A

ALIGNING TORQUE TEST - OPPOSITE:

Aligning torque - steer	deg/kNmm	N/A	N/A	N/A
Aligning torque - camber	deg/kNmm	N/A	N/A	N/A

REAR AXLE:
<ION18>/subsystems,tbl/ION18_RearSusp,sub
<ION18>/tires,tbl/fsae_tire_rear,tir
<ION18>/tires,tbl/fsae_tire_rear,tir

(PARAMETER)	(UNITS)	(AVERAGE)	(LEFT)	(RIGHT)
STATIC:				
Static toe	deg	-0,08	-0,08	-0,08
Static camber	deg	0,97	0,97	0,97
Static caster	deg	0,11	0,11	0,11
Kingpin angle	deg	6,94	6,94	6,94
Track width	mm	1196		
Roll center height	mm	86,06		
Roll center lateral shift	mm	0		
Wheel center rise	mm	-15,59	-15,59	-15,59
Tire loaded radius	mm	253,29	253,29	253,29
BOUNCE TEST:				
Wheel rate	N/mm	27,36	27,36	27,36
Spring rate	N/mm	61,3	61,3	61,3
Spring ratio	mm/mm	0,65	0,65	0,65
Damper ratio	mm/mm	0,65	0,65	0,65
Spring rate at wheel	N/mm	25,68	25,68	25,68
Bump steer	deg/mm	0,00667	0,00667	0,00667
Bump caster	deg/mm	-0,0043	-0,0043	-0,0043
Bump camber	deg/mm	-0,05991	-0,05991	-0,05991
Longitudinal wheel center recession	mm/mm	0,02	0,02	0,02
Lateral wheel center migration	mm/mm	0,12	0,12	0,12
ROLL TEST:				
Wheel rate in roll	N/mm	27,65	27,65	27,65
Roll stiffness	Nmm/deg	3,45E+05		
Roll steer (per degree)	deg/deg	-0,05624	-0,05624	-0,05624
Roll steer	deg/mm	0	0,00661	-0,00661
Roll camber w,r,t, chassis (per degree)	deg/deg	-0,50417	-0,50417	-0,50417
Roll camber (per degree)	deg/deg	0,49578	0,49578	0,49578
Roll camber	deg/mm	0	-0,06036	0,06036

LONGITUDINAL COMPLIANCE TEST:				
Brake steer	deg/kN	N/A	N/A	N/A
Traction steer	deg/kN	N/A	N/A	N/A
Contact patch braking stiffness	N/mm	N/A	N/A	N/A
Wheel center longitudinal stiffness	N/mm	N/A	N/A	N/A
Caster compliance - braking	deg/kN	N/A	N/A	N/A
Caster compliance - traction	deg/kN	N/A	N/A	N/A
LATERAL COMPLIANCE TEST - PARALLEL:				
Lateral compliance steer	deg/kN	N/A	N/A	N/A
Lateral compliance camber	deg/kN	N/A	N/A	N/A
Wheel center lateral stiffness	N/mm	N/A	N/A	N/A
LATERAL COMPLIANCE TEST - OPPOSITE:				
Lateral compliance steer	deg/kN	N/A	N/A	N/A
Lateral compliance camber	deg/kN	N/A	N/A	N/A
Wheel center lateral stiffness	N/mm	N/A	N/A	N/A
ALIGNING TORQUE TEST - PARALLEL:				
Aligning torque - steer	deg/kNmm	N/A	N/A	N/A
Aligning torque - camber	deg/kNmm	N/A	N/A	N/A
ALIGNING TORQUE TEST - OPPOSITE:				
Aligning torque - steer	deg/kNmm	N/A	N/A	N/A
Aligning torque - camber	deg/kNmm	N/A	N/A	N/A

SPMM REPORT SIGN CONVENTIONS
==============================

	CONVENTION FOR POSITIVE VALUE
Toe change	Wheel toe in when moved into jounce
Caster change	Caster increases when wheel moves into jounce
Camber change	Top of tire moves outboard when wheel moves into jounce
Wheel center rise	Wheel travelling into jounce
Lat, force deflection	Wheel moves to the right when pushed inboard
Lat, force steer	Wheel toe in when pushed inboard
Lat, force camber	Top of tire moves out when pushed inboard
Brake steer	Toe in due to a rearward force at the tire contact patch
Recession steer	Toe in due to a rearward force at the wheel center
Tractive force steer	Toe in due to a forward force at the wheel center
Aligning torque steer	Toe in due to an aligning torque at wheel center

Appendix G

Pre-Study Report

Universitetet i Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

PRESTUDY REPORT FOR BACHELOR THESIS

Field of study/specialising: *Mechanical engineering/Engineering Structures and Materials*	Spring semester, 2018 Open / ~~Restricted access~~
Authors: *Tor Inge Berge*	…………………………………………………

Faculty supervisor:

Hirpa G. Lemu, UiS.

Thesis title:

Performance analysis and Multi-body dynamic simulation of Formula Student car suspension

Credits:

20 points. Essay about scientific theory/ethics is included.

Key words: - Suspension - ADAMS - Formula Student - ION Racing - Car - Autodesk Inventor - Multi-body simulation - Lagrangian - Damping	Pages: ……………. Stavanger, ………………….

Abstract

This is a pre-study report for my master thesis at the University of Stavanger. The thesis is written in the period 1th of February to 15th of June 2018. I will use this thesis to learn how to use ADAMS Car for set-up and simulation, and develop tools for next years team to use and further develop. The title of the thesis is *Performance analysis and multibody dynamic simulation of Formula Student car suspension system.* This pre-study report will define what problems I am going to analyze in the thesis, some expected problems and a time table on how I plan to work up until the deadline in June.

Contents

1 Introduction

This pre-study report will give a brief summary of the content of the master thesis "Performance analysis and multi-body dynamic simulation of Formula Student car suspension system". It is written in conjunction with the design and manufacturing of ION Racings 7th car. This pre-study report will go into the goals for the master thesis and possible additional goals if time allows it.

1.1 Background

ION Racing have competed in Formula Student UK on Silverstone Race Track in England 6 years in a row. In the past ION Racing have built 2 combustion and 4 electric cars. For each year the design have evolved and improvements have been made. The first two cars used tubular steel frames with the same motorcycle engine. For the third car it was decided to make the move over to carbon fibre monocoque and electric drive train. This have been the two main design choices through the next 4 cars. The only mayor change is going from two to one electric motors. This choice was mainly done to reduce weight and reduce the complexity of the control systems for the car. This years vehicle will build on much of the experience gained from last years car, Fenrir, which have been the most successful car in ION Racings history. This was done by focusing on reliability in the design and not so much on the weight of the car. There was also more focus on simulation and having good documentation for the design choices. This makes it much easier for this years team to get up to speed why choices were made and what to focus on this year.

In the previous years for ION Racing, the design process for the suspension and damping systems have been done mostly with hand calculations and excel spreadsheets with basic steady state calculations. This is good for rough estimates of weight transfer and loads in the suspension, but do not take into account movement in the suspension and roll in the chassis. To get a better understanding of the behaviour of the car, ION Racing invested in a license for the OptimumG software package in the autumn 2017. OptimumG is a software package containing OptimumTire, OptimumKinematics and OptimumDynamics. These were used in the design phase of the suspension system and analysis of the chassis movement for ION Racings 2018 car. The OptimumG software are easy to set up and give useful results in a short time, which makes it valuable in the short design phase of a formula student car. The problem is that the Dynamic part of the package do not have the ability to evaluate subsystems, change dampers or go more in-depth on each part. Because of these issues it was desired to look at alternatives for more advanced simulations when this is necessary. ADAMS Car have been used for ION Racings 2014 car, and is available at the university. It is also one of the most used multi-body simulation software, used by major car companies like VW and Audi. It is also one of the first software developed for multi-body simulation back in 1977. In ADAMS Car there are two profiles available for the user. This is the *Expert* and *Normal* user. The *Expert* user profile is designed for the expert in a user group. This profile gives the user access to the Template Builder which can create templates from the bottom for specific systems. The *Normal* user can only change the position of hard points in templates and change values for parts, like springs, dampers and engine.

1.2 Thesis definitions and limitations

The master thesis will explore the use of ADAMS Car to simulate suspension and handling of the formula student car designed and built by ION Racing UiS for the 2018 season. It will give a basic introduction of the theory behind multi-body dynamics and some history of its uses. The main purpose is to develop a template and run simulations of the car built by ION Racing UiS for this years Formula Student competition. The templates are developed to make it easier for the next years team to start with ADAMS Car and to run simulations. The simulation cases will be based on the dynamic events in the competition. This is done in part to be able to predict how fast the car is compared to other cars, and to be able to compare the simulated results with actual testing. As the first competitions starts in mid July, and the Master is due in mid June, the comparison of the simulated and actual tested cases might not be a part of the thesis, depending on how early the car is ready to drive. In addition to ADAMS Car the OptimumG software packages will be used to compare results and do some simulations which will be faster to do on the OptimumG software. This will also show if it is necessary to spend time to develop advanced simulations in ADAMS for next years team, or if it would be just as well spent time working with OptimumG.

2 Definition of problems

ADAMS in it self has a steep learning curve. Most of the user interface(UI) is outdated and mostly text based as shown in figure 1. This might turn away some users which is more used to newer, a more streamlined UI. ADAMS uses a database system which catalogues templates, subsystems and assemblies into folders containing text files. These text files contain all the data for each template and it is possible to edit them with any text editing software, like Notepad. All data on dynamic parts are also stored in folders and text files. To edit properties of a spring, damper, gearbox or engine, it is possible to just edit the values in the text file. When starting a new design of a suspension system, a template has to be used. This can be done using the templates supplied by MSC, or by generating new ones with the *Template Builder*. Building the templates is one of the first problems that has to be solved. Next is to create the subsystems from the templates and combining these into assemblies. When a front or rear suspension assembly have been made, it is possible to run simulations on them. These are mostly used to study the motions, reactions and loads in the suspension under various applied loads and motions. If a full assembly of a vehicle have been made, with suspension, drive train, brakes and wheels, it is possible to start with more advanced simulations. These can be turns, accelerations and braking events, or they can be combined into more advanced maneuvers. Finally when the simulations are done, the results can be analyzed and compared. When working with the software it is preferable to start with a working template, here based on FSAE2012 template supplied by MSC, and run test simulations on this, just to try out the simulation step. Then start modifying some of the parts in the assembly, and run the simulations agein. This will help narrowing in on systems which may cause problems. If a simulation works, and one subsystem is modified, and the next simulation do not work, it is likely the modified subsystem is the cause. This is the procedure chosen to learn the software and solve problems during this thesis.

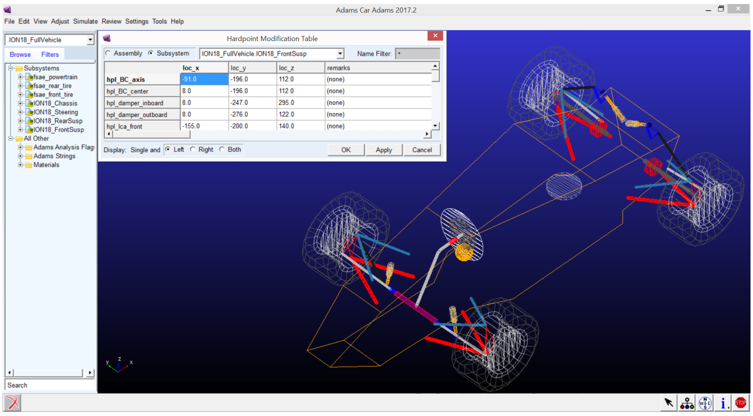


Figure 1: The user interface in ADAMS Car showing table to modify hard point positions and the full vehicle assembly.

2.1 Templates

Templates in ADAMS Car is the basis for all subsystems. The templates defines what parts a subsystems contains, how they are connected and at which point they connect and interact with other subsystems. To be able to create templates, the software has to be set to *expert* mode. This lets the user access the *Template Builder*. Here the user is able to define a template from scratch. This is done by defining *Hard points* which are used as construction nodes and connection points between parts. Geometry is created between the hard points to create parts and joints are defined at nodes connecting parts. In this process the problem is to define the joints in the right way and create the necessary communication points between templates to let them exchange data. At the moment most of these problems has been solved and templates for both front and rear suspension with steering and anti-roll systems are done, with the exception of a communications error between the steering and suspension. There seem to be missing a link there which results in steering input on the steering wheel not giving any movement in the wheels. When this problem is solved, a template for the power train has to be created. This power train has to have the electric motor power curve with a gear ratio and some differential data. Some of the data might be possible to reverse engineer from other templates, while some of it might need to be made from scratch.

2.2 Assemblies

Assemblies and subsystems are available in the *Normal* user mode in ADAMS Car. Subsystems are created from templates. When working with a subsystem it is possible to move the hard points from the template, and modify the force elements, like dampers or springs. It is not possible to change joints or part relations. Subsystems for the anti-roll bar, steering and suspension are combined into assemblies to create the front suspension, while for the rear the suspension, anti-roll bar and power train is combined. These assemblies can be analyzed individually for suspension dynamics or kinematics. This is done with simulated suspension rigs with pistons moving the wheels and chassis. If more advanced simulations are required, the suspension systems can be combined into a full vehicle model. In a full vehicle model assembly a chassis is also added. As the assemblies are mostly just putting together the templates, there are not many problems expected to be directly related to the assemblies.

2.3 Simulation

Simulations are run on the assemblies. These can range from simple simulated test rigs to 3D roads where a simulated driver drives the car through a course. For simple suspension evaluations some suspension assembly motions will be simulated to compare the results with the OptimumG package results. The event simulations that will be run in this thesis, are run on the full vehicle assembly. The Acceleration event can be run with one of the standard simulations, but the skid pad might need to be custom made. If time allows it, and the simulation generator is not to complicated to work with, a Autocross track might be generated. The simulations are expected to be the hardest part of the work. Not directly because simulations are difficult, but it highlight problems that need to be addressed in the templates and assemblies. This will lead to much back and forth between each step to generate a reliable simulation result.

2.4 Results

With the simulation done, the results have to be analyzed. ADAMS Car got a post-processor which can visualize the results by animation and in plots or tables. The most important results needed from the simulations is the time it takes to complete the event. This is used to compare the time with results from previous competitions to see where the simulated car might have been placed in the event. The placement in each event will give an amount of points based on the placement in the event. This can be combined to predict and overall placement in the competition. But results showing forces in parts, motion of the suspension, or effects of oscillations in the suspension is also of interest for further study and understanding.

3 History

The beginning of multi-body dynamics starts with the formulations outlined by Joseph-Louis Lagrange in his work *Mcanique Analytique* [1]. These formulations have remained with no significant changes up to this date. There have only been some additions to take into account friction, by Coulomb, beam theory, by Euler, and lubrication by Reynolds. These formulations, together with the increasing computing power available, have made it possible to solve quite complex systems, like multi cylinder engines or suspensions. The first registered study of motion of a vehicle in response to steering input, was done by Segel [2] in 1956. His study was done on a Buick on a flat road and was able to predict with some accuracy the motion of the vehicle. This investigation sparked an interest by the vehicle manufactures in multi-body dynamics. In the mid 1960's Raymond R. McHenry developed the HVOSM mathematical model and computer simulation to predict how a vehicle would respond to irregularities in the road [3]. This simulation was used for developing the Astro-Spiral Jump used in the James Bond movie *Man with the Golden Gun*, as shown in figure 2. The stunt was first simulated by trail and error to figure out what speed and ramp layout was needed for the stunt to work. In 1977 N. Orlandea presented a practical solution methodology for large rigid multi-body systems[4]. This methodology based on the Lagrangian dynamics and their work led to the development of ADAMS. ADAMS is a acronym for Automatic Dynamic Analysis of Mechanical Systems. ADAMS was adapted by many of the major car manufacturers, like Audi, VW, BMW, Volvo and Renault [5]. This adaptation have generated many standard simulation cases directly towards vehicle handling and ride situations. To run simulations on a vehicle in a lane change on the highway in ADAMS, the user can choose the standard lane change simulation and specify some variables, like speed and time of the event. In recent years many multi-body dynamics simulation software have been developed. Most companies focusing on engineering simulation have a multi-body dynamics software in their catalogue.

Figure 2: The Astro Spiral Jump as shown in *The Man with the Golden Gun*.

4 Timetable and Gantt Chart

Predicting the time needed for each task is difficult, but an estimation will be presented here. Milestones will be set to have goals to work for through the whole period. This will also make it easier to allocate time for each of the tasks.

4.1 Timetable

The timetable sets the milestones from 1st of February till the deadline in June. Some work as already been done, like most of the templates are completed, but there are still some adjustments to be done. Milestones:

- 2nd March: Full working assembly with power train and tires
- 23th March: Simulation Tracks modelled
- 2st April: First working event simulation
- 11th May: Simulation results from all events
- 25th May: Results analyzed
- 15th June: Deadline for hand-in

4.2 Gantt chart

The Gantt chart in table 1 show the expected time allocation. There are some uncertainties in the expected time table, as some problems might appear that need to be solved.

	Week																		
Task	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Template finishing																			
Assembly																			
Simulation events setup																			
Simulation																			
Results analysed																			
Finishing Thesis																			

Table 1: Gantt chart of the time allocated for each task. Green are working weeks, Red is milestones for each task.

References

[1] Rahnejat, H., "Multi-body dynamics: Historical evolution and application," *Proceedings of the Institution of Mechanical Engineers*, vol. Part C:Journal of Mechanical Engineering Science, no. 214 (1), pp. 149–173, 2000.

[2] L. Segel, "An overview of developments in road-vehicle dynamics: Past, present and future," *Vehicle Ride and Handling International Conference*, pp. 1–12, 1993.

[3] B. G. "McHenry, "A short history of nearly everything!(...about mchenry and computers in highway safety," *Collision Magazine*, vol. 6, no. 2, 2012.

[4] N. Orlandea, M. A. Chance, and C. D. A., "A sparsly-oriented approach to the dynamic analysis and design of mechanical systems," *Journal of Engineering for Industry*, no. 99, 1977.

[5] M. Blundell and D. Harty, *The Multibody Systems Approach to Vehicle Dynamics.* Elsevier Science, 2004.