

University of Wales Trinity Saint David

School of Engineering

Developing and Comparing a range of Simulation Methods to Predict the Handling Balance of the MCR Race Car

Major Project Assessment

(M6X00417)

Cornelius May

1701329

- Motorsport Engineering BEng –

Wednesday, 22 April 2020

'This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.'

Signed:



Date: 22/04/2020

Abstract

Considering the similarity of the vehicles on track and the extent to which modifications are restricted by regulations, a vehicles suspension set-up is often considered the make or break of a team's racing season (Morse, 2016). Furthermore, budgets are always tight, whether that's down to regulations, or down to what the team can gather through sponsorship. As a result, more efficient uses of time and money are always of interest to racing teams. Teams, such as UWTSD's MSportEng, need quick and reliable ways to test and validate their suspension setup.

Currently, in order to test the MCR and get it ready for an event, there are two methods used. Test days are used in order try-out, and improve pre-determined settings such as kingpin offset, camber, caster and toe angle, as well as to experiment with new settings for spring and damper rates and tire pressures. This is a drawn out, labour intensive process, as each test has to be run one after the other, requiring time to investigate individual set-ups, teams to make alterations between runs and space to conduct trials. Alternatively, the Base Performance Simulator (BPS) can be used. Whilst this can only test the same settings as a test day, the significantly reduced cost, planning and human resource associated, makes it a far more useful tool. Furthermore, as a result of BPS's ability to record data from simulated sensors, a wider range of information is available to data engineers for the evaluation of each set-up, removing the amount the team has to rely on the driver's, often biased, opinion.

Alternatives used in industry aim to take the inconsistencies of a human drivers out of the equation. The automotive industry aims to improve this area of inaccuracy by using driver-out-the-loop (DOL) simulation (Ziegler and Höpler, no date). This in combination with standardised driving manoeuvres makes the data easier to evaluate and setup changes easier to compare to each other (Wilde *et al.*, 2005).

Although a wide range of software, capable of performing DOL simulations, are available, teams have a tendency to invest in one piece of software, opting to use it even when another piece of software might be better suited to their needs in a certain area.

Acknowledgments

Engineering is about teamwork (*Engineering Teamwork | Ethics Wiki | Fandom*, 2020), so no single person can carry out a study or investigation such as this one, without support. At this point the author would like to acknowledge the following for the parts they played throughout this project. The author would like to recognise the support of Max Hart, whose own projects supported the completion of this one. Furthermore, MSC Software must also be recognised for supplying their software packaged, MSC/Adams and for delivering technical support. The lecturers and technicians at UWTSD, predominantly Tim Tudor and Paul Davies, deserve mentioning for their technical expertise and guidance. Finally, the author would like to thank his parents for their support throughout his university studies.

Contents

Abstract.....	ii
Acknowledgments.....	iii
Nomenclature	vi
List of Equations.....	vii
List of Tables	vii
List of Figures	viii
1 Aims and Objectives.....	1
2 Timing plan.....	1
3 Research/Literature Survey	3
3.1 MCR.....	3
3.2 Vehicle Dynamics	4
3.3 Simulation	6
4 Work carried out.....	8
4.1 Data Collection.....	8
4.2 Simulation Development Approach.....	9
4.3 Quarter Car	9
4.3.1 Cos Bump Test.....	9
4.3.2 SOLIDWORKS Motion Simulation.....	10
4.3.3 Simulink.....	12
4.3.4 Adams	14
4.3.5 Real World.....	16
4.3.6 Results.....	17
4.4 Half Car.....	22
4.5 Full Car	22
4.5.1 Adams	22
4.5.2 Simulink.....	23
4.5.3 Real World.....	38
4.5.4 Results.....	38
5 Discussion and Conclusion	49
5.1 Suspension Evaluation	49
5.1.1 Springs.....	49
5.1.2 Dampers.....	49
5.1.3 Anti-Roll Bars.....	49
5.2 Simulation Software.....	49

5.3	Uncertainties in results	50
6	Pandemic Impact	51
7	Recommended Further Work	52
8	Bibliography	53
9	Appendices.....	58
9.1	Appendix A: Data Collection	58
9.1.1	Tires Spring Rates.....	58
9.1.2	Damper Rates.....	61
9.1.3	Centre of Gravity	65
9.2	Appendix B: Cos Bump Testing	68
9.3	Appendix C: Weight Transfer	74
9.4	Appendix D: Adams Data Input.....	75
9.4.1	Springs.....	75
9.4.2	Dampers	76
9.5	Appendix E: Chassis Stiffness	77
9.6	Appendix F: SimScape Multibody Simulation	77
9.7	Appendix G: IPG CarMaker	78
9.8	Appendix H: Miscellaneous.....	78

Nomenclature

CoG – Centre of Gravity

Adams – Automated Dynamic Analysis of Mechanical Systems

BPS – Base Performance Simulator

DOL – Driver-Out-the-Loop

DIL – Driver-In-the-Loop

GUI – Graphical User Interface

CAD – Computer Aided Design

k – Spring rate

y – Displacement of the ground

a – Distance from CoG to the front axle

b – Distance from CoG to the rear axle

A – ARB spring rate

I – Rotational Inertia

$\emptyset$ – Roll

θ – Pitch

HSB – High Speed Bump

HSR – High Speed Rebound

LSB – Low Speed Bump

LSR – Low Speed Rebound

$()_f$ – Front

$()_r$ – Rear

$()_u$ – UnSprung

$()_L$ – Left

$()_R$ – Right

Newton's Equations

F – Force

F_a – Force due to ARB

a – Acceleration

m – mass

7DoF Model

DoF – Degree of Freedom

m – mass sprung

z – Displacement of the sprung mass

c – Damper coefficient

List of Equations

Equation 1: Newton's 2nd Law of Motion Equation	25
Equation 2: 1DoF Forces	25
Equation 3: 2DoF Forces - Sprung	25
Equation 4: 2DoF Forces - Un-Sprung	26
Equation 5: 4DoF – Sprung	26
Equation 6: 4DoF - Un-Sprung Front	27
Equation 7: 4DoF - Un-Sprung Rear	27
Equation 8: Torque Equations	27
Equation 9: 4DoF - Rotational Inertia	27
Equation 10: 7DoF - Sprung Mass	28
Equation 11: 7DoF - Un-Sprung Mass – Front Left	29
Equation 12: 7DoF - Un-Sprung Mass – Front Right	29
Equation 13: 7DoF - Un-Sprung Mass – Rear Left	29
Equation 14: 7DoF - Un-Sprung Mass – Rear Right	29
Equation 15: 7DoF – Pitch	29
Equation 16: 7DoF – Roll	30
Equation 17: Contact Patch Weight Transfer	39
Equation 18: Centre of Gravity Height	66
Equation 19: Elastic Weight Transfer	74
Equation 20: Un-Sprung Geometric Weight Transfer	74
Equation 21: Sprung Geometric Weight Transfer	74
Equation 22: Total Lateral Weight Transfer – Cumulative	74

List of Tables

Table 1: Suspension Component Masses	8
Table 2: Spring Settings	39
Table 3: Damper Settings	42
Table 4: Anti-Roll Bar Settings	47
Table 5: Tyre Spring Rates	59
Table 6: Ambient Temperature Damper Settings	64
Table 7: 2D mass distribution	65
Table 8: CoG Height Results	67

List of Figures

Figure 1: Timing Plan - Gantt Chart.....	2
Figure 2: Timing Plan - Layout.....	2
Figure 3: MSportEng's MCR (Team MCR - University of Wales, Trinity Saint David, no date)	3
Figure 4: 7 Post Shaker connections (Miller, 2002) & (INDYCAR Q&A: Shaker Rig 101, 2019)	4
Figure 5: SOLIDWORKS Quarter-Car Assembly	10
Figure 6: SOLIDWORKS full front assembly with sledge	11
Figure 7: Spring-Mass-Damper System (Longoria, 2014)	12
Figure 8: Double Spring-Mass-Damper System	12
Figure 9: SimScape Quarter-Car Model	13
Figure 10: Un-Sprung Spring-Mass-Damper System.....	14
Figure 11: MSC/Adams - Editing hardpoints.....	15
Figure 12: SOLIDWORKS Cos Bump Results - Tire vs. Damper Displacement	17
Figure 13: Adams Cos Bump Results - Tire vs. Damper Displacement	18
Figure 14: Simulink Cos Bump Results - Tire vs. Damper Displacement.....	19
Figure 15: Pushrod suspension rocker (Pushrod-Pullrod, 2020)	19
Figure 16: SimScape rocker model.....	20
Figure 17: Adams vs. Simulink (SimScape) - Damper Displacement.....	20
Figure 18: Real World - Cos Bump Results.....	21
Figure 19: Available analyses	23
Figure 20: SimScape Full Car Model.....	24
Figure 21: 1DoF Model.....	25
Figure 22: 2DoF Model.....	26
Figure 23: 4DoF Model.....	27
Figure 24: Force due to ARB	28
Figure 25: 7DoF Model.....	28
Figure 26: 7DoF full Simulink model	30
Figure 27: 7DoF Un-Sprung Model	31
Figure 28: 7DoF Sprung Model	31
Figure 29: 7DoF Pitch Model.....	32
Figure 30: 7DoF Roll Model.....	32
Figure 31: Simulink User Interface.....	33
Figure 32: MoTeC Double Lane Change - Lateral Acceleration Output.....	34
Figure 33: Simulink Double Lane Change - Lateral Acceleration Input.....	34
Figure 34: Simulink Vehicle Set-Up options.....	35
Figure 35: Excel Set-up Sheet.....	36
Figure 36: Custom Set-up input	37
Figure 37: ISO Track Set-up (ISO3888-2, 2011).....	38
Figure 38: Simulink Double Lane Change – Springs.....	39
Figure 39: BPS Double Lane Change - Springs	40
Figure 40: Adams Double Lane Change - Springs.....	41
Figure 41: Simulink Double Lane Change – Dampers	43
Figure 42: Adams Double Lane Change – Dampers.....	44
Figure 43: Simulink Double Lane Change - Contact Patch vs. Damper Forces	45
Figure 44: Simulink Double Lane Change - Dampers	46
Figure 45: BPS Double Lane Change – Dampers	46
Figure 46: Simulink Double Lane Change - ARB WT.....	47

Figure 47: Simulink Double Lane Change - ARB Roll	48
Figure 48: Tyre Stiffness Rig	58
Figure 49: Front Tyre - Force Deflection Curve	59
Figure 50: Rear Tyre - Force Deflection Curve	59
Figure 51: Spring rate at given tyre pressure	60
Figure 52: Damper Dynamometer	61
Figure 53: Damper cross section diagram (Calver, 2001)	62
Figure 54: Front Damper Force vs. Velocity Graph	63
Figure 55: Rear Damper Force vs. Velocity Graph	63
Figure 56: 2D CoG Calculation	66
Figure 57: CoG Height test set-up	67
Figure 58: 1-Cos(x) Bump dimensions	69
Figure 59: Cos Bump Test setup	70
Figure 60: 1-Cos(x) Bump	71
Figure 61: Caster induced Weight Transfer (Caster – Geometry Explained – Suspension Secrets, no date)	74
Figure 62: Chassis stiffness (Herrmann, 2016)	77
Figure 63: SimScape Multibody Vehicle Simulation	77
Figure 64: Double Lane Change - Lateral Acceleration	78
Figure 65: Simulink Double Lane Change - Roll Stiffness distribution	78

1 Aims and Objectives

The MSportEng team does not currently use any DOL (Driver-out-the-Loop) simulation to evaluate vehicle set-up on each track before individual events. As a result, the primary target of the project was to create and evaluate this type of vehicle model of the MCR. The team should be able to use the simulation in order to investigate the effect of new suspension components on lap times.

A secondary target was set to evaluate created models of the car in different pieces of software. The purpose was to determine which software would produce the best simulation model for the team and which would be best suited for the type of investigation the team would be most likely to need.

2 Timing plan

Producing and validating a simulation is an iterative process (Still, 2017). Simulations must be run to determine where potential issues might have arisen, as well as comparing these results to physical testing to ensure results are realistic. Other vital steps include data gathering, as well as a final evaluation step after every simulation run.

The timing plan, laid out below, helped keep this project on track, making sure that milestones and gateways were met. By showing when tasks needed to be started or completed, this prevented tunnel vision on a single issue, which could have stunted overall progression (Rosenhead, 2012). Furthermore, the timing plan made collaborative areas of the project easier to plan (Artto, 1999).

As the Gantt chart shows, throughout the project, multiple tasks had to be completed simultaneously. Whilst this was essential in ensuring the project was completed on time. The fact that no single task occupied the author 100% of the time, meant that the time could be used more efficiently, reducing wastage (Allcott, 2014).

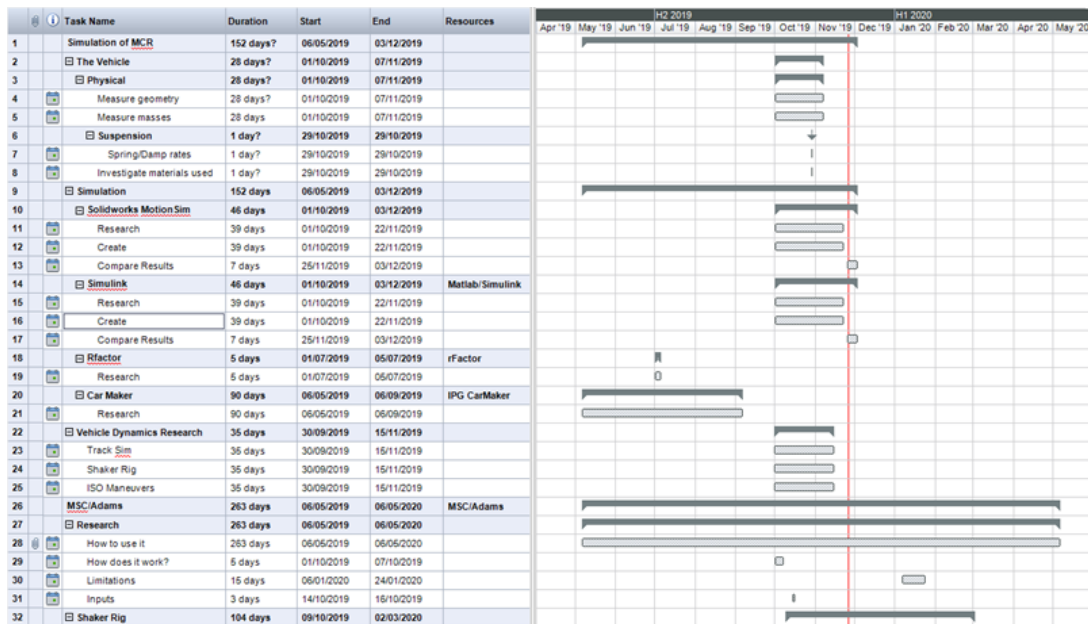


Figure 1: Timing Plan - Gantt Chart

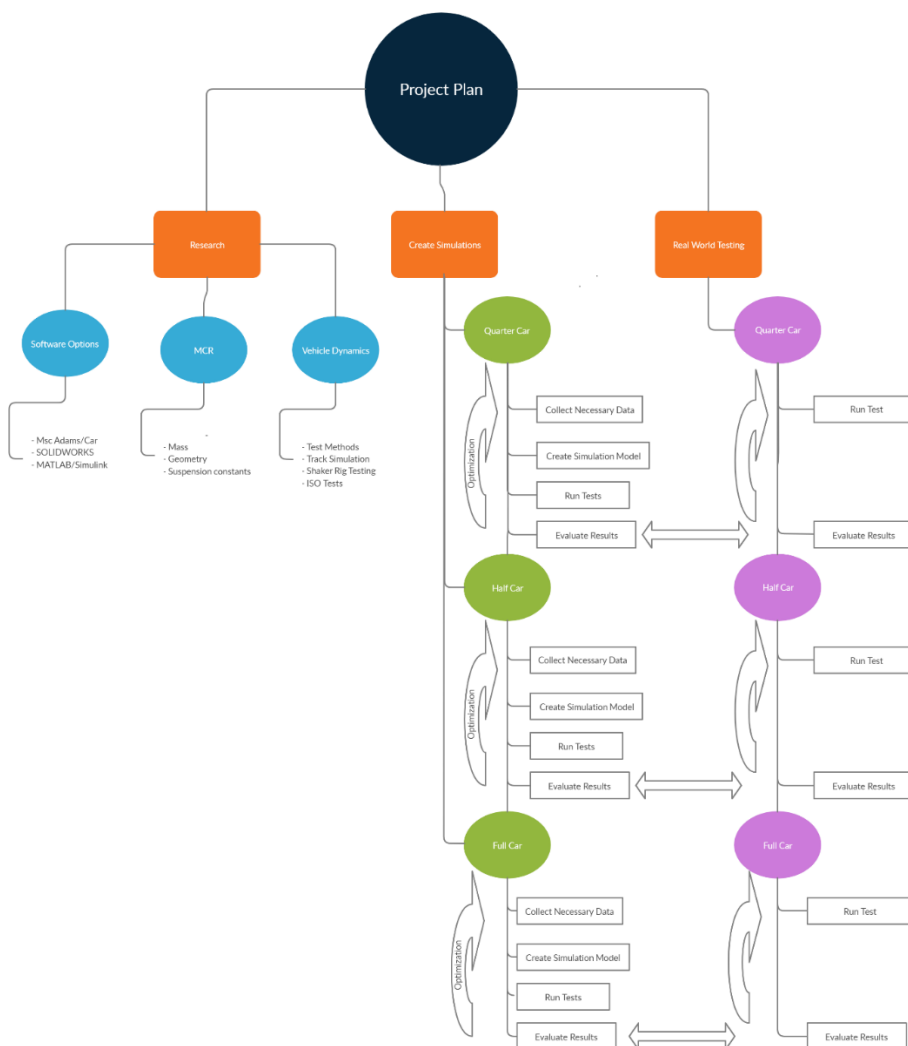


Figure 2: Timing Plan - Layout

3 Research/Literature Survey

3.1 MCR

This piece of work is based on the Universities of Wales Trinity Saint David's MCR race car, which is run and maintained by students on the MSportEng team. A substantial amount of data used throughout this project was collected by the team through on-track testing and at race events.



Figure 3: MSportEng's MCR (Team MCR - University of Wales, Trinity Saint David, no date)

The vehicle competes in the Sports 2000 championship in races lasting between 30 and 50 minutes depending on the type of event. These events are always preceded by a qualifying session and sometimes by an optional practice session. All events are run, on purpose-built racetracks (*Sports 2000 Duratec Championship - SRCC - SPORTS RACING CAR CLUB, 2020*).

The data collected throughout testing, qualifying and races can be used for the purpose of evaluating the accuracy of simulation models created. The vehicle is fitted with a MoTeC data logger, logging at 0.01 second intervals. Amongst a range of 300 channels (*MoTeC > C187 > Overview, 2020*), the data the logger is capable of recording spring/damper length, vehicle speed and lateral and longitudinal acceleration.

All MCRs are rear wheel drive, with a rear mounted 2.0L Ford Duratec engine, fitted with a Hewland gearbox. The suspension geometry consists of double wishbone push-rod geometries front and rear (*MCR S2 – MCR Race Cars, 2020*). The coil over dampers allow for springs to be exchanged as necessary, whilst bump and rebound can be adjusted independently.

3.2 Vehicle Dynamics

There are multiple different ways to characterise the suspension behaviour of a car. Some more accurately represent real world driving behaviours, whilst others are easily repeatable in real life and in simulation.

3.2.1.1 7-post Rig

This method of testing works by placing the vehicle on a platform. The platform is attached to the vehicle by 7 connections, four of which are the connections to the wheels, and the other three are the connection to the chassis. The rig works by moving these 7 connections, or posts, vertically to create movement within the wheels as the body (*7-Post Rig Testing Facility, 2019*).

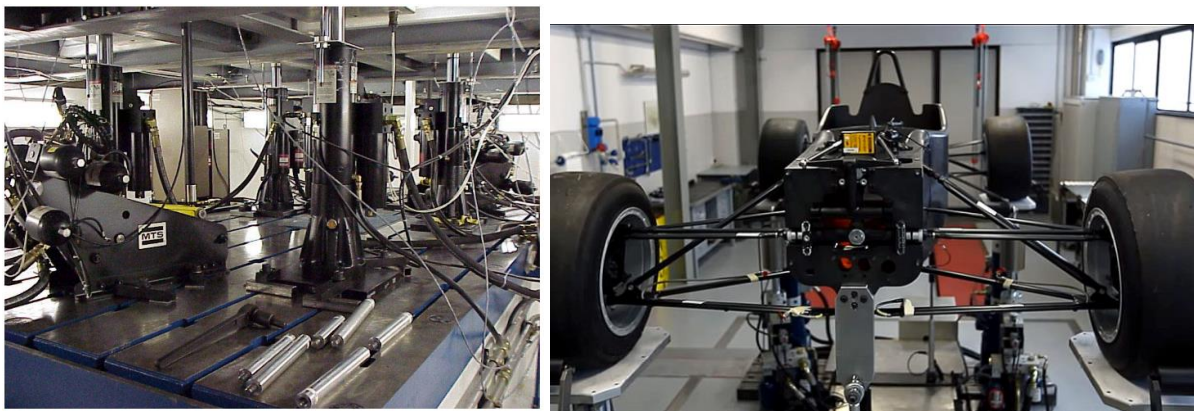


Figure 4: 7 Post Shaker connections (Miller, 2002) & (*INDYCAR Q&A: Shaker Rig 101, 2019*)

Using a 7-post rig provides a method to evaluate a suspensions behaviour that is easily transferable from the real world to a simulation. The data that is fed into the shaker rig tends to be data that was collected from the real world (Simons, 2019). This same data can later be read into a simulation model as an input.

Using this method allows the tester to isolate certain characteristics of the car (*INDYCAR Q&A: Shaker Rig 101, 2019*) since the inputs into the geometry are controlled and repeatable. This means that if a part of the suspension, such as a part of a geometry or a damper setting, is changed, the tester knows for certain that it is solely that change that is affecting the results.

The downside to using this method of testing is primarily the cost of purchase and maintenance (*INDYCAR Q&A: Shaker Rig 101, 2019*). This is why the 'real world' part of the 7 post rig will not be used as part of this project, however its basic principle will be carried forward.

3.2.1.2 Track Simulation

Depending on the amount of data and time available, track simulations can be a very powerful and cost effective way to evaluate the effectiveness of changes made to a car (Dempsey *et al.*, 2012). There are however significant issues in correlating simulation data to real world data when using a track as your proving ground (Essers, 1997).

In motorsport, the most important area of interest is how quickly a vehicle can go around a track (Edelstein, 2016). Considering this, it makes sense to look at a vehicle's performance on track throughout its entire development phase. However, lap times are not everything since lap times with the same driver change from lap to lap or day to day (Smith, 1978). One way to overcome this issue is by averaging out a large number of laps, however then there is the issue of what to do when the driver is unavailable. Some types of simulation software attempt to overcome this issue by developing software-controlled drivers (Essers, 1997). Whilst these don't necessarily produce the fastest lap times, results are consistent and repeatable. Results other than lap times can be used to indicate potential gains.

3.2.1.3 Testing with Standards

Testing using a standard provides a method to consistently repeat real world tests and compare these to simulations. In the UK these are governed by BSI's ISO standards (*ISO - BSI - British Standards Institution*, 2020). The track to be used pre-defined by the standard, as well as in some cases speed and steering input are also given. These tests are commonly used within the automotive industry in order to produce repeatable results, however, they are not often used within motorsports, due to the way this type of test does not commonly test a vehicle at its limits.

Examples, considered for use in this investigation include tests to evaluate behaviour of a vehicle on a split coefficient friction surface (ISO14512, 1999), severe lane-change (ISO3888-1, 2018 & ISO3888-2, 2011), a steady state circle (ISO4138, 2012), braking under cornering (ISO7975, 2019) and lifting off under cornering (ISO9816, 2018). These standards define manoeuvres that can be used to provide consistent bases for comparison when evaluating the kinematic behaviour of a car.

3.2.1.4 Bump

In order to allow accurate correlation, a test method that can be carried out in simulation and in the real world is required. This test needs to be defined as accurately as possible and ideally should not rely upon the ability of a driver. Simplicity is key as it makes re-creation more precise. The use of a bump in the shape of a $1-\cos(x)$ curve has proven useful for testing motorbike suspension (Morley, 2007).

3.3 Simulation

3.3.1.1 MSC Adams/Car

Adams/Car, developed by MSC Software (MacNeal-Schwendler Corporation) is a multibody-system simulator using a GUI, specifically designed for use on four wheeled vehicles (*Adams Car - Real Dynamics for Vehicle Design and Testing*, 2020). Adams, which stands for “Automated Dynamic Analysis of Mechanical Systems”, is the underlying solver and is capable of simulating any multibody mechanical system designed by its user. It uses the process of designing a 3D geometry, separating them into rigid and hinged elements and simulating them through a “real world physics” scenario (*MSC Software’s Adams Multi-body Dynamics & Motion Analysis software*, 2020).

In the /Car interface, Adams allows the user to use templates to define the geometry of the vehicle. The templates define subsystems, such as the chassis or powertrain, of the vehicle. An example of how this is used is the front suspension, where a double wishbone suspension with pushrod spring/dampers was taken from the FSAE database, supplied by MSC. The *hard points*, which define the size and orientation of components within the subsystem, can be modified to suit the geometry of the car being simulated.

The solver uses the equations of motion defined by the Newton-Euler formulation and the Lagrange formulation to define the motion of the geometry (*Using ADAMS/Solver*, 2002), including how the components interact with each other and the outside world (*An Overview of How to Use ADAMS/Solver Chapter 1 ADAMS/Solver and MSS*, no date).

Adams is capable of two modes of simulation. The first, which is the one used throughout this investigation, assumes all bodies are rigid, leaving them with 6 degrees of freedom. Whilst this is less realistic, it results in equations that are easier to solve and therefore shorter simulation times. Furthermore, less information is required to be found via testing for this method. Finally, many other modes of simulation that will be used for the purpose of comparison, are not capable of simulating flexible bodies. The alternative gives each body a stiffness allowing them to deform under load (*Using ADAMS/Solver*, 2002).

3.3.1.2 SOLIDWORKS Motion Simulation

Since CAD models are often required when creating simulations, some CAD packages, such as SOLIDWORKS implement their own simulation package, allowing the designer to have an idea as to whether or not their design is going to work for its intended purpose, without requiring an external program.

SOLIDWORKS Motion Simulation assumes all components are rigid whilst allowing the user to define how each of the parts interact with each other. This includes defining a wide range of hinges and which parts can apply a contact load on each other (*Engineering Design and Technology Series An Introduction to Motion Analysis Applications with SolidWorks Motion, Student Workbook*, 1995).

3.3.1.3 MatLab Simulink

MatLab (Matrix Laboratory), developed by MathWorks, combines a programming language and development environment, originally designed for matrix manipulations. It has become a more broad language, with the implementation of functions and other add-ons, capable of being exported for use in hardware and as a result is being used as a fully-fledged programming language (*Simulink - Simulation and Model-Based Design - MATLAB & Simulink*, no date).

Simulink is a graphical programming environment within MatLab allowing users to develop programmes without the need to understand the full MatLab programming language. A number of libraries of *blocks* are available to the user, each one aimed at performing different groups of tasks (*Introduction to Simulink with Engineering Applications - Steven T. Karris - Google Books*, no date). SimScope is an example of a library of blocks. It is designed to be used to represent physical components and processes, removing the need for the user to go back to first principles (Miller and Wendlandt, no date).

3.3.1.4 Base Performance Simulator

BPS (Base Performance Simulator) is a motorsport track racing simulator based on the rFactor gaming engine (Partridge, 2013). Along with its own user interface for rFactor, BPS also provide their own simulator rig and hardware (*Home / Base Performance*, no date). At UWTSD, the single seater configuration, made to imitate a Formula 3 tub is used. rFactor is used extensively throughout the motorsport industry giving engineers the opportunity to test their cars on a range of tracks, in order to tune various areas of the car, including suspension geometry or gear ratios. Furthermore, the simulator gives teams the opportunity to train their drivers on each track, prior to an event (*Motorsport / rFpro*, no date).

4 Work carried out

4.1 Data Collection

Each software package takes a different approach in the way in which it goes about simulating the car. Depending on which software package is being used, different pieces of information are required, and sometimes in different formats. An example of this is the un-sprung mass. Whilst in the SimScape model it would only be necessary to know the overall corner un-sprung mass, in order to create the SOLIDWORKS Motion Sim, the individual mass of each un-sprung component was required. This also included assuming that one third of partly sprung components, such as push-rods, was sprung and the rest un-sprung. A comprehensive list of all unknowns was collated so that information gathering could commence.

Table 1: Suspension Component Masses

<u>Part</u>	<u>Front/Rear</u>	<u>Sprung/Un-Sprung (S/U)</u>	<u>Mass (kg)</u>
Brake Disk	Front	U	4.2
Push Rod	Front	S/U	0.6
Tie Rod	Front	S/U	0.35
Stub Axle (w/o upright)	Front	U	0.9
Upright (w/o Stub axle)	Front	U	3.65
Rim (w tire)	Front	U	9.25
Upper Wishbone	Front	S/U	0.85
Upright	Rear	U	6.75
Lower wishbone	Rear	S/U	1.6
Upper wishbone	Rear	S/U	1.3
Tie rod	Rear	S/U	0.45
Lower wishbone	Front	S/U	2.05
Rim w. tire	Rear	U	11.05
Rocker	Rear	S/U	0.7
Push rod	Rear	S/U	0.65
Rocker	Front	S/U	0.55
Steering rack	Front	S	3.6
Brake calliper	Front	U	2.05
Damper	Front	S/U	1.45
Damper	Rear	S/U	1.55
Chassis		S	200.5

4.2 Simulation Development Approach

As in programming, simulations often do not work successfully on the first attempt. The models are built and as much detail and information is input as possible in order to achieve the best possible correlation with the real-world. Whilst this is a sound approach in theory, in practice, where the first attempt is often unsuccessful, this large amount of data can be difficult to sift through when looking for the cause of an error. Furthermore, the failure of a simulation is not always obvious since, for the purposes of this investigation, a poorly correlating simulation could also be considered a failed simulation. As a result, it was decided that the models should be built in such a way that their complexity would gradually increase. The model would be evaluated at each stage, ensuring that there are no mistakes before moving on to a more detailed model. This approach would first see quarter-car, then half-car and finally full-car models being created. At each stage of development, the simulations could be compared, not only to the results from other simulation environments, but also to the real vehicle.

4.3 Quarter Car

4.3.1 Cos Bump Test

In order to compare the simulation models to each other and to the real-world vehicle, a test was required that could be created identically in each simulation environment and in the real world. Furthermore, it was essential that the test required minimal to no driver input, in order to ensure that human error would not be the cause of any poor correlation. Finally, it would be ideal if this test was as simple as possible, reducing the number of areas of possible uncertainty.

Although the ISO-standards considered previously were a good start, they all required some level of driver input when they were re-enacted in the real world. Eventually it was determined that, to keep human error out of the equation, the vehicle would have to move at a constant velocity and in a straight line, as this can easily be measured and repeated. The issue with a steady state scenario, such as those described in ISO 4138:2012 or ISO 10521-2:2006, is that, although sprung forces and vehicle geometry could be evaluated, due to the chassis remaining at a constant height, the dampers were not being affected. In order to involve the vehicles dampers in the scenario, an obstacle was positioned in front of one side of the car. This would upset the steady state behaviour of the car, causing the dampers to act out a force on the suspension system.

It was determined that a $1-\cos(x)$ function with an amplitude of 40mm would be the most accurate way of defining the obstacles geometry. Step inputs might have been easier to re-create in simulation, however there are many details that must be defined in order to ensure perfect replication of the obstacle. The $1-\cos(x)$ function, sufficiently defines all aspects of the obstacle. The speed at which the

vehicle passes over the bump was defined and kept constant throughout, so that only changes to the suspension would affect results. The full test procedure (see appendix) describes all aspects of how the test was carried out.

4.3.2 SOLIDWORKS Motion Simulation

Now that the test to be carried out was known, it was possible to begin creating models for the purpose of simulation. The first model to be created was the SOLIDWORKS model. Initially, this consisted of a quarter car model made up of upper and lower wishbones, the upright, a rim, a tire, the chassis, and an approximation of the steering rack. The basic CAD model had already been created for the purpose of creating an aerodynamic package for the MCR however, it had not been designed for use with Motion Sim (Hart *et al.*, 2019). Once the geometries and masses of the individual components had been confirmed, it was possible to assemble the components into a quarter car assembly and create a Motion Simulation. Finally, to complete the motion simulation, the geometry of the road, including the Cos Bump, was created.

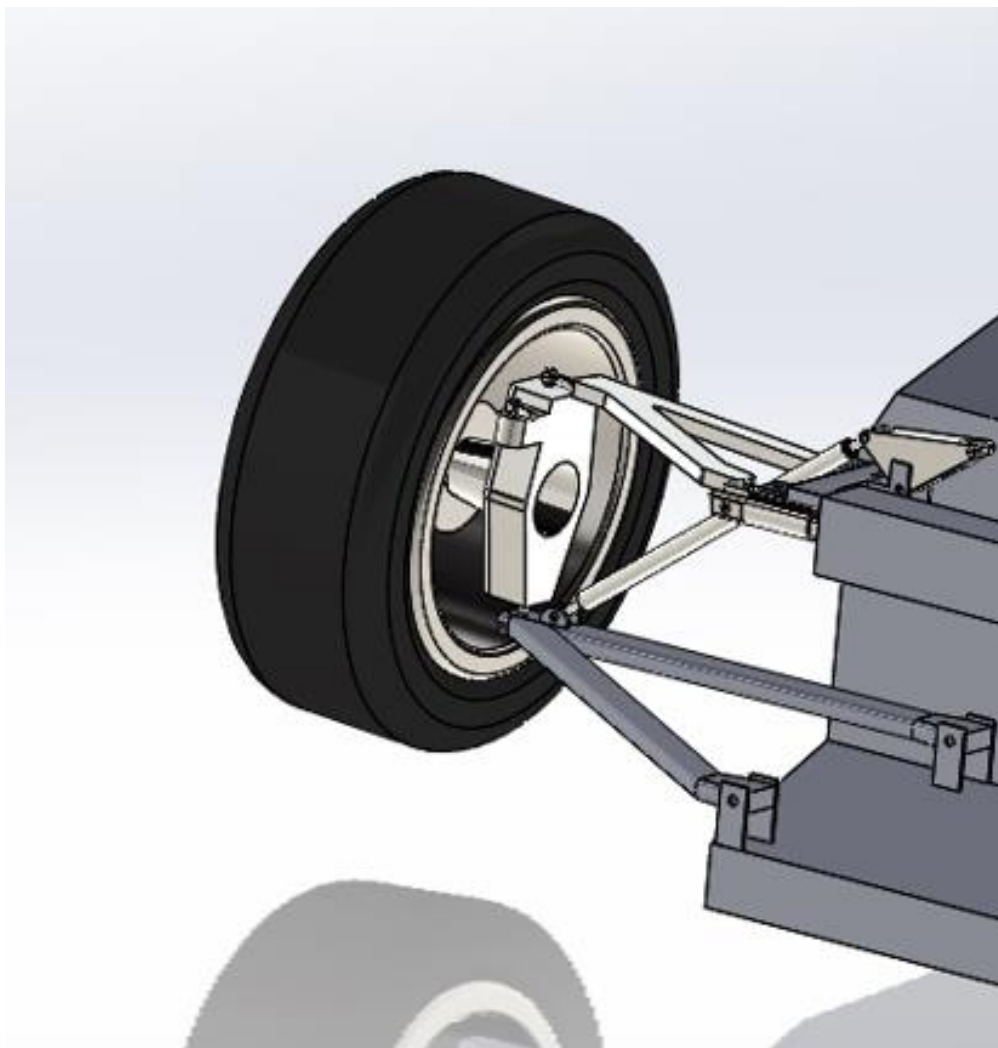


Figure 5: SOLIDWORKS Quarter-Car Assembly

In order to be able to run the simulation, a few assumptions had to be made. The first assumption was that the chassis was at a fixed height in relation to the road. For a quarter-car model, making the chassis sprung would not be necessary. Furthermore, the steering system was locked in relation to the chassis, so that the road wheel would continue to point in the direction of motion.

Next before the simulation could be run, the built in spring and damper tools were used instead of modelling these components in the conventional sense. These tools allow the user to pre-define spring/damping rates and resting length, as well as their points of contact with the assembly (*SOLIDWORKS Help*, 2018). The alternative method of modelling these components would require a significant amount of investigation into the material used for the springs, and for the dampers to be assembled from its own individual components (Chang, 2019).

For the purposes of comparison and since a full chassis was available in CAD, it was decided to run the test using both a fixed chassis height model, more comparable to the Adams and Simulink analysis, as well as a floating chassis analysis, more comparable to the real world analysis. Normally, the floating chassis would be achieved by modelling the rear suspension components of car, however, to save time, a sledge geometry was installed instead. This would allow the vehicle to behave approximately as expected, so that the front suspension behaviour could be investigated. With the floating chassis, the 1-Cos x bump test was repeated as before and compared to the behaviour of the real car.

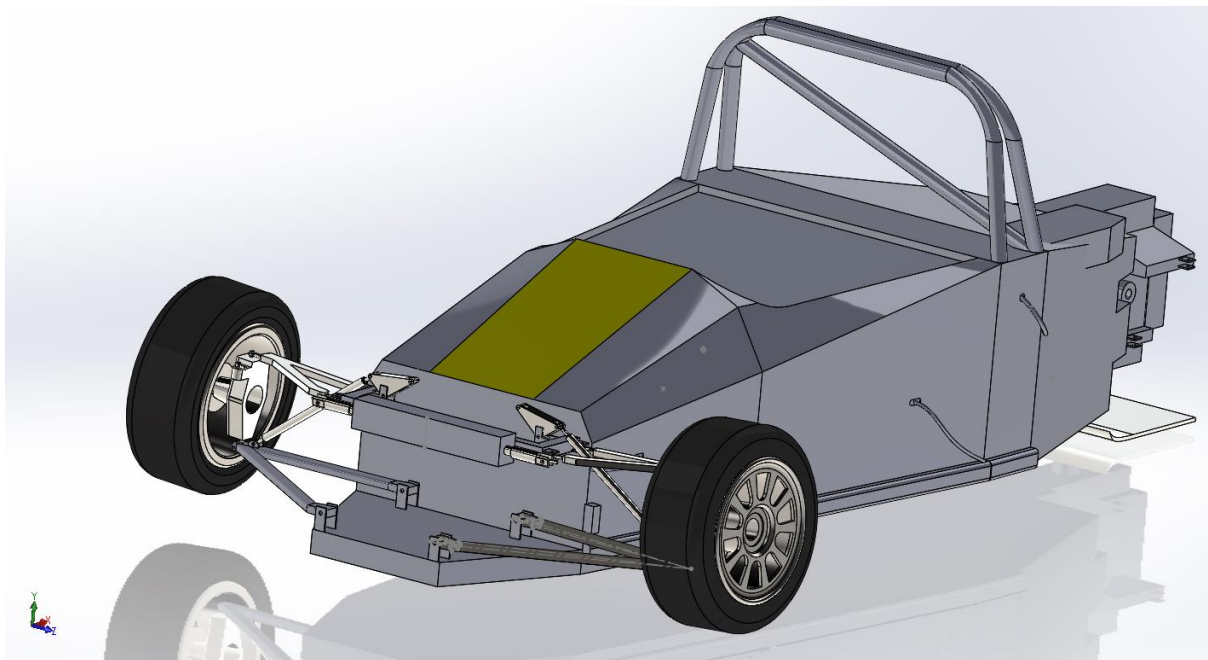


Figure 6: SOLIDWORKS full front assembly with sledge

In order for the motion simulation to run successfully, material properties, including density, stiffness and maximum yield stress have to be defined for each component (*SOLIDWORKS Help*, 2018). Whilst SOLIDWORKS provides an extensive library of materials, it is not recommended to assume that these properties are correct for the components on the car. There is the alternative option of defining these material properties as a custom material however, this path was not taken as determining these properties for each component, as with the springs, would have been a time-consuming task. Since the mass was defined manually, this had no effect on the simulation, as SOLIDWORKS assumes all components to infinitely stiff. Other values are only needed for FEA analysis of the individual component after the motion simulation (Lombard, 2013).

4.3.3 Simulink

The next model to be created was the SimScape model, created in the MATLAB based Simulink environment. The basic principle of this simulation approach draws parallels between the suspension geometry and a spring-mass-damper system. For this scenario, two systems, such as the one in Figure 7, are put together end to end, such as in Figure 8. The tire is seen as one spring/damper whose properties are represented by k_u and c_u . The un-sprung mass is represented by the lower mass in the system m_u . The springs and dampers are represented by the other spring and damper, k and c , and finally the sprung mass by the remaining mass, m . The values y , z and z_u represent the displacements of the three separate sections.

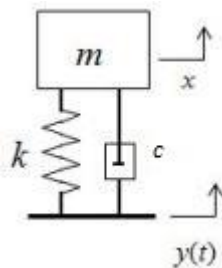


Figure 7: Spring-Mass-Damper System (Longoria, 2014)

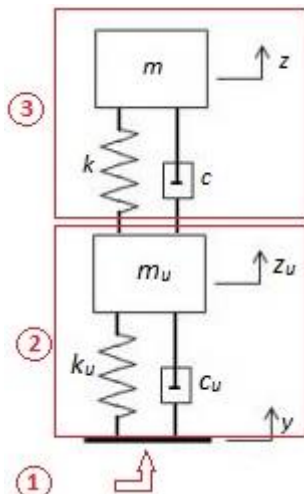


Figure 8: Double Spring-Mass-Damper System

The models described above can fundamentally be broken down into 3 parts per quarter car. The main input into the system is the vertical displacement of the ground. This can be seen in Figure 9 by the subsystem titled *Road*. Due to the components available in the SimScape library, this subsystem first reads the road data from a MATLAB file, containing the vertical displacement of the road. Before this information can be passed on to the SimScape *Ideal Translational Velocity Source* block, the displacement must be differentiated into the velocity of the road, using a *Derivative* block, and then converted from a Simulink signal to a SimScape signal using a *Simulink-PS Converter* block.

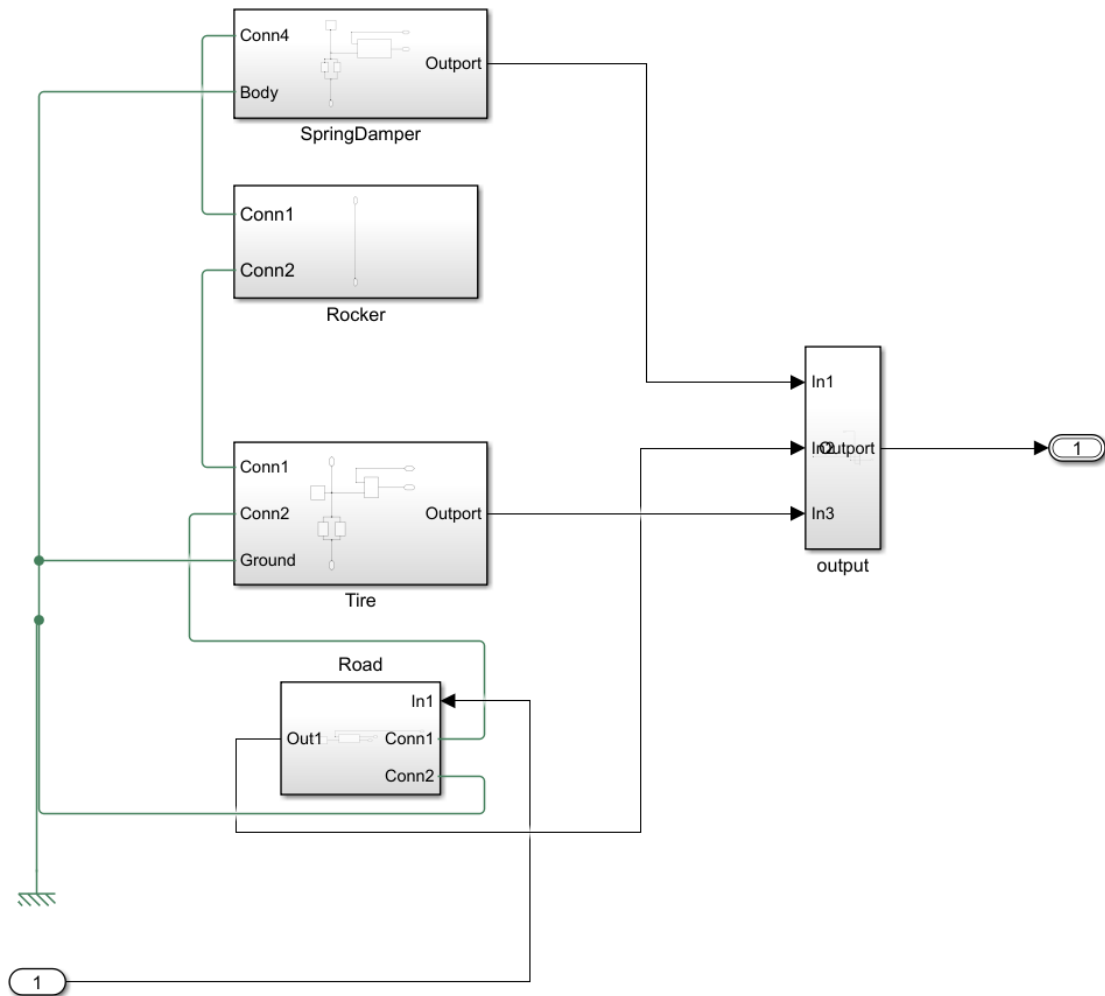


Figure 9: SimScape Quarter-Car Model

Section 2 of the system is the mass, spring rate and damping rate of the un-sprung mass. Figure 10 shows how these three components and their properties are represented by three blocks in the SimScape environment. It is evident that this diagram has been derived from the spring-mass-damper system, seen in Figure 7. A detail which the double spring-mass-damper system does not take into account is the motion ratio between the wheel and the spring/damper in the MCR. These are represented by the *Rocker* subsystem and is positioned between parts 2 and 3.

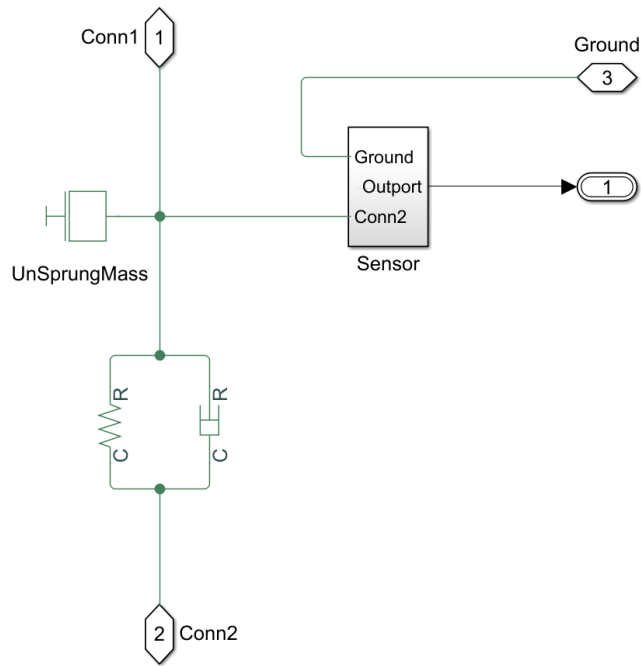


Figure 10: Un-Sprung Spring-Mass-Damper System

Finally, section 3 is almost identical to section 2, the difference being the constants within each block for the sprung subsystem containing different values to the un-sprung subsystem. Since the Cos Bump test is conducted at a constant lateral and longitudinal velocity, there is no need for an input for this information. Whilst a number of inputs are required as constant values for this model, such as spring and damping rates, the only variable input is the displacement of the ground (y).

4.3.4 Adams

In order to create the model in MSC/Adams, the Adams Car interface was used, as this allowed the use of predefined templates to make the creation of a model a simpler process. After searching the standard libraries MSC offers, including the `acar_shared`, `acar_concept`, `FSAE_2012` and `FSAE_2018` databases, it was determined that the `fsae_front_suspension.asy` assembly, from the `FSAE_2018` database, would be the most appropriate to use, since it was designed for use with a double-wishbone, push-rod suspension set-up, such as the one used by the MCR.

Although the chosen template represented the correct type of suspension, the dimensions of the geometry were those of a formula student car. Using the table interface, the locations of the hardpoints could be defined. In order to help determine the values, the CAD model in SOLIDWORKS was used to measure the location of each hardpoint with a universal reference point.

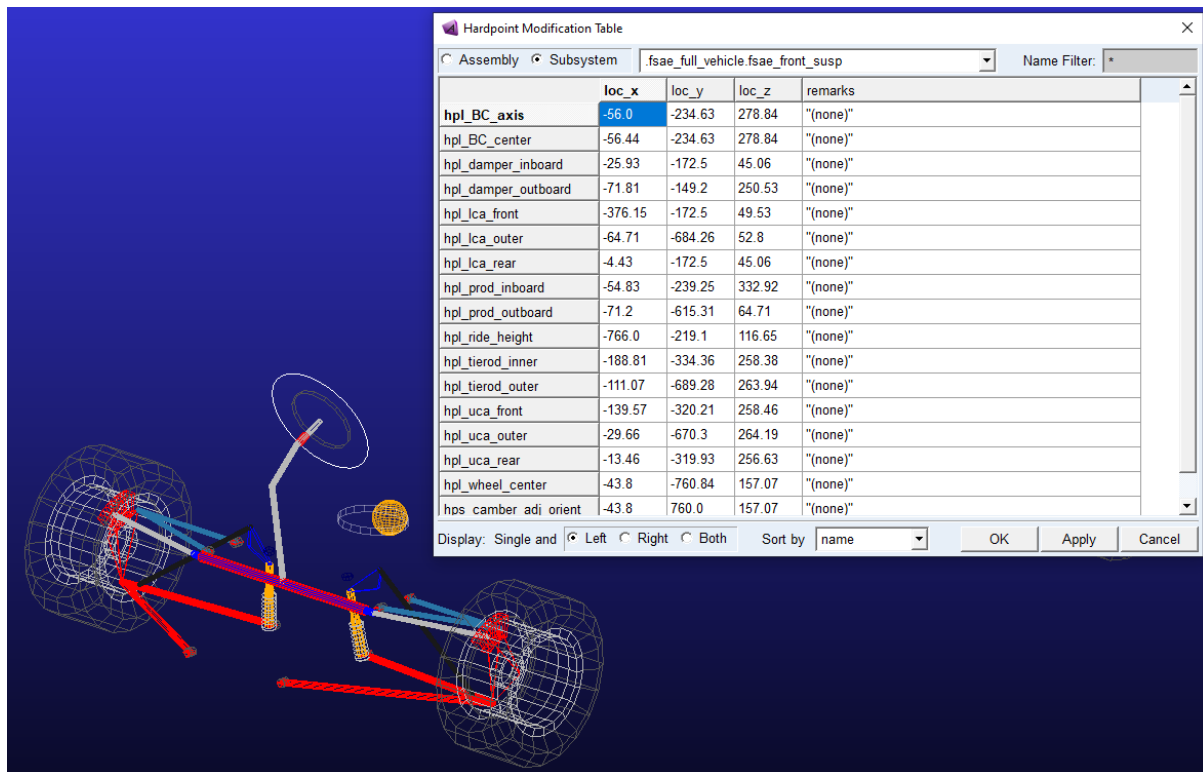


Figure 11: MSC/Adams - Editing hardpoints

With its standard settings applied, Adams assumes most components to be infinitely stiff. Motion is only allowed at pivot points within the geometry. Springs and dampers, whose settings can be defined within the settings definition file for that component (see appendix), are exceptions. It is also possible to change the characteristics of the tires within a dedicated property file.

Since only the suspension has been modelled at this point, the choices of simulation scenario are limited to the *Suspension Analysis* subsection. As the intention was to use a pre-defined road profile, the *External Files* analysis was chosen, allowing the user to use a customised file as the y input.

Once the analysis has been calculated, the *Postprocessing Window* can be viewed to see the output of the simulation. This window allows a wide range of data to be visible from that analysis. Data is recorded at the same rate as the input file provided by the user. If data does not show the expected results, it is recommended to use *Animation Controls* to view an animation of the analysis ('Formula SAE Applications using Adams/Car', 2018).

4.3.5 Real World

To conduct the real-world test, there were a few pieces of equipment required. It was essential that the car was available and set-up correctly. The procedure for ensuring the car was ready for this test, as well as the test procedure itself, can be seen in the appendix. Furthermore, because they were not being modelled in any of the simulations, the anti-roll bars were disconnected from the suspension. Since the car was moving in a straight line, the steering system had to be locked into position, in order to avoid the need for a person to introduce discrepancies into the results by not being able to hold the steering wheel perfectly straight throughout the manoeuvre.

It was also important for the vehicle to be fitted with a data logger. In the case of the MCR, a MoTeC data logger, measuring vehicle speed, GPS location, 3 directions of acceleration, and the change in length of each damper, was used. The channels for these signals had to be zeroed correctly to ensure results were calibrated. Strain gauges, attached to the push rods were also recorded, allowing an approximation of the contact-patch loads to be measured.

Finally, a model-board version of the 1-Cos bump, created using a CNC mill, was placed in the path of the vehicle. Because of the geometry of the bump, its ends were fragile. In order to avoid them breaking when the vehicle rolled over it, the entire bump was mounted on a 10mm piece of MDF. To avoid the 90° incline at the start and end of the MDF affecting the results, a 0.5m run up and run off, before and after the bump, were included, giving the car the opportunity to settle. However, to ensure that any potential uncertainty that might arise from this change in the test procedure, each of the simulations were adjusted to match.

4.3.6 Results

The intention of the quarter car model was primarily to ensure that a simplified version of each simulation was working as expected before moving on to a more complex version. In order to achieve this, the same test procedure had to be run in each simulation, ensuring that all settings were kept as similar to each other as possible.

The first important step after running any simulation is to sanity check the results (Blundell and Harty, 2015). Do the results look logical? This requires the user to have some understanding of what they are simulating, so that they know what they are expecting. It also requires the user to understand the results and how to interpret them. It is also important to compare the results.

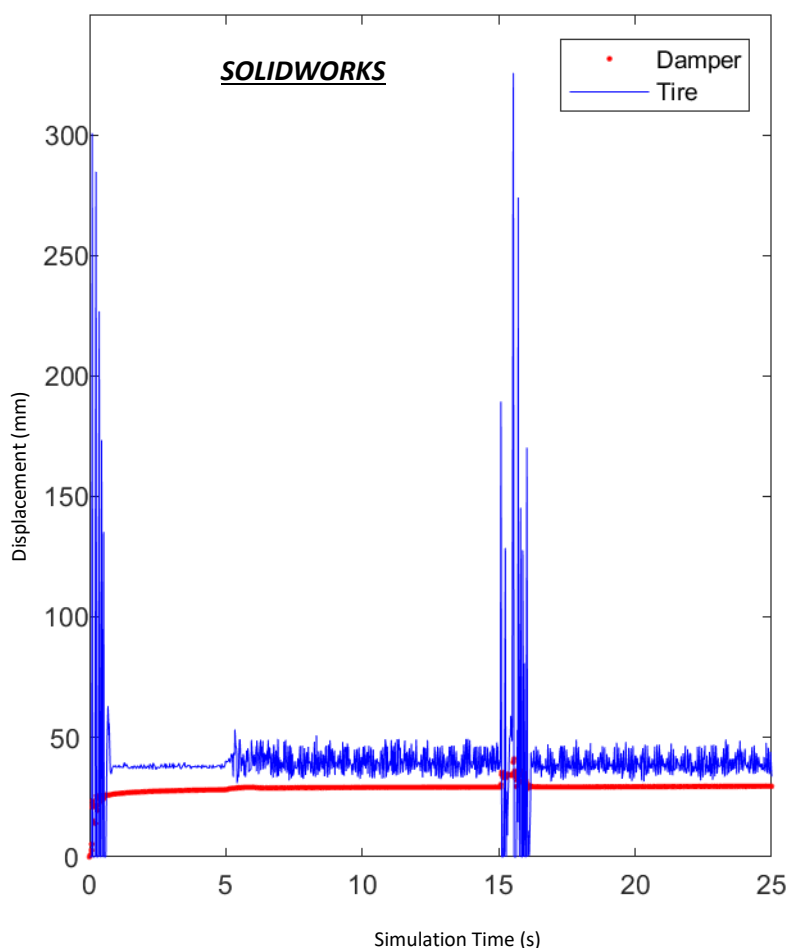


Figure 12: SOLIDWORKS Cos Bump Results - Tire vs. Damper Displacement

Looking first at the results of the SOLIDWORKS analysis, Figure 12, shows some initial erratic behaviour of the displacement before it settles down to a more constant value. This is followed, after approximately 15 seconds, by a peak in the displacement before the results settle again. The initial disturbances could cause some concern however, when looking at the set-up of the test in SOLIDWORKS it becomes apparent that this was expected. Before the test procedure is run, the SOLIDWORKS model has a few seconds at static, allowing the simulation and the vehicle components to come to a static resting position. The random values are a result of the simulation attempting to resolve these initial conditions. The peak seen later on is evidently a result of the vehicle running over the Cos bump. Whether or not the magnitude of these values is correct can only be confirmed when the results are compared to another simulation. An initial look at the magnitude of the displacement indicates that an error has occurred, as a 300mm displacement would not normally be expected. Furthermore, the difference between the peak in displacement for the tire and the damper does not appear correct. It is immediately obvious that the tire moves further than the damper moves, which, considering the purpose of the rocker, is obviously wrong. This error becomes obvious when the Simulink and Adams results are considered and is likely the result of an incorrectly installed rocker. When evaluating these results, one should also note that throughout steady state section, the signal coming from the tires is not following a constant value. This is most likely down to the precision of the simulation in measuring this value. Considering the magnitude of this error is over 20mm, it must be taken into account when determining the accuracy of the results. Finally, if the x-axis had showed *Horizontal Vehicle Distance* as opposed to *Simulation Time*, comparing results against Adams, Simulink and the real-world results would have been easier and more accurate.

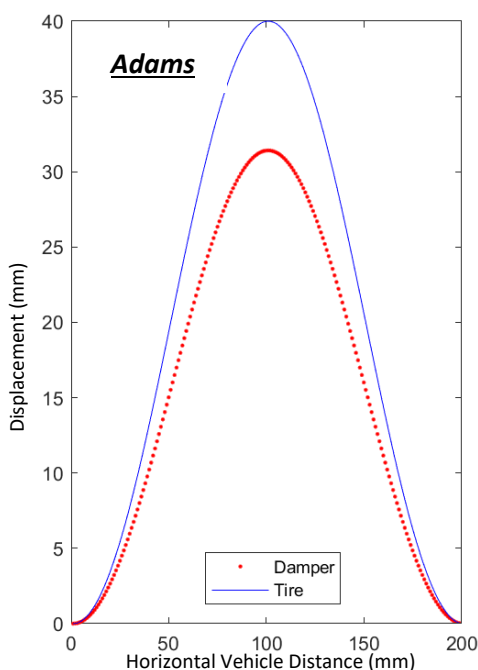


Figure 13: Adams Cos Bump Results - Tire vs. Damper Displacement

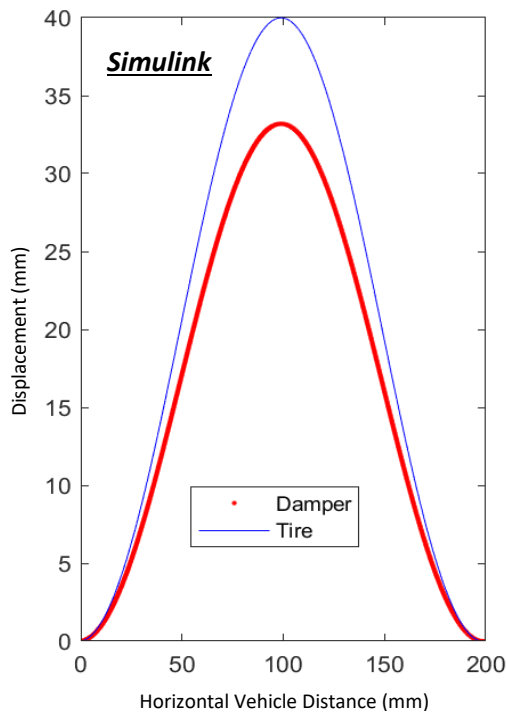


Figure 14: Simulink Cos Bump Results - Tire vs. Damper Displacement

In Adams and Simulink, the results appear very similar. The input (tire displacement) of each is identical. The output appears to be following the same trend, peaking at the same point. However, overlaying the results, as was done in Figure 17, shows that the results are not identical, as initially assumed. Although trends and peaks are similar, the Simulink values have a higher amplitude as well as a steeper incline before and after the apex. Furthermore, the Adams results are slightly delayed as shown by a displacement of the apex towards the right. This is possible as a result of the way in which the rocker was modelled. In Adams, the entire geometry of the rocker and of the pushrod system is modelled and taken into account when calculating the tire to spring/damper motion ratio. In the Simulink/SimScape model on the other hand, a simple lever was used, as previously discussed. This does not account for the delayed response of the Adams geometry. This is likely the result of the Simulink model's inability to account for the inertia of the geometry, something which Adams is capable of.

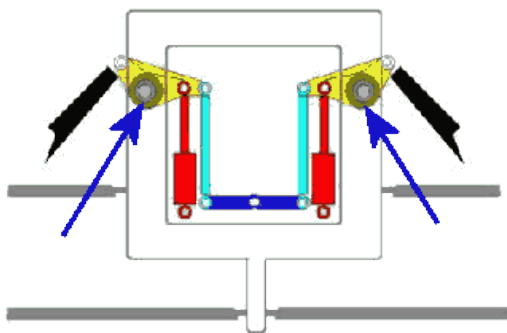


Figure 15: Pushrod suspension rocker (Pushrod-Pullrod, 2020)

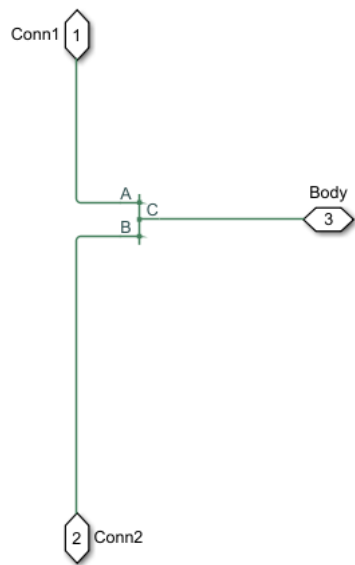


Figure 16: SimScape rocker model

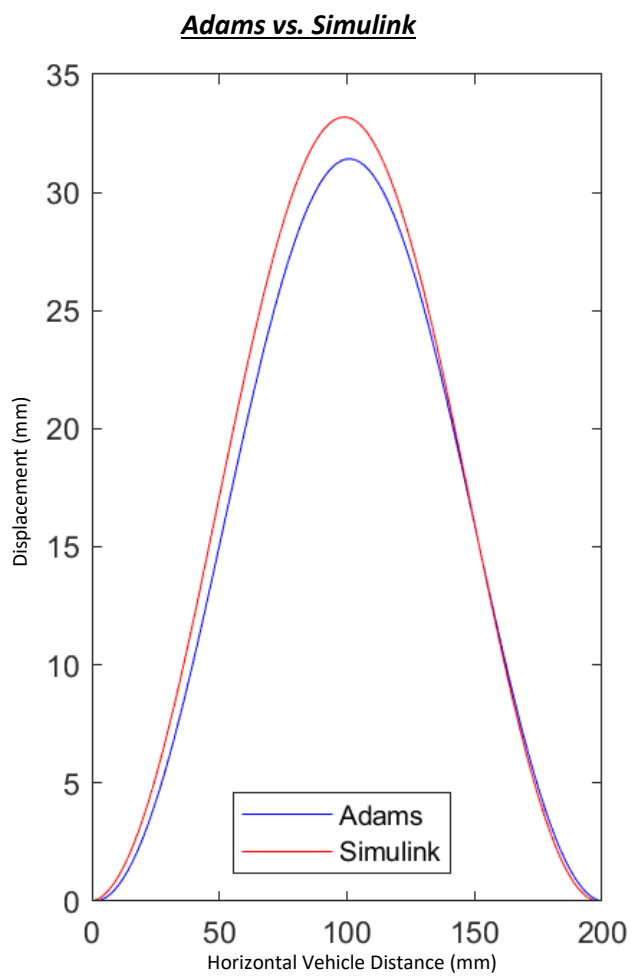


Figure 17: Adams vs. Simulink (SimScape) - Damper Displacement

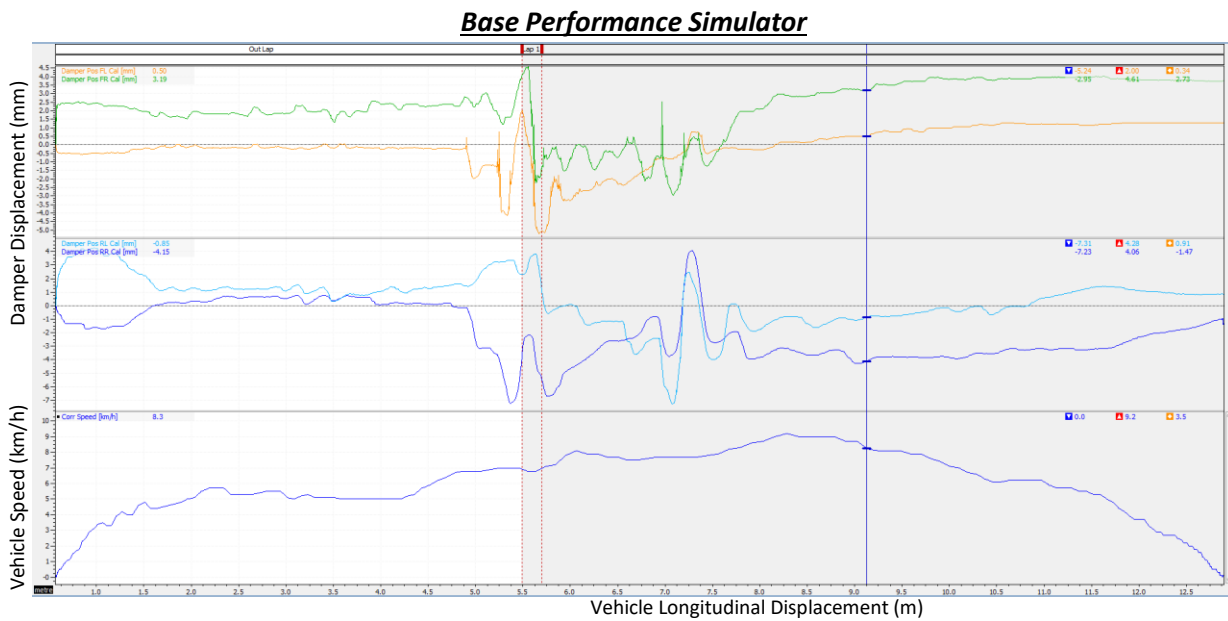


Figure 18: Real World - Cos Bump Results

In all the simulations previously run, the damper more or less followed a $1-\cos(x)$ shaped curve with an amplitude different to the input but with the same frequency. This was assumed to be correct however, the graph above confirms that assumption. The first, red, vertical, dotted line in Figure 18 shows the point where the front left wheel hits the apex for the bump. The yellow curve, which displays the displacement of the front left damper, shows a curve following a similar trend to the input. Furthermore, the light blue line, following the displacement of the rear left damper, shows its apex slightly later, indicating the point this wheel hits the apex of the bump.

This data also shows the importance of taking the entire vehicle, and not just an isolated section, into consideration. It shows that the left wheels, moving up as they move over the bump, are also causing the right-hand side wheels to move up. This is due to the weight transfer of the chassis, which is not fixed in the real world. This, as well as a number of other factors, affected how much displacement was seen at the dampers.

As a result of the quarter car model, it was determined that the approach to the full vehicle simulation would have to be adjusted. The Simulink simulation would have to be rethought completely in order to allow inertia to be accounted for. The results suggested that the Adams simulation might be the most reliable, as the trends correlated to what was expected and discrepancies in results were determined to be caused by issues in the other simulations. Finally, SOLIDWORKS Motion Sim proved not to be a useful tool to continue using. Although results appeared to follow expected trends, a fundamental flaw in the software, namely its lack of a driver model, means that conducting track-based analysis would not be possible.

4.4 Half Car

Due to the geometry of the MCR, the chassis must still remain at a fixed height compared to the road surface, as a free-floating geometry would not work. As a result, and since there are no anti-roll bars in the models, the motion of the individual corners was found to be no different between the half and quarter car models.

4.5 Full Car

The full car simulation finally allows the chassis to float freely, allowing its mass to affect the behaviour of the car and the suspension geometry.

4.5.1 Adams

The first step in creating the Adams model, was choosing the correct assembly to use. The *FSAE_2018* database contains an assembly named *fsae_full_vehicle* which allows the use of the same subsystems used for the quarter-car. This meant that the hardpoints used previously did not need to be input again. Once the hardpoints for the rear suspension components had been defined, further vehicle constants had to be defined for the rear of the vehicle before it was possible to run the first successful simulation.

The next issue only became apparent once the first attempt at running a full vehicle simulation was undertaken. All tests run on the quarter and half car models, had not required any steering input. With the full-car model, steering inputs are defined by Adams' driver model. Due to the changes made to the geometry when updating the hard points, the steering ratio should have gone from a positive to a negative ratio. Not doing this caused the wheels to move in the opposite direction to what the Adams driver was expecting, making closed-loop corrections impossible, resulting in the driver not being able to keep the vehicle on the pre-defined track.

Once this was corrected, a wide range of analysis could be run to investigate different areas of the model. Since, moving forward, the aim of the investigation was to characterise the vehicles suspension on a given track, the course events section of the *Full-Vehicle Analysis* was looked at and the *Double Lane Change* event chosen. This simulation would be easily comparable to the double lane change manoeuvres simulated in Simulink and that would later be conducted on the real-world vehicle.

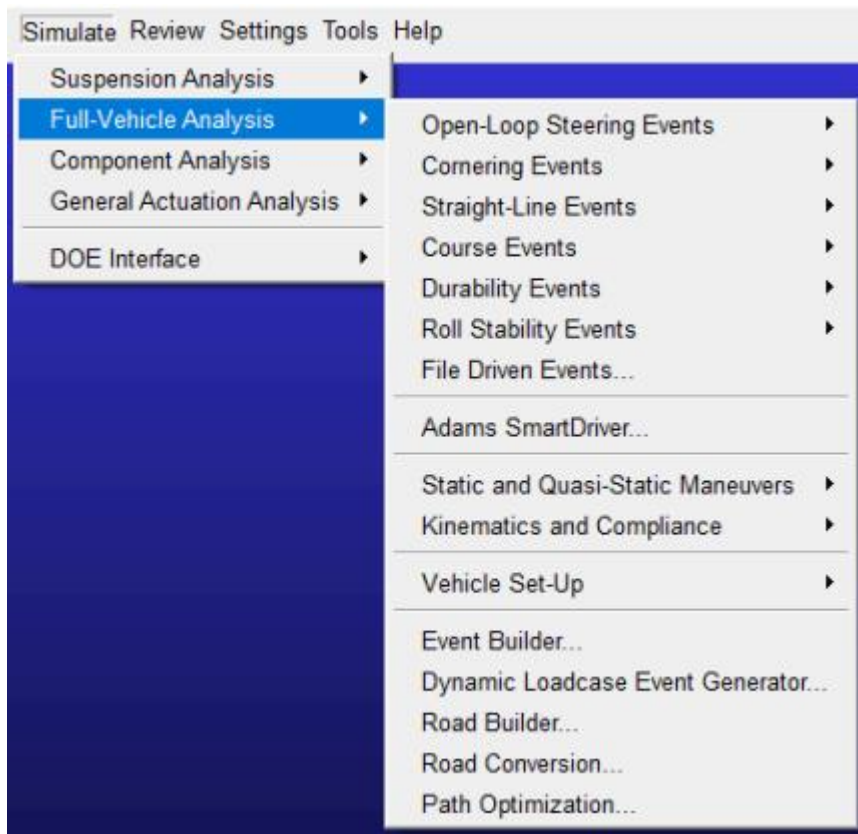


Figure 19: Available analyses

4.5.2 Simulink

The development of the fixed chassis model to the floating chassis model proved to cause some issues. The fundamental issue with SimScape is the way in which the blocks interact with each other. The blocks used up to this point, *translational* blocks (*Simscape Documentation - MathWorks United Kingdom*, no date) only transfer translational motion in 1 direction.

The first alternative that was attempted was based on the original quarter-car SimScape model. The quarter-car was replicated four times, one per corner of the car, before being connected along the front and rear axles separately into two half car models. To mimic the weight transfer from left to right, each corner was connected to a lever. The ratio of the lever, left to right, was set to the same ratio as the lateral location of the centre of mass. The ground reference from both the front and rear levers were connected to a single mass, whose value was set to the sprung mass of the vehicle. Furthermore, the two left and two right corners were connected to longitudinal levers, again with the lever ratio being the same as the longitudinal location of the centre of mass. The ground reference was connected to the same sprung mass used previously.

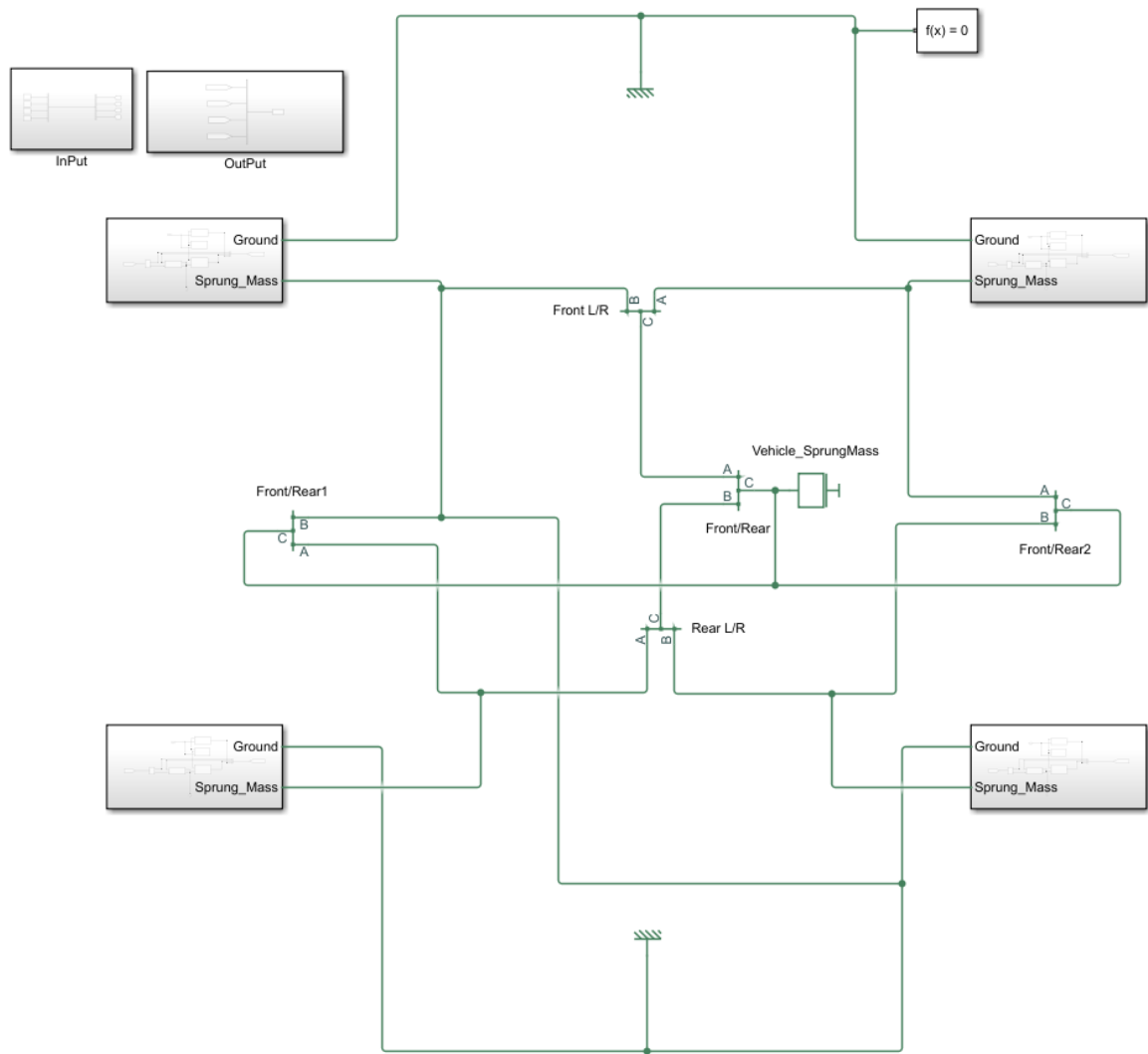


Figure 20: SimScape Full Car Model

This model has a number of flaws resulting in it not being used. The first issue was that this model only output the vertical motion of the sprung mass. Although values for pitch and roll could be calculated by looking at the change in angle of each of the levers, this solution would not resolve the next issue. Due to the 1D nature of the blocks, although momentum and linear inertia do effect results, they are not able to take into account the rotational inertia of the sprung mass. Although this could be calculated, feeding the results back would not be possible. As a result, it was determined that this model had too many sources of inaccuracy.

It was decided that the standard Simulink library had to be used in order to implement an approach based on the basic equations of motion, in particular Equation 1, derived from Newton's 2nd law of motion (Watkins, 1997).

$$F = m \cdot a$$

Equation 1: Newton's 2nd Law of Motion Equation

In order to derive the equations for use in this 7 degree of freedom (7 DoF) model, first a 1 DoF model would be created, which could then be developed into a 7 DoF model.

The forces acting in the 1DoF model, are the forces due to the acceleration of the mass, m , and the forces due to spring and the damper. This way Newton's law of conservation of energy is preserved (Schulz, 2020).

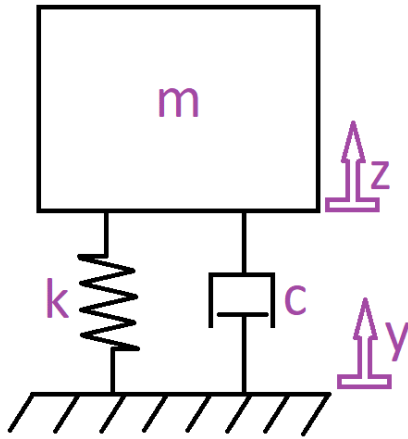


Figure 21: 1DoF Model

Using Hooke's law, the force caused by the spring is calculated by the spring rate multiplied by its change in length (*How to Calculate a Spring Constant Using Hooke's Law - dummies*, no date). The magnitude of displacement is calculated by the change in displacement of the ground (y) and the mass' displacement (z). The force due to the damper is calculated multiplying the damping rate by the difference in speed between the ground (y) and the mass (z). Finally, the acceleration of the mass is the same as the second derivative of its displacement (z). This results in:

$$-m \ddot{z} = k(z - y) + c(\dot{z} - \dot{y})$$

Equation 2: 1DoF Forces

Developing this into a 2DoF model requires the model to be split into two sections. The first section considers everything above the un-sprung mass (m_u). This equation is almost identical to the 1DoF equation, the only difference being that instead of the ground, the displacement of the un-sprung mass is the second displacement.

$$-m \ddot{z} = k(z - z_u) + c(\dot{z} - \dot{z}_u)$$

Equation 3: 2DoF Forces - Sprung

Next, the motion of the un-sprung mass itself is calculated. Both the springs and dampers, above and below the un-sprung mass, must be considered. The forces they apply are in opposite directions to each other and each has its own rate.

$$-m \ddot{z}_u = k_u(z_u - y) + c(\dot{z}_u - \dot{y}) - [k(z - z_u) + c(\dot{z} - \dot{z}_u)]$$

Equation 4: 2DoF Forces - Un-Sprung

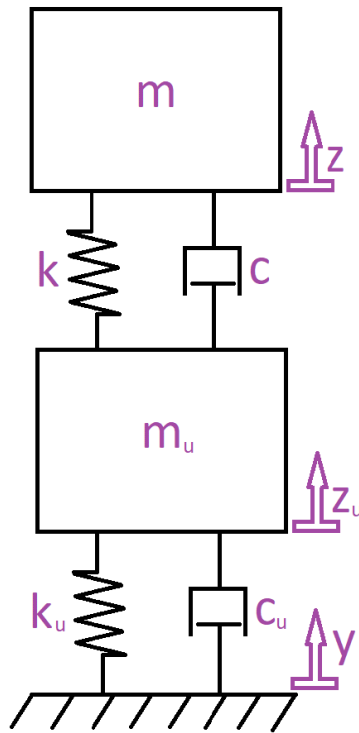


Figure 22: 2DoF Model

The 4DoF model, requires 2 individual 2DoF models side by side, sharing a sprung mass. As well as the vertical displacement of each mass, the rotation of the sprung mass must be taken into account when calculating the forces of the springs and dampers. As the set of equations below show, this is done by multiplying the distance between the sprung/damper and the centre of mass, by the sine of the angle of rotation. Furthermore, as well as differentiating sprung and un-sprung masses, components left to right must also be differentiated from each other. Since this model could be used to represent a vehicle's longitudinal motion, components are labelled as f or r.

$$\begin{aligned} -m \ddot{z} = & k_f(z - z_f - [a \sin \theta]) + k_r(z - z_r + b[b \sin \theta]) \\ & + c_f(\dot{z} - \dot{z}_f - [a \sin \dot{\theta}]) + c_r(\dot{z} - \dot{z}_r - [b \sin \dot{\theta}])) \end{aligned}$$

Equation 5: 4DoF – Sprung

$$-m_f \ddot{z}_f = k_{uf}(z_f - y_f) - k_f(z - z_f - [a \sin \theta]) + c_{uf}(\dot{z}_f - \dot{y}_f) - c_f(\dot{z} - \dot{z}_f - [a \sin \theta])$$

Equation 6: 4DoF - Un-Sprung Front

$$-m_r \ddot{z}_r = k_{ur}(z_r - y_r) - k_r(z - z_r + [b \sin \theta]) + c_{ur}(\dot{z}_r - \dot{y}_r) - c_r(\dot{z} - \dot{z}_r + [b \sin \theta])$$

Equation 7: 4Dof - Un-Sprung Rear

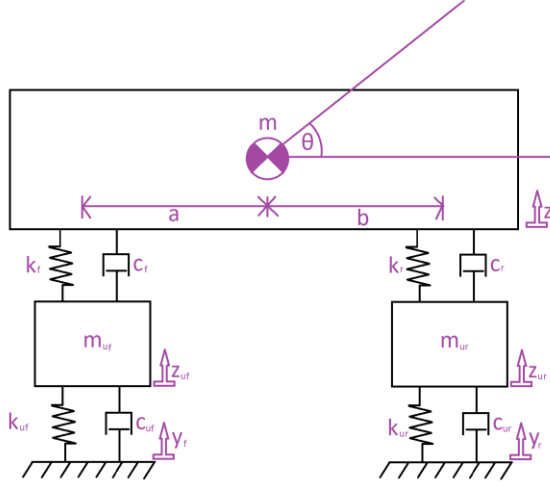


Figure 23: 4DoF Model

By rearranging the equation below, the pitching inertia of this 4DoF model can be calculated. The equation is based on the principle that torque is equal to rotational inertia multiplied by angular acceleration, and that torque is equal to force multiplied by distance (*Torque Formula: Moment of Inertia and Angular Acceleration*, no date). The rotational inertia is constant for a geometry and can be calculated using a range of methods. SOLIDWORKS offers to calculate the rotational inertia for the user, using the given geometry and material density (2020 *SOLIDWORKS Help*, 2020), so this option was used. The forces are supplied by the springs and dampers acting on the sprung mass. Using this, the rotational acceleration, or second derivative of change in angle, can be found.

$$\tau = F \times x = I \times \alpha$$

Equation 8: Torque Equations

$$-I \ddot{\theta} = -ak_f(z - z_f - [a \sin \theta]) + bk_r(z - z_r + [b \sin \theta]) - ac_f(\dot{z} - \dot{z}_f - [a \sin \theta]) + bc_r(\dot{z} - \dot{z}_r + [b \sin \theta])$$

Equation 9: 4DoF - Rotational Inertia

Finally, the 4DoF model can be developed into a 7DoF model which can be used to approximate the behaviour of a full vehicle. Adding a second axis of rotation, to simulate roll, and two further un-sprung masses, makes the model 3 dimensional. A major difference is the addition of anti-roll bars (ARB) to these equations. These are modelled as springs, whose force can be calculated by multiplying its spring rate, by the difference in height between the two un-sprung masses it is connecting. In the case of the MCR, there is one ARB at the front and another at the rear, each connecting un-sprung masses left to right. There are no longitudinal ARBs. For the purposes of comparing the Simulink model to other models that do not contain an ARB, the spring rate of the ARB (A) was set to zero.

$$F_a = A \times (z_{uR} - z_{uL})$$

Figure 24: Force due to ARB

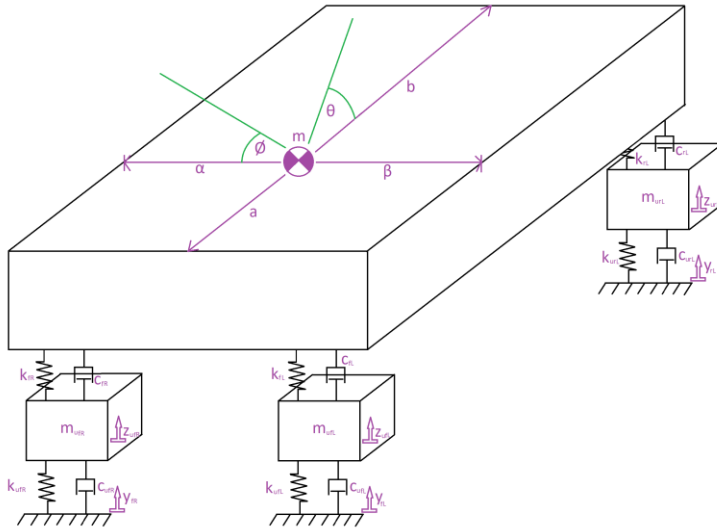


Figure 25: 7DoF Model

$$\begin{aligned} -m\ddot{z} = & k_{fL}(z - z_{fL} - [a \sin \theta] - [\beta \sin \phi]) + c_{fL}(\dot{z} - \dot{z}_{fL} - [a \sin \dot{\theta}] - [\beta \sin \dot{\phi}]) \\ & + [A_f \times (z_{fR} - z_{fL})] + k_{fR}(z - z_{fR} - [a \sin \theta] + [\alpha \sin \phi]) \\ & + c_{fR}(\dot{z} - \dot{z}_{fR} - [a \sin \dot{\theta}] - [\alpha \sin \dot{\phi}]) - [A_f \times (z_{fR} - z_{fL})] \\ & + k_{rL}(z - z_{rL} + [b \sin \theta] - [\beta \sin \phi]) + c_{rL}(\dot{z} - \dot{z}_{rL} + [b \sin \dot{\theta}] - [\beta \sin \dot{\phi}]) \\ & + [A_r \times (z_{rR} - z_{rL})] + k_{rR}(z - z_{rR} + [b \sin \theta] + [\alpha \sin \phi]) \\ & + c_{rR}(\dot{z} - \dot{z}_{rR} + [b \sin \dot{\theta}] + [\alpha \sin \dot{\phi}]) - [A_r \times (z_{rR} - z_{rL})] \end{aligned}$$

Equation 10: 7DoF - Sprung Mass

$$-m_{fL} \ddot{z}_{fL} = k_{ufL}(z_{fL} - y_{fL}) - k_{fL}(z - z_{fL} - [a \sin \theta] - [\beta \sin \phi]) + c_{ufL}(\dot{z}_{fL} + \dot{y}_{fL}) - c_{fL}(\dot{z} - \dot{z}_{fL} - [a \sin \dot{\theta}] - [\beta \sin \dot{\phi}]) - A_f \times (z_{fR} - z_{fL})$$

Equation 11: 7DoF - Un-Sprung Mass – Front Left

$$-m_{fR} \ddot{z}_{fR} = k_{ufR}(z_{fR} - y_{fR}) - k_{fR}(z - z_{fR} - [a \sin \theta] - [\beta \sin \phi]) + c_{ufR}(\dot{z}_{fR} + \dot{y}_{fR}) - c_{fR}(\dot{z} - \dot{z}_{fR} - [a \sin \dot{\theta}] - [\beta \sin \dot{\phi}]) + A_f \times (z_{fR} - z_{fL})$$

Equation 12: 7DoF - Un-Sprung Mass – Front Right

$$-m_{rL} \ddot{z}_{rL} = k_{urL}(z_{rL} - y_{rL}) - k_{fL}(z - z_{rL} - [a \sin \theta] - [\beta \sin \phi]) + c_{urL}(\dot{z}_{rL} + \dot{y}_{rL}) - c_{rL}(\dot{z} - \dot{z}_{rL} - [a \sin \dot{\theta}] - [\beta \sin \dot{\phi}]) - A_r \times (z_{rR} - z_{rL})$$

Equation 13: 7DoF - Un-Sprung Mass – Rear Left

$$-m_{rR} \ddot{z}_{rR} = k_{urR}(z_{rR} - y_{rR}) - k_{fL}(z - z_{rR} - [a \sin \theta] - [\beta \sin \phi]) + c_{urR}(\dot{z}_{rR} + \dot{y}_{rR}) - c_{rR}(\dot{z} - \dot{z}_{rR} - [a \sin \dot{\theta}] - [\beta \sin \dot{\phi}]) + A_r \times (z_{rR} - z_{rL})$$

Equation 14: 7DoF - Un-Sprung Mass – Rear Right

$$\begin{aligned} -I_p \ddot{\theta} = & bk_{rL}(z - z_{rL} + [b \sin \theta] - [\beta \sin \phi]) + bc_{rL}(\dot{z} - \dot{z}_{rL} + [b \sin \dot{\theta}] - [\beta \sin \dot{\phi}]) \\ & + bk_{rR}(z - z_{rR} + [b \sin \theta] + [\alpha \sin \phi]) + bc_{rR}(\dot{z} - \dot{z}_{rR} + [b \sin \dot{\theta}] + [\alpha \sin \dot{\phi}]) \\ & - ak_{fL}(z - z_{fL} - [a \sin \theta] - [\beta \sin \phi]) - ac_{fL}(\dot{z} - \dot{z}_{fL} - [a \sin \dot{\theta}] - [\beta \sin \dot{\phi}]) \\ & - bk_{fR}(z - z_{fR} - [a \sin \theta] + [\alpha \sin \phi]) - ac_{fR}(\dot{z} - \dot{z}_{fR} - [a \sin \dot{\theta}] + [\alpha \sin \dot{\phi}]) \end{aligned}$$

Equation 15: 7DoF – Pitch

$$\begin{aligned}
-I_R \ddot{\theta} = & \alpha k_{fR} (z - z_{fR} - [a \sin \theta] + [\alpha \sin \phi]) + \alpha c_{fR} (\dot{z} - \dot{z}_{fR} - [a \sin \dot{\theta}] + [\alpha \sin \dot{\phi}]) \\
& + \alpha A_f \times (z_{uR} - z_{uL}) + \alpha k_{rR} (z - z_{rR} + [b \sin \theta] + [\alpha \sin \phi]) \\
& + \alpha c_{rR} (\dot{z} - \dot{z}_{rR} + [b \sin \dot{\theta}] + [\alpha \sin \dot{\phi}]) + \alpha A_f \times (z_{uR} - z_{uL}) \\
& - \beta k_{fL} (z - z_{fL} - [a \sin \theta] - [\beta \sin \phi]) - \beta c_{fL} - \beta A_f \times (z_{uR} - z_{uL}) \\
& - \beta k_{rL} (z - z_{rL} + [b \sin \theta] - [\beta \sin \phi]) - \beta c_{rL} (\dot{z} - \dot{z}_{rL} + [b \sin \dot{\theta}] - [\beta \sin \dot{\phi}]) \\
& - \beta A_r \times (z_{uR} - z_{uL})
\end{aligned}$$

Equation 16: 7DoF – Roll

Using the Simulink environment, each of these equations can be converted into a program. Using a variety of blocks allows each of the functions of the equation to be replicated. Figure 26 shows the basic setup of the Simulink model. Each of the equations for the 7DoF model is represented by one of the subsystems.

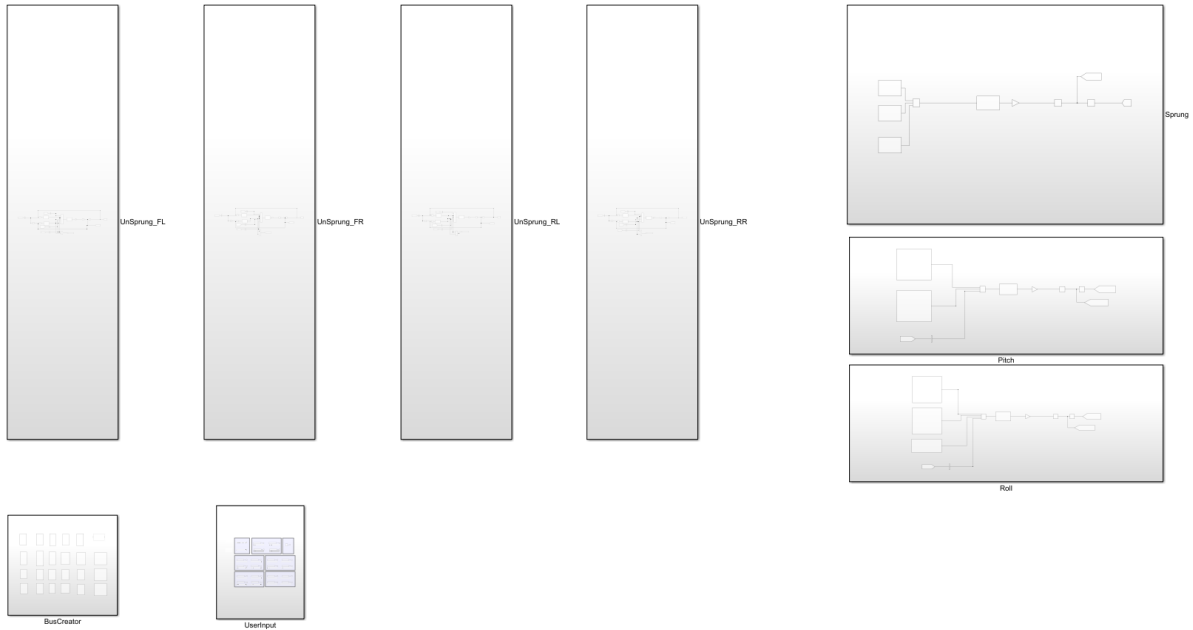


Figure 26: 7DoF full Simulink model

In order to model the un-sprung mass, four subsystems are used to represent each corner. Rearranging the equations given in Equation 11 to Equation 14 make the displacement of the mass (z) the subject, allowing its use as the output of the subsystem.

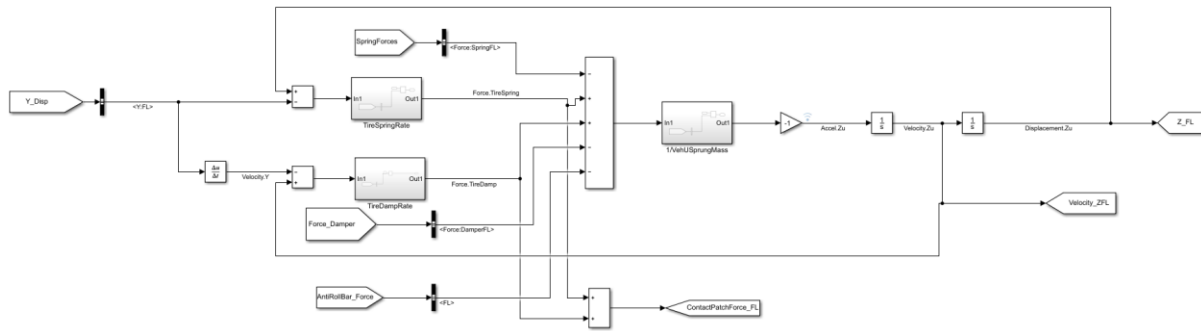


Figure 27: 7DoF Un-Sprung Model

Looking at Figure 27 starting from the left, the first input is the displacement of the road under the corner being looked at. Since this value is used multiple times, it splits off in two direction. Following the top line, the difference between the displacement of the road and the displacement of the un-sprung mass is calculated giving the change in length of the spring. When this is multiplied by the tire spring rate in the next subsystem, the result is the force produced by the spring component of the tire. In the lower line, the tire damper force is calculated, by first differentiating the road displacement (y) and then calculating the difference between that value, and the differential of the displacement of the mass (z). This value is multiplied by the tire damping rate, and fed into the sum block, along with the tire spring force calculated previously. The sum block also shows inputs of the forces from the springs and dampers, which are calculated in the subsystem labelled *Sprung*, as well as the force due to the ARB, which is calculated in the *Bus Creator*. These values are added and subtracted by the *Sum* block before they are divided by the magnitude of the un-sprung mass. The sign is changed, and the values are integrated twice to give the displacement of the un-sprung mass, relative to the ground. For values, such as the displacement of the un-sprung mass, which are required as an input and are also calculated as a result of this subsystem, MatLab uses simultaneous equations to resolve the unknown values.

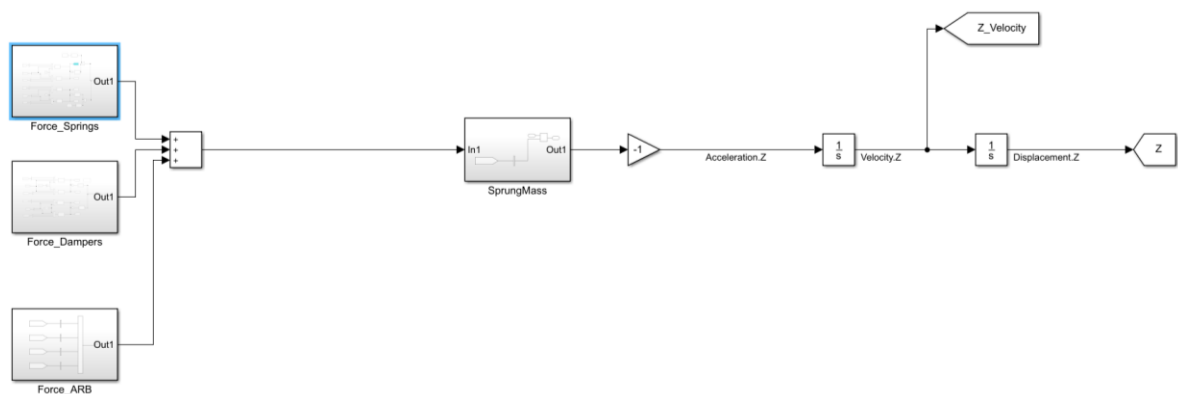


Figure 28: 7DoF Sprung Model

In the first three subsystems of the *Sprung* subsystem, the forces because of the springs, and dampers are calculated. The sum of these values and the ARB forces, divided by the sprung mass of the vehicle, gives the vertical acceleration of the sprung mass. This value integrated twice, give the vertical displacement.

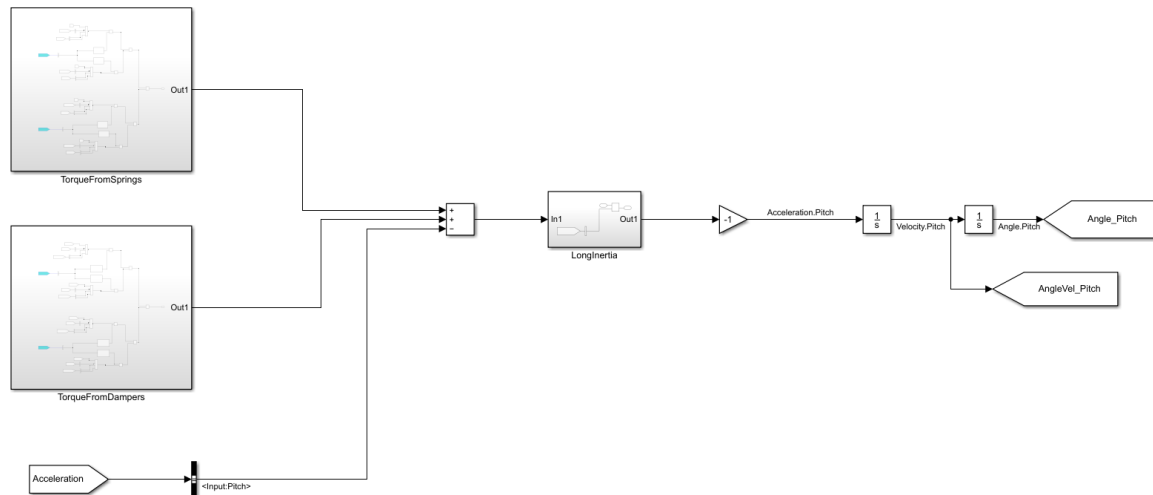


Figure 29: 7DoF Pitch Model

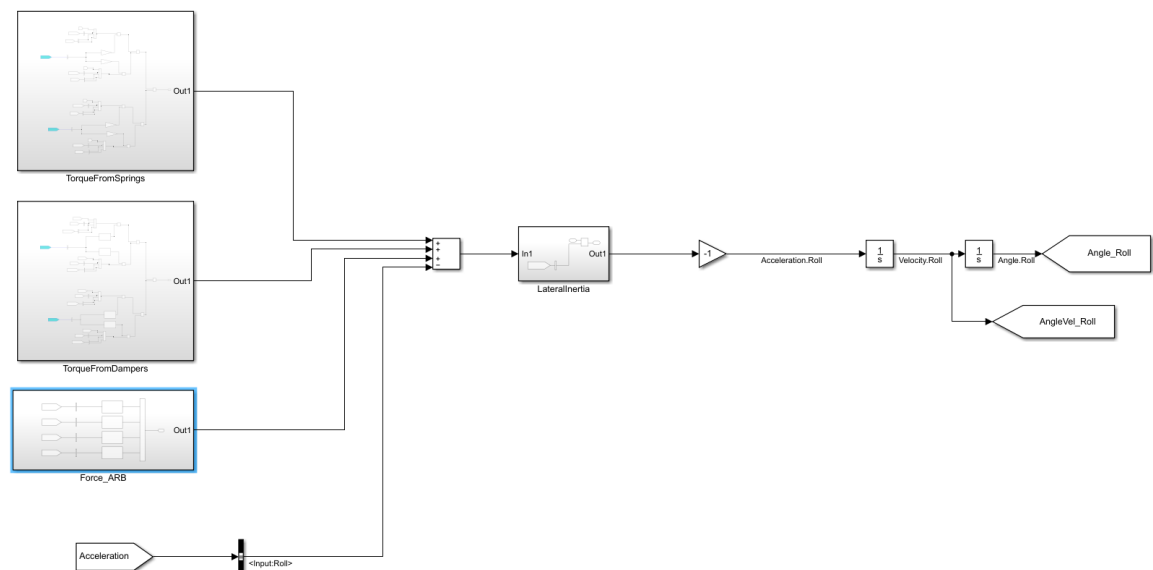


Figure 30: 7DoF Roll Model

In individual subsystems, the individual forces on each spring, damper and ARB are calculated and multiplied by the lever arm. Dividing the sum of these values by the rotational inertia about the given axis and integrating twice, gives the change in angle about that axis.

For the purposes of usability of the model, inputs were placed into easy to reach locations, so that users who are not familiar with Simulink, or the specific model, can easily enter variables and define constants, as well as choosing from a range of scenarios to analyse. The first area where the user can input values is in the *UserInput* subsystem. Here the scenario, spring rates and damping rates can be defined. These values must be updated before the program is run. For the spring and damping rate, this interface gives the user the option to choose from the components that are available for use on the MCR. For the springs, this means choosing a spring rate that relates to a specific part available in the stores. For the dampers, this means choosing a damping rate that relates to the settings available on the installed dampers. Alternatively, custom values can be entered.

The figure displays the Simulink User Interface, which is organized into seven distinct panels for parameter configuration:

- Test Panel:** Contains dropdown menus for 'Vehicle Motion' (set to 'Steady State Cornering') and 'Road' (set to 'Flat').
- Spring Rates (N/m) Panel:** Configures spring rates for four locations: Spring - Front Left (46000), Spring - Front Right (46000), Spring - Rear Left (80000), and Spring - Rear Right (80000). Each has a 'Custom' dropdown and a '-T-' button.
- Test Number Panel:** Displays the 'Test Number' as 4, with a '-T-' button.
- Damp Rates - High Speed Bump (N.s/m) Panel:** Configures high-speed bump damping rates: HSB - Front Left (6000), HSB - Front Right (6000), HSB - Rear Left (2200), and HSB - Rear Right (2200). Each has a 'Custom' dropdown and a '-T-' button.
- Damp Rate - Low Speed Bump Panel:** Configures low-speed bump damping rates: LSB - Front Left (120000), LSB - Front Right (120000), LSB - Rear Left (4000), and LSB - Rear Rear (4000). Each has a 'Custom' dropdown and a '-T-' button.
- Damp Rate - High Speed Rebound Panel:** Configures high-speed rebound damping rates: HSR - Front Left (6000), HSR - Front Right (6000), HSR - Rear Left (4900), and HSR - Rear Left1 (4900). Each has a 'Custom' dropdown and a '-T-' button.
- Damp Rates - Low Speed Rebound Panel:** Configures low-speed rebound damping rates: LSR - Front Left (120000), LSR - Front Right (120000), LSR - Rear Left (4000), and LSR - Rear Right (4000). Each has a 'Custom' dropdown and a '-T-' button.

Figure 31: Simulink User Interface

There are two separate parts of the scenario that are defined by the program's user. The first is the motion of the vehicle. This setting defines the lateral and longitudinal acceleration of the vehicle and the longitudinal velocity. The second setting defines the road surface.

For the purpose of comparing this simulation to the others, at this point the ISO standard double lane change scenario was chosen from the options. To achieve this the lateral acceleration followed a single time period sine wave, followed by a short moment with zero lateral acceleration, followed by a second sine wave. Investigating the lateral acceleration output by BPS when running the double lane change, indicated that this was the ideal input to replicate the manoeuvre in Simulink. The amplitude and frequency of the 2 sine waves, and the gap in between, were determined by the BPS output. Finally, the lateral acceleration was converted from acceleration in G, to m/s^2 and the y axis converted from displacement to simulation time, by assuming a constant velocity of 80kph.

Base Performance Simulator: Double Lane Change – Lateral Acceleration

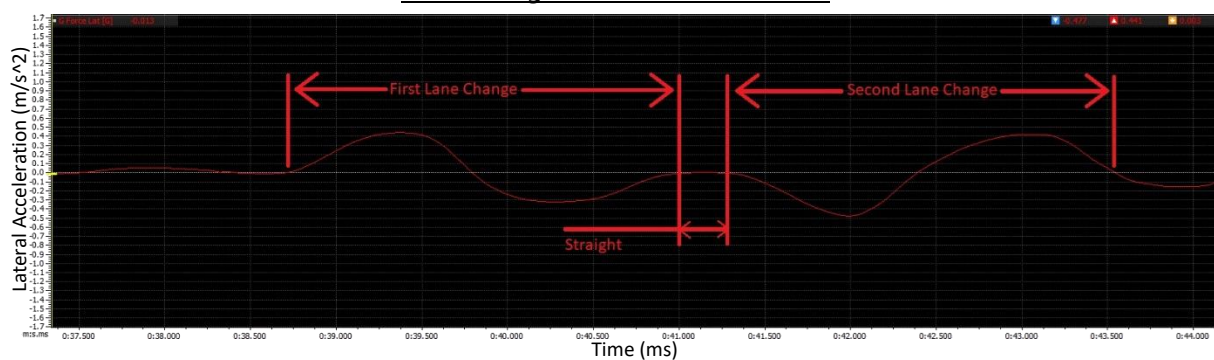


Figure 32: MoTeC Double Lane Change - Lateral Acceleration Output

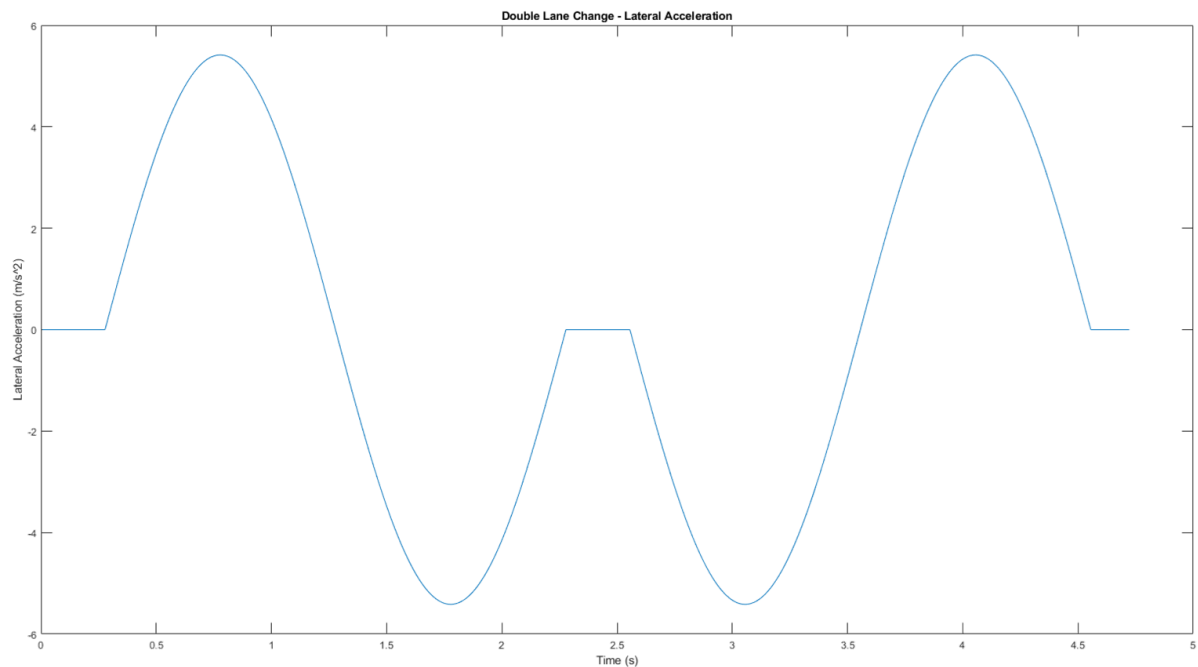


Figure 33: Simulink Double Lane Change - Lateral Acceleration Input

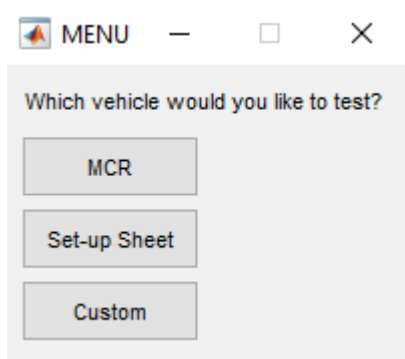


Figure 34: Simulink Vehicle Set-Up options

Once the program is run by the user, the initialisation program, or *Init Function*, is run in the background. This function can be used to initialise variables that are needed to run the program, including reading data from external files. Here, the user will be given the option to choose which vehicle set-up they would like to use. There will be the option of a standard MCR set-up, a set-up sheet, or a custom step by stem input window.

The standard MCR set-up consists of measured constants, identified as part of this investigation. Alternatively, if the user chooses to use the setup sheet, this is defined in an external spreadsheet the user must populate before running the program. Choosing this option will read the data from the spreadsheet and assign each value to a variable in the workspace. Then the variable can be called in the Simulink program. The layout from this set-up sheet was dictated by the research conducted by (Hart, 2020).

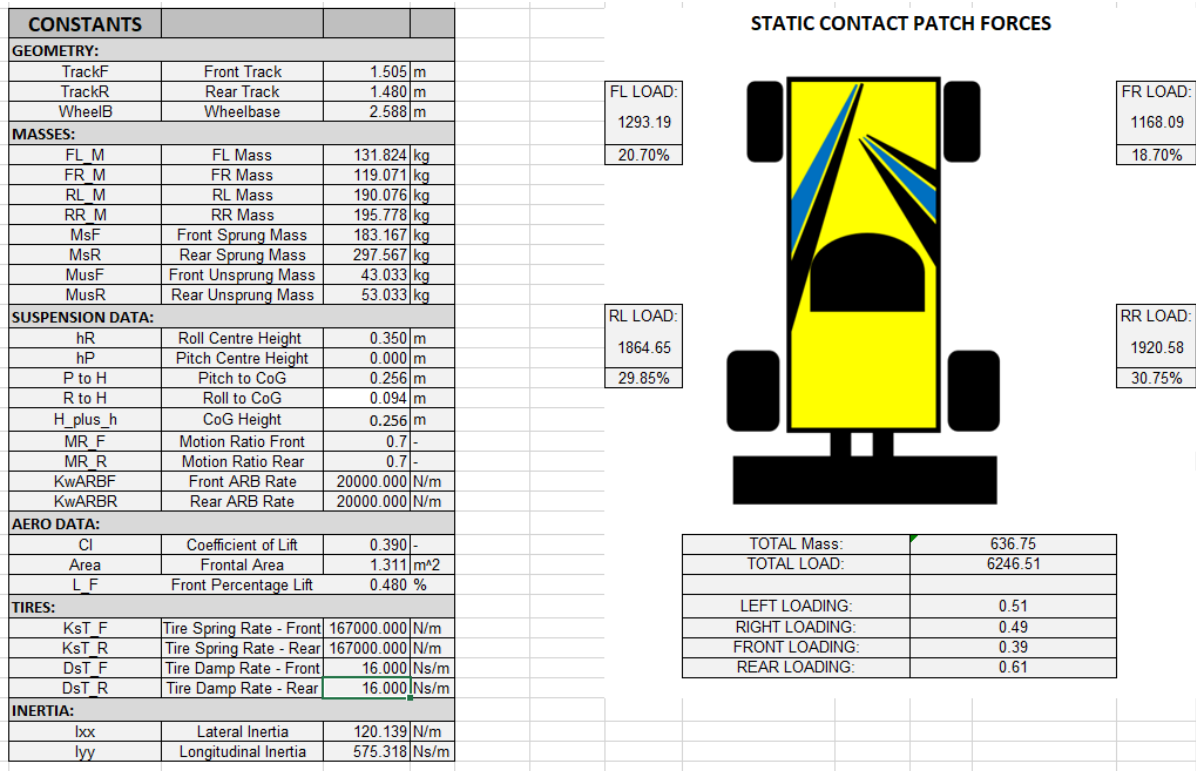


Figure 35: Excel Set-up Sheet

The final option the user has is the *Custom* set-up option. This allows the user to input the values after starting the program. The interface for this can be seen in Figure 36. It has the downside that the input is not saved or recorded in any way, as well as not allowing the user to go back and make changes once the *OK* button has been clicked.

The figure displays five overlapping MATLAB dialog boxes used for customizing a vehicle simulation:

- Vehicle Dimensions:** Includes input fields for Vehicle Length (m), Vehicle Width (m), Longitudinal Inertia, and Lateral Inertia.
- Vehicle Constants (Top):** Includes input fields for Vehicle Sprung Mass (kg), Front Corner Unsprung Mass (kg), and Rear Corner Unsprung Mass (kg).
- Vehicle Constants (Middle):** Includes input fields for CoG Height (m), Roll Axis Height (m), and Pitch Axis Height (m).
- Vehicle Constants (Bottom):** Includes input fields for Mass Distribution Left (%) and Mass Distribution Rear (%).
- Aero:** Includes input fields for Coefficient of Lift (Cl), Front Surface Area (m²), and Front Aero Lift (%).
- Tire Rates:** Includes input fields for Front Tire Spring Rates (N/m), Rear Tire Spring Rates, Front Tire Damp Rates, and Rear Tire Damp Rates.

Figure 36: Custom Set-up input

Once the analysis is completed, the results are output to a file, labelled according to the value input in the *test number* input box on the top right of the *UserInput* subsystem. The values written to this file are predefined by the program and are deemed to include all the necessary values for the program's user to choose from. If data is missing, signals can be logged within Simulink and the *Data Inspector* used to view the results. This has the downside of not being able to save the raw data and all outputs must be chosen before the simulation is run. The frequency at which the data is recorded is dependent on the program being run. Although Simulink allows the programmer to pre-define this value, this can sometimes cause issues when changes are made to the program. As a result, the *variable step size* was used.

4.5.3 Real World

Due to a number of factors, there was no opportunity to conduct real world testing using the MCR. The racing season had to take precedence over this project. As a result, an alternative was found using the universities Base Performance Simulator.

A track, containing the ISO double lane change was modelled using Bobs Track Builder Pro and subsequently loaded as a mod onto the computer containing the BPS installation. The track was placed on an ideal, flat surface with a friction coefficient approximately that of tarmac. The dimensions of the track layout were dictated by ISO 3888, and was aimed to be identical to what was used in the other simulations. The vehicle model used had previously been validated against the real MCR (Noble and Bridgeland, 2018). Finally, the pitlane speed-limiter was implemented to keep the vehicle at the constant entry speed, as required by the standard (ISO3888-1, 2018).

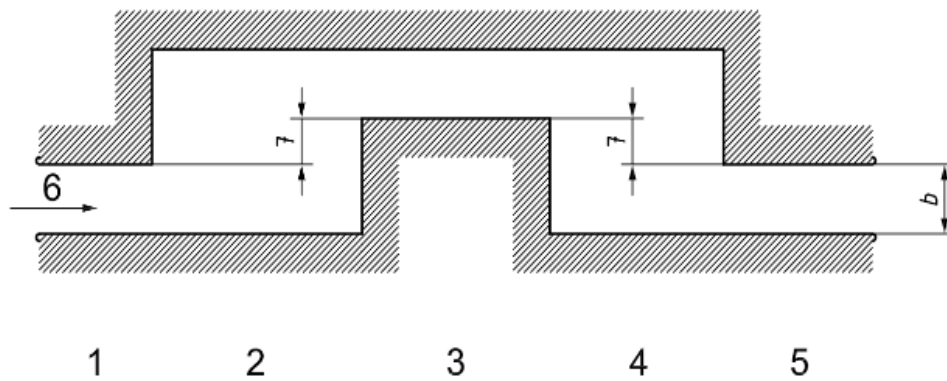


Figure 37: ISO Track Set-up (ISO3888-2, 2011)

In order to record the data, BPS is able to output data channels to files compatible with MoTeC where the data could be evaluated. All available channels are automatically recorded, meaning decisions prior to practical testing are not required.

4.5.4 Results

Throughout this part of the investigation, springs, dampers and ARBs were investigated as they allowed straightforward comparison between set-ups, as well as comparison between simulations and the real world.

The primary method chosen to evaluate the performance of each set-up was by comparing the weight transfer of each run. This can show how much grip the vehicle is likely to lose when lateral acceleration is applied. The weight transfer is calculated using Equation 17 which relies solely on the loads applied through the tires. Alternative ways of calculating weight transfer (see appendix) can be a source of uncertainties.

*Weight Transfer*_{Lateral}

$$= \frac{\text{ContactPatchLoad}_{FR} - \text{ContactPatchLoad}_{FL}}{2} + \frac{\text{ContactPatchLoad}_{RR} - \text{ContactPatchLoad}_{RL}}{2}$$

Equation 17: Contact Patch Weight Transfer

4.5.4.1 Springs

Table 2: Spring Settings

Test Number	Front Spring Rate (N/mm)	Rear Spring Rate (N/mm)
1	26	60
2	46	80
3	125	180

Simulink: Double Lane Change – Weight Transfer

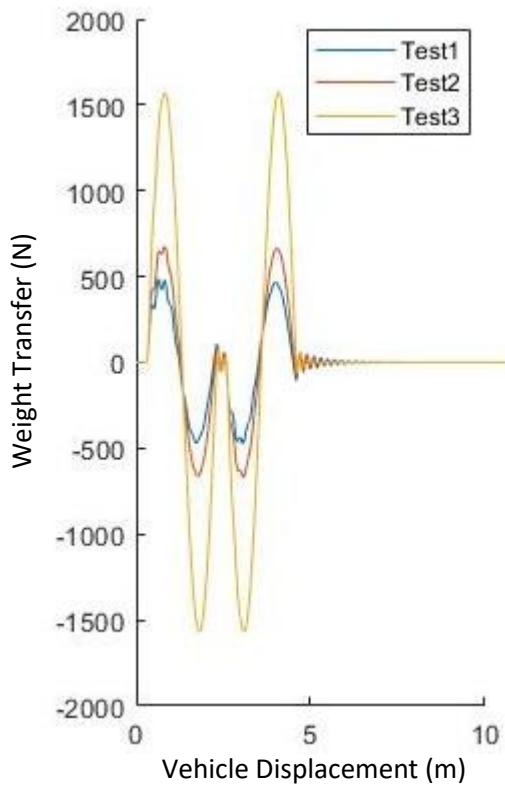


Figure 38: Simulink Double Lane Change – Springs

First, three different spring settings were chosen and compared to each other. The spring rates used can be seen in Table 2. Limited by the rates available for use in BPS, these settings were chosen to represent the lowest, middle and highest spring rates. When considering the equations for weight transfer, it would usually be expected for the overall value to be unaffected by the changes in spring rates. As the elastic weight transfer is directly affected, the geometric weight transfer should change in the opposite direction. Looking only at the Simulink results (Figure 38) this does not appear to be the case. However, comparing these results to the results of the Adams analysis of the same settings, raises a few questions.

The first and most obvious issue, is the fact that within Simulink, despite expectations, the overall weight transfer is highly susceptible to the roll stiffness whilst in Adams the roll stiffness is barely affected. This is most likely down to the fact that within Adams, all three areas of weight transfer, Elastic, Un-Sprung Geometric and Sprung Geometric, are taken into account, whilst Simulink, due to the equations the model is based on, only takes into account the elastic weight transfer. This fact is further supported by the BPS data (Figure 39), showing similar results to those in Adams. This is expected as BPS, similarly to Adams, models all types of WT. Since Simulink is not capable of doing calculating geometric weight transfer, when elastic WT increases, the overall WT increases as well.

BPS: Double Lane Change – Weight Transfer

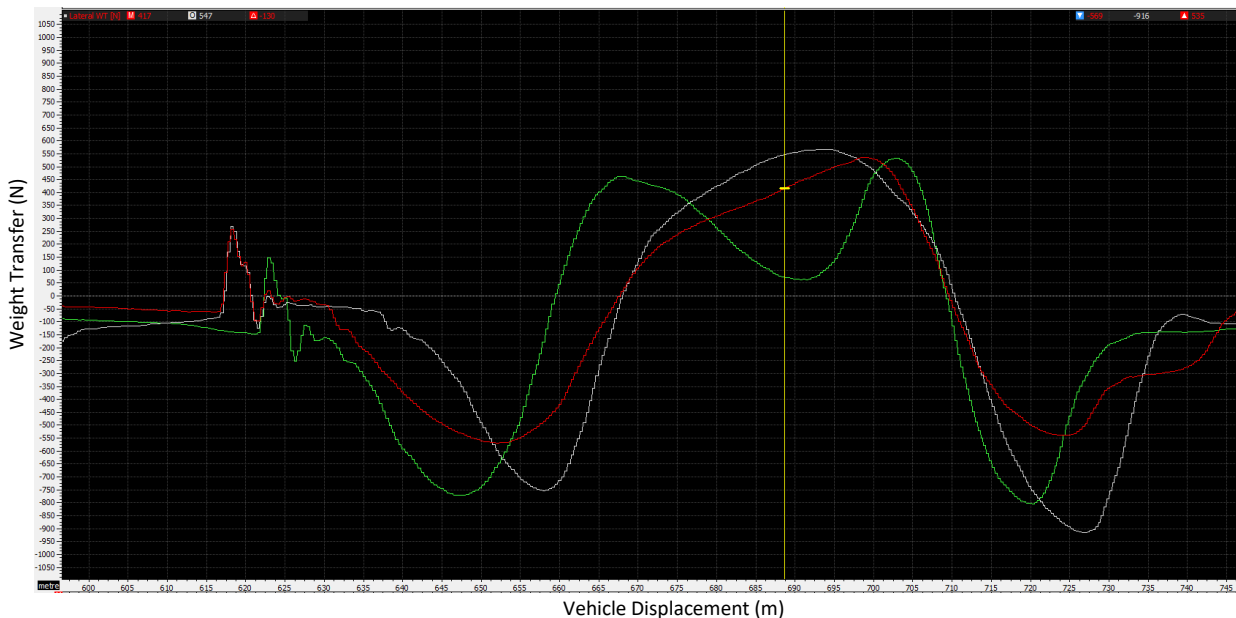


Figure 39: BPS Double Lane Change - Springs

The next issue, evident from comparing the two analysis runs, is the overall trend of the curves. The Simulink analysis results in two Cos shaped curves with the same positive and negative peak. The Adams results on the other hand show that on each lane change, the initial peak, and therefore weight transfer, is higher in magnitude than the second. This is because of the driver model reacting to the scenario in Adams, as opposed to the “ideal” input in Simulink.

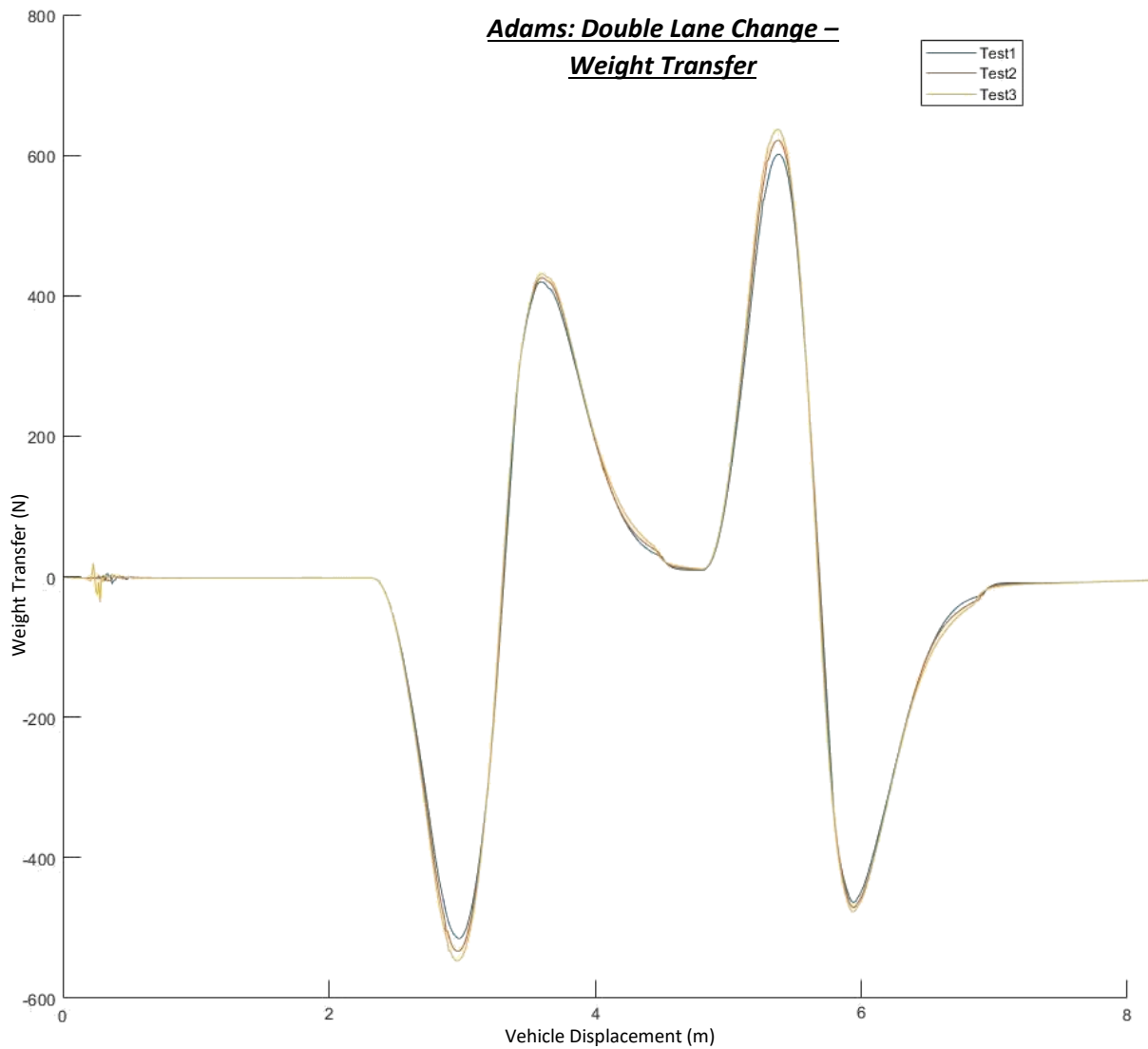


Figure 40: Adams Double Lane Change - Springs

The final problem is the change in magnitude of the maximum weight transfer seen during the manoeuvre. Initially, this difference could be thought to be caused by the driver model used in Adams, potentially producing a less smooth manoeuvre than the “ideal” manoeuvre run in Simulink. However, this would only account for a small difference in the results, whereas these results show a difference of more than a factor of 2. Furthermore, the lateral acceleration between the two simulations is very similar (Figure 64), proving that the driver model is not the cause of these large discrepancies. Since at low spring rates the results are much closer together, it is reasonable to suggest that this difference in results is likely to be as a result of the Simulink model not modelling all aspects of weight transfer. Since the magnitude of the BPS results are much closer to those in Adams, 7.6% difference vs. 279.2% to Simulink, we can conclude that Adams produces a more precise result.

4.5.4.2 Dampers

Next, the dampers were investigated. Looking at the figures below shows the minimal effect the dampers have on the overall behaviour of the vehicle. Figure 41 shows how the dampers effect the rate at which the vehicle settles into a steady state section. This can be seen by looking at the end of the double lane change manoeuvre at around the 5 second mark, where the vehicle continues in a straight line. The lower damping rate shows how the dampers help the vehicle settle more quickly, as well as reducing the underdamped behaviour overall.

Table 3: Damper Settings

Test Number	Front Fast Bump (Ns/mm)	Front Fast Rebound (Ns/mm)	Front Slow Bump (Ns/mm)	Front Slow Rebound (Ns/mm)	Rear Fast Bump (Ns/mm)	Rear Fast Rebound (Ns/mm)	Rear Slow Bump (Ns/mm)	Rear Slow Rebound (Ns/mm)
4	2200	4900	4000	4000	2200	4900	4000	4000
4	6000	6000	120000	120000	6000	6000	120000	120000

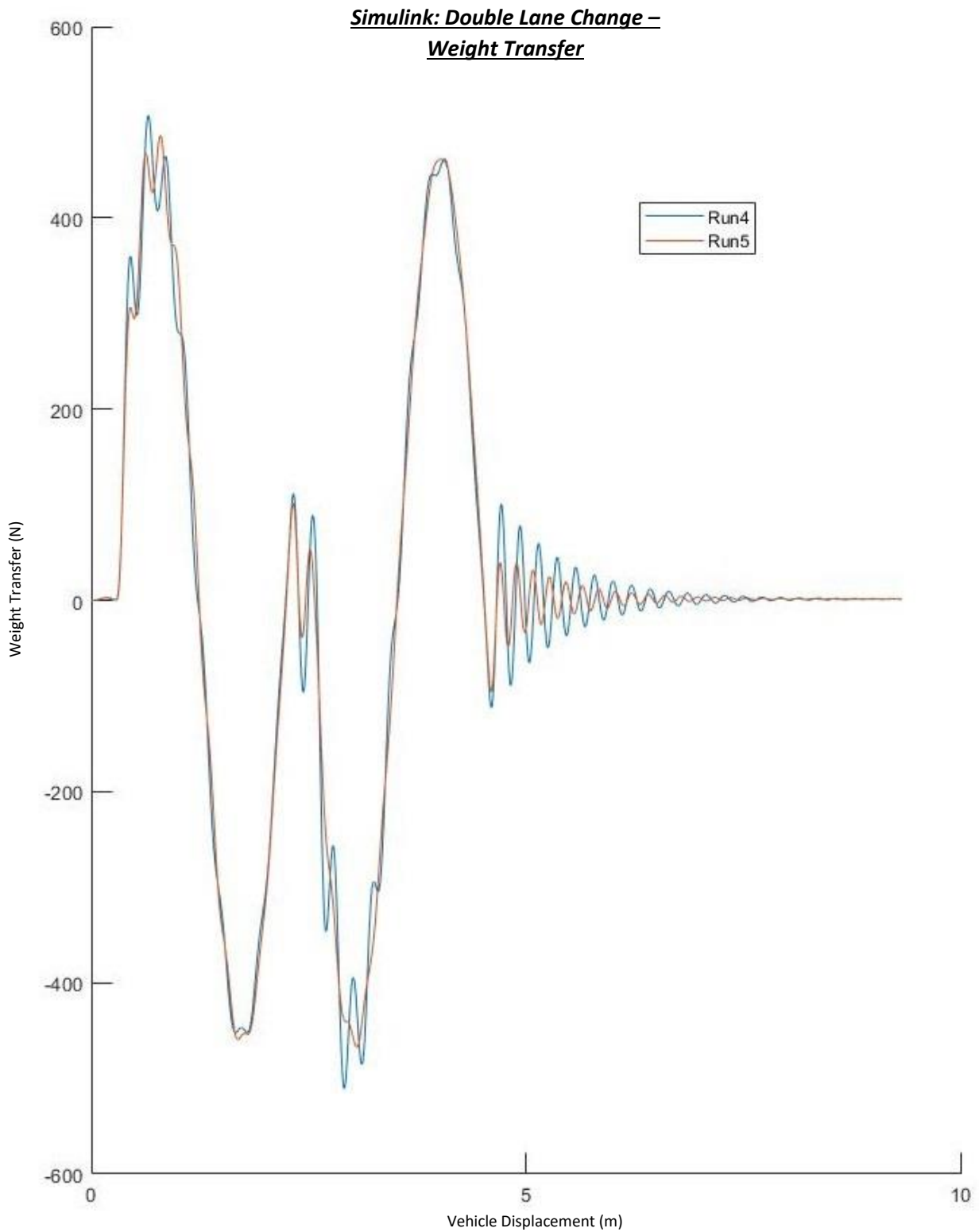


Figure 41: Simulink Double Lane Change – Dampers

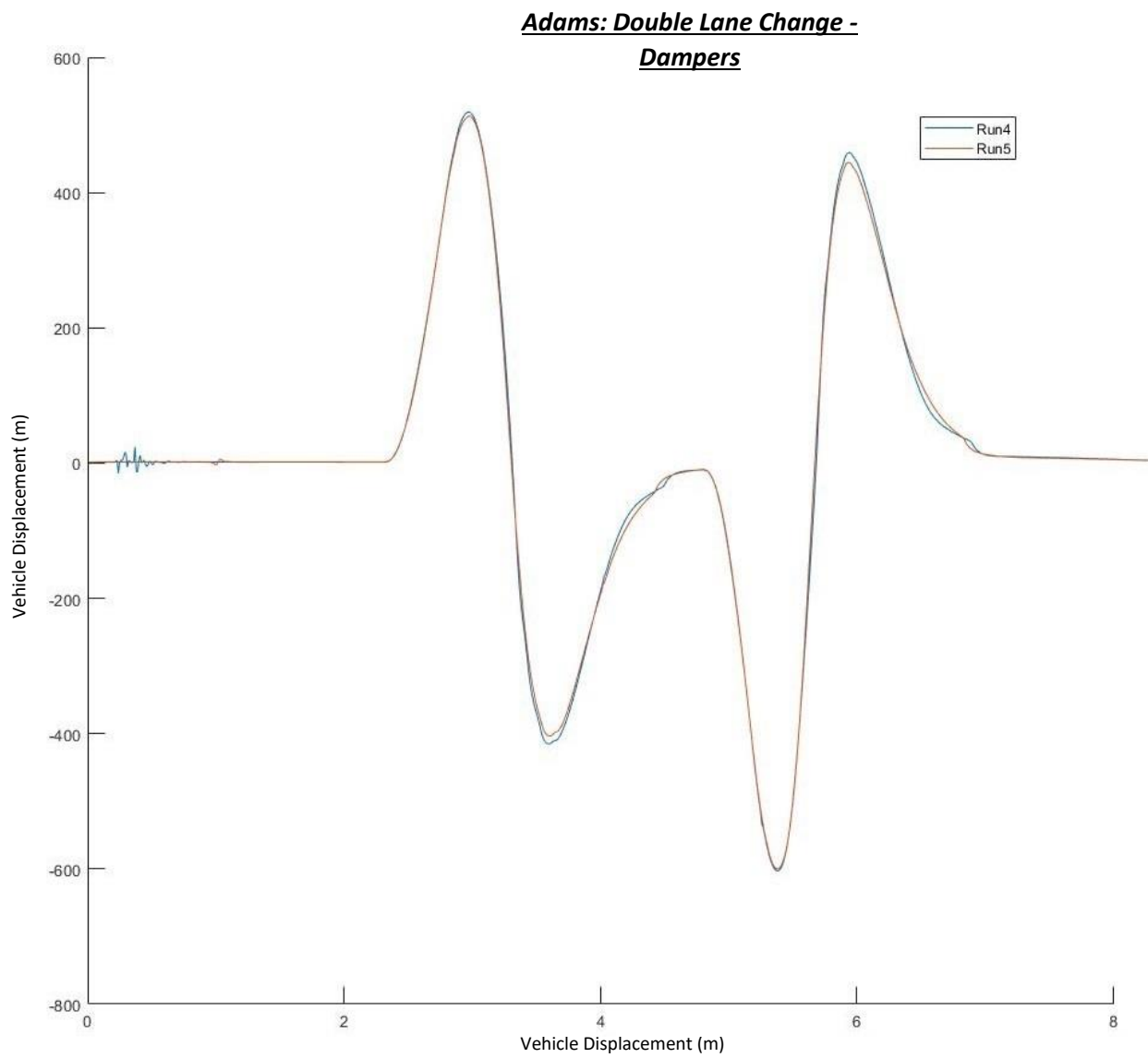


Figure 42: Adams Double Lane Change – Dampers

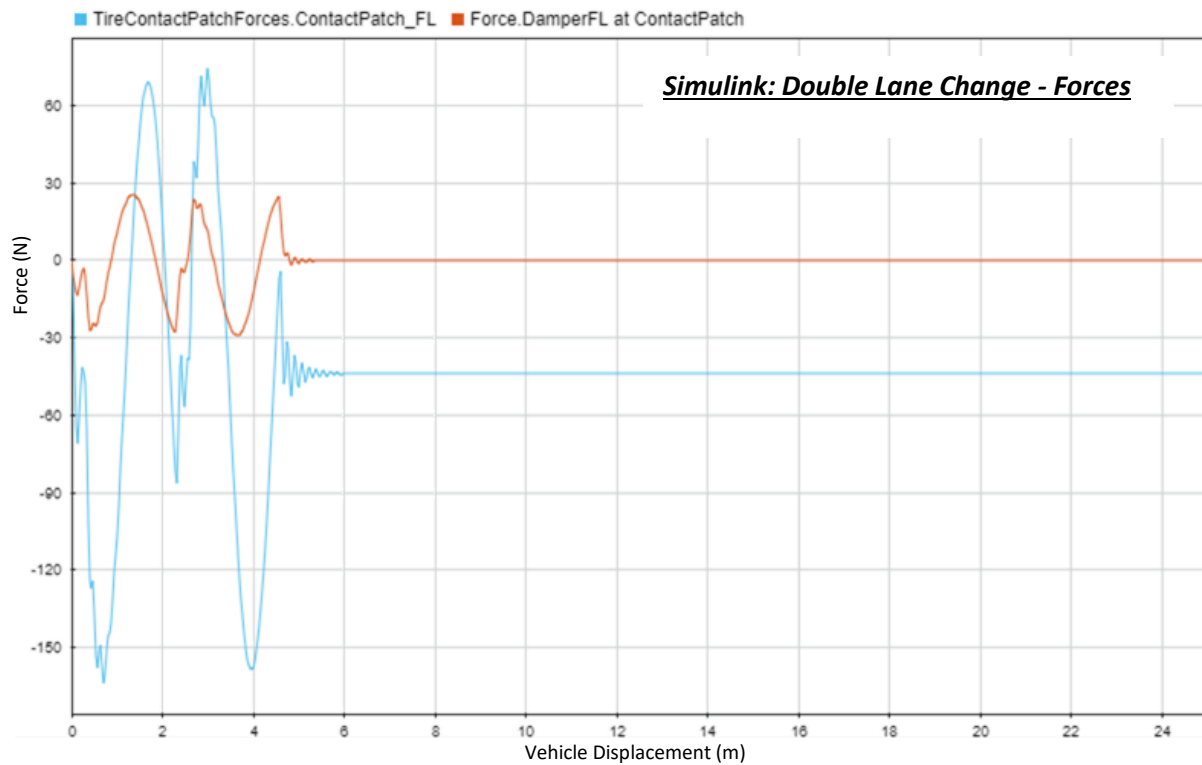


Figure 43: Simulink Double Lane Change - Contact Patch vs. Damper Forces

Whilst the figures above have revealed that the dampers effect on overall weight transfer is minimal, Figure 44 shows that they do effect the roll angle of the car. Since the force of a damper is proportional to its rate of change of length, a higher damping ratio causes the roll of the vehicle to increase at a lower rate. As a result, throughout the given manoeuvre, the roll angle cannot reach its maximum value before the lateral acceleration is reduced.

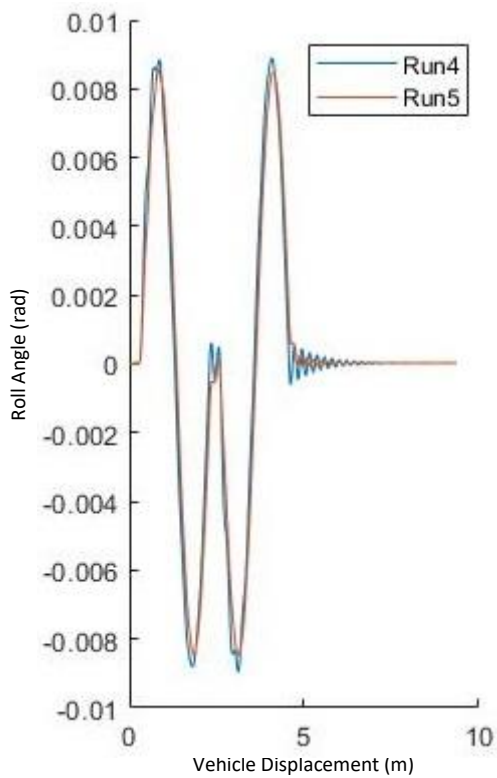


Figure 44: Simulink Double Lane Change - Dampers

Since the Adams and Simulink results agree with each other and the theory, it is logical to conclude that these results are correct. Since BPS contradicts all other evidence, there is likely a source of uncertainty causing inaccurate results. The results below, gained by repeating the test in BPS, shows how the effect of a real driver can make a large difference to the results gained.

BPS: Double Lane Change - Dampers

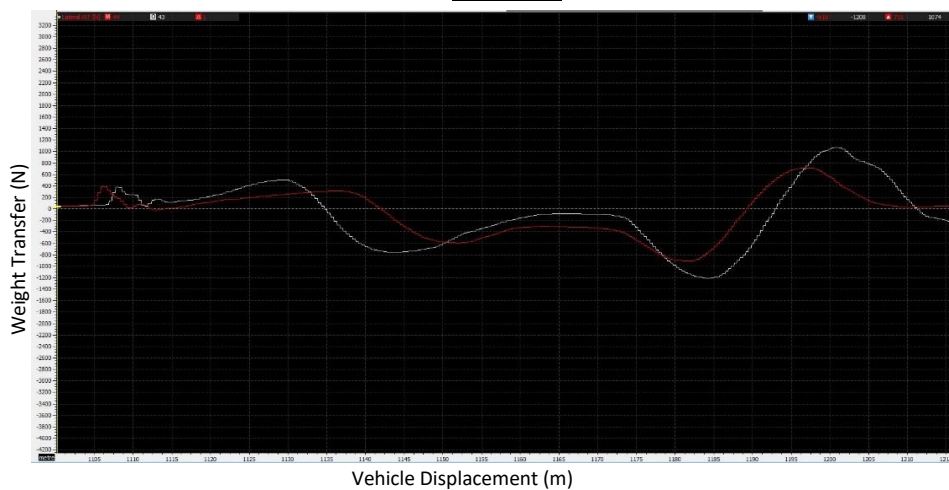


Figure 45: BPS Double Lane Change – Dampers

4.5.4.3 Anti-Roll Bars

Table 4: Anti-Roll Bar Settings

Test Number	ARB Rate Front (N/mm)	ARB Rate Rear (N/mm)
6	60	48
7	84	72
8	60	72
9	84	48

Figure 46 and Figure 47 show how the ARBs effected the vehicle's behaviour over the course of the double lane change manoeuvre. Both the effect on overall weight transfer, and the effect on the roll angle where investigated.

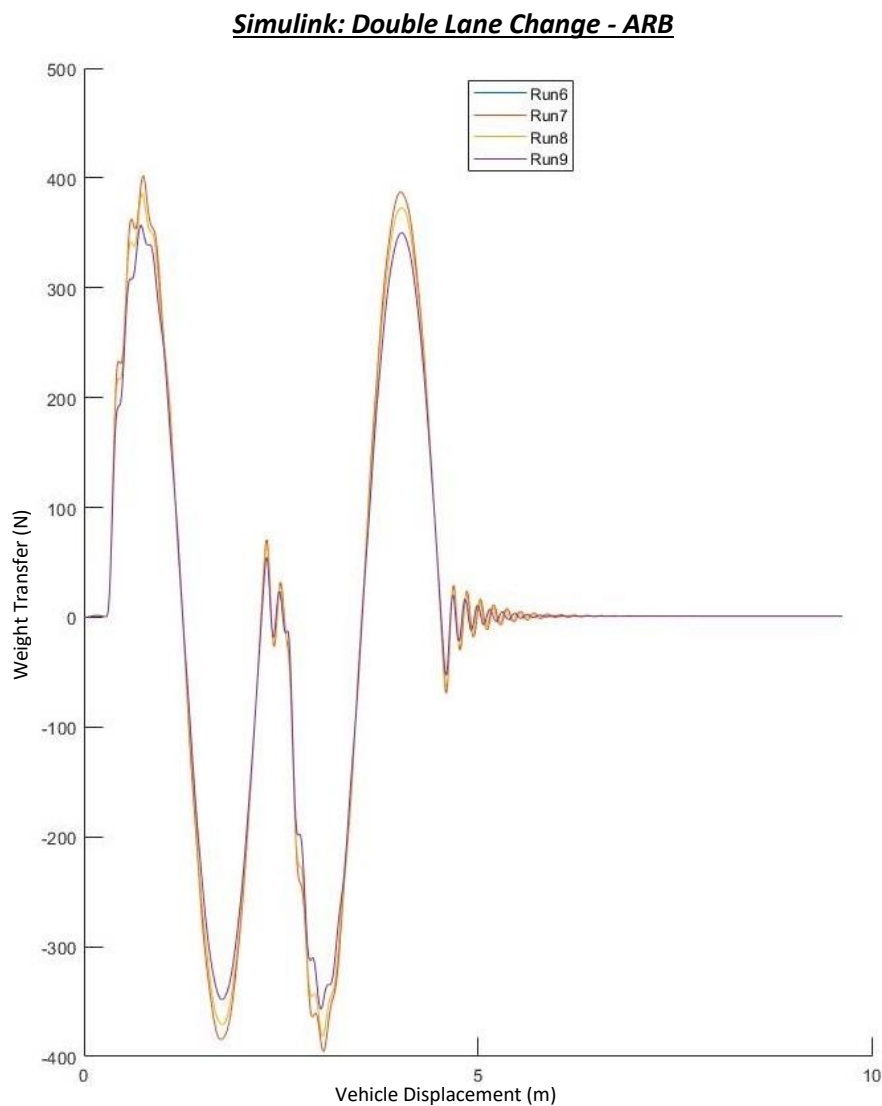


Figure 46: Simulink Double Lane Change - ARB WT

Simulink: Double Lane Change - ARB

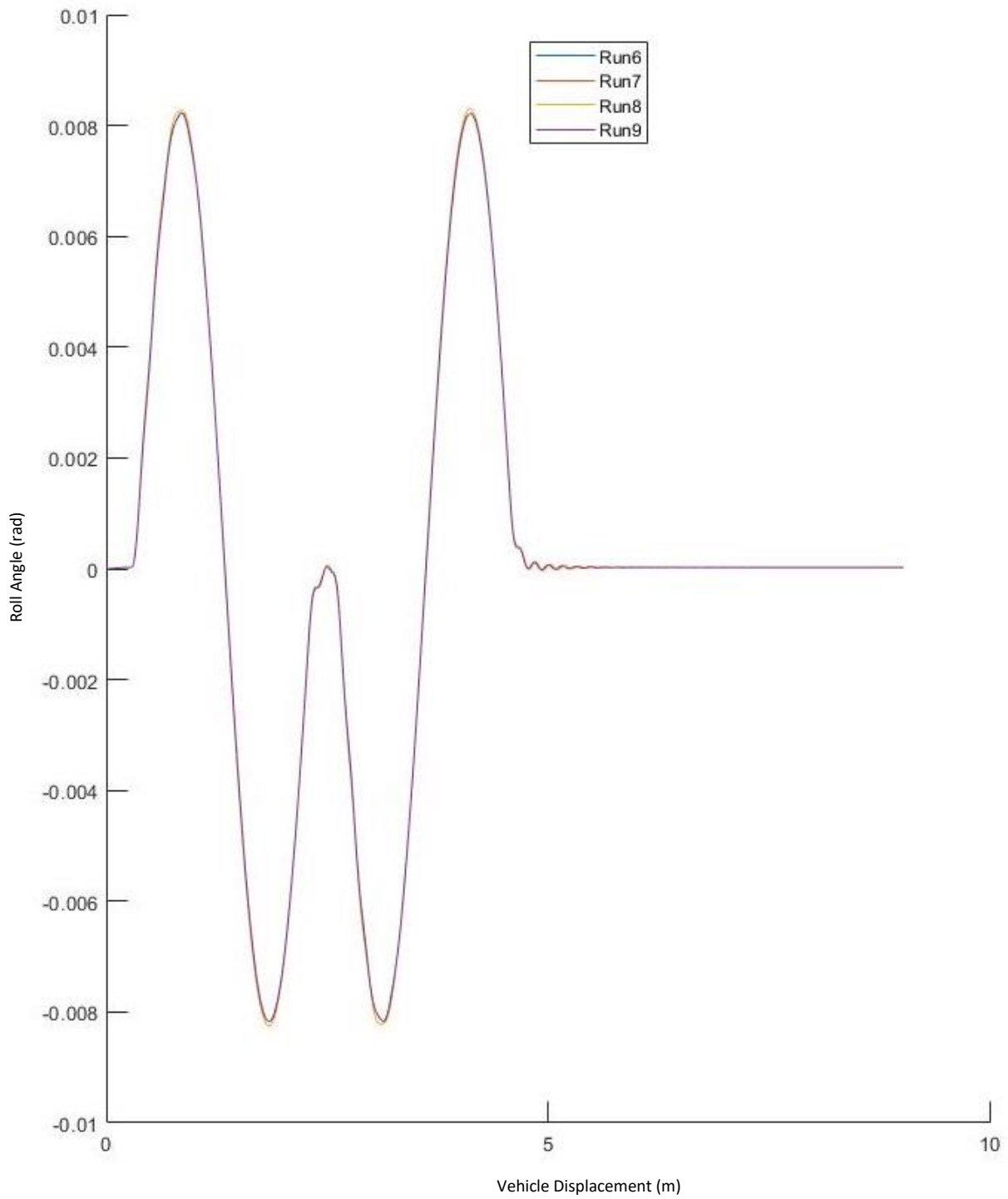


Figure 47: Simulink Double Lane Change - ARB Roll

5 Discussion and Conclusion

Throughout this investigation, two areas were looked at and compared. The primary aim was to compare pros and cons of different types of simulation software in order to determine which one would be optimal for use by UWTSD's MSportEng Team. The secondary aim was to look at the effects of changing the suspension set-up on the Team's MCR vehicle.

5.1 Suspension Evaluation

5.1.1 Springs

As the results show, the springs play a large role in splitting up the weight transfer between its elastic and geometric components. Although springs alone can not have a direct effect on the overall weight transfer of a vehicle, they can help to move the roll stiffness distribution forwards or backwards.

5.1.2 Dampers

The data from the double lane change manoeuvre revealed that the dampers have very little overall effect on the weight transfer of the vehicle. Although they can affect the amount of roll the sprung mass is going to see throughout a manoeuvre, this is solely as a result of them reducing the rate of change of the roll angle, as opposed to reducing the overall maximum roll angle.

5.1.3 Anti-Roll Bars

Opposite to the dampers, the ARBs did help to control the weight transfer distribution. As a result, the ARBs could be used to counter act any over or understeer present in the car to make the car a more neutral handling vehicle. Furthermore, this behaviour could be tuned depending on a driver's preference.

5.2 Simulation Software

SOLIDWORKS, Simulink and MSC/Adams each have their own areas of strength. Each one was developed for a different purpose and was not necessarily intended for use in simulating a vehicle. Having evaluated each of the three pieces of software, Adams/Car would appear to be the best suited for use in this investigation.

SOLIDWORKS was determined to be unsuitable predominantly due to its lack of a driver model. This makes track-based simulations impractical to produce and as a result, suspension set-up cannot be evaluated.

Simulink and Adams were both capable of producing double lane change simulations, the results of which could be used to evaluate the suspension. Simulink's drawbacks were primarily the fact that it also does not have a driver model. Although vehicle behaviour could be estimated using inputs from elsewhere, such as inputting lateral acceleration, this information still has to be gathered outside this simulation. Issues specifically with the model built for this investigation are the fact that only elastic weight transfer is modelled, sometimes resulting in misleading results. Furthermore, not all available data is automatically saved after a run, occasionally making it necessary for the user to re-run an analysis to record some data previously missed. However, MatLab has become a piece of essential software within the engineering and automotive industries. Since Simulink is part of the MatLab package, this simulation method is likely to be available to most potential users, without any added cost. As well as evaluating simulations based on weight transfer, understeer is also often used. Throughout this investigation however, this was not possible, as the Simulink model does not show a steering input or lateral motion.

Adams/Car on the other hand, is purpose built. The double lane change manoeuvre was pre-programmed by MSC and the driver model could determine values for steering input, based on trajectory and location on the track, the same way a real-world driver would, but without the inconsistency. The templates allow a user to quickly produce a basis vehicle model with pre-populated values. Furthermore, all available data, such as lateral acceleration or tyre load, is automatically recorded after every successful run. As a result, if previously un-used data is required, the user will not need to re-run an analysis. Understeer based investigations would also have been possible. The downsides to Adams are with the way in which it handles user error. Its user messages can be difficult to understand, meaning that external support is required to resolve issues. Throughout this investigation, this issue caused many delays. Furthermore, although Adams is used extensively throughout the automotive industry, the fact that it has a large number of rivals means that it is unlikely to already be available to a potential user, requiring money to be spent, making it a likely second choice to a cheaper, or even free alternative.

5.3 Uncertainties in results

Errors are introduced into each simulation model differently depending on what information it requires the user to input. Looking at Adams for example, the accuracy of results is dependent on the dimensions of the suspension geometry, component masses, component inertias and spring rates. For this investigation, suspension lengths were measured to within $\pm 0.01\text{m}$, whilst masses could only be measured to within an accuracy of $\pm 0.1\text{kg}$. Every measurement's accuracy was impeded by the precision of the equipment available.

Whilst the same is true for Simulink, further errors were introduced throughout the process of creating the simulation program. The approach used made it necessary to approximate and simplify vehicle behaviours as well as making assumptions on their interactions. An example of this is the model's failure to account for slip on the tires, potentially affecting the amount of lateral acceleration vs. longitudinal acceleration is being applied throughout a given manoeuvre. Furthermore, due to the approach taken, the simulation can only account for the elastic behaviour of the vehicle.

Whilst it might initially appear as though the simulation is making continuous calculations throughout; this is not possible. For the purpose of comparison, throughout this investigation all simulations were set to solve at 0.001s intervals. In between the measured results, only linear interpolation is possible. This offered a good trade-off between sufficiently demonstrating trends and adequate result accuracy for comparison, without draining resources to the point where the investigation could not be completed within the timeframe.

Whilst real world testing does not suffer with issues as a result of frame rate, there are many other sources for uncertainty. As with the real-world testing, measurements that are used in calculations, are only as accurate as the measuring equipment available. Furthermore, tests cannot be run in the same ideal manner as they can in simulation. For example, the double lane change test took place on a perfectly flat surface with a known coefficient of friction (μ). There is no such thing as a perfectly flat surface in real life and the coefficient of friction can vary with temperature or because of wear and tear. Real world testing is far less flexible than simulation, requiring space and a team, as opposed to a single person and a computer. This reduces the amount of data available for comparing and validating simulation models.

6 Pandemic Impact

As a result of the lockdown, caused by the COVID-19 outbreak, progress in some areas of this project was impeded. Access to the university, including all buildings was no longer possible, making data collection from the BPS impossible. Furthermore, the Sports 2000 season was suspended, meaning new data could not be collected from the car. Testing on the university grounds was also no longer possible.

The impact this had on the project, was that simulations could not be progressed from the double lane change manoeuvre to a full track simulation. For the Adams simulation, none of the road data for any of the tracks run by the MCR is available. The road data held by the BPS could have been converted, had access been possible. For the Simulink simulation, lateral and longitudinal acceleration would

have been required as a minimum, as well as road surface data if possible. This could have been collected through BPS or from real world testing.

7 Recommended Further Work

Both the Adams and Simulink simulations could be developed, to make results more precise. Neither simulation currently takes the stiffness of the chassis into consideration. Whilst in Adams this is a simple case of entering the material values, the Simulink model would likely require fundamental changes, potentially making it necessary to build a new program from the ground up. Furthermore, values for chassis stiffness would need to be measured, making a specialised rig necessary.

The results gained in this investigation, only showed trends between a range of set-up changes. Each of the spring, damper and ARB settings should be tested using every available value, in order to determine the optimal value for the MCR.

Whilst the effect of changing suspension constants was investigated on the double lane change, a better way to quantify potential gains would be through a full track simulation. These could be run in Adams to help understand if changes are significant across a race and as a result, worth investing in.

Finally, the simulations could be used in order to investigate the effect of altering components completely. Changing the geometry of the suspension, through new wishbones or uprights for example, could result in significant gains around a track. The simulation could help predict potential improvements, without physical components needing to be manufactured.

8 Bibliography

2020 SOLIDWORKS Help (2020). Available at:

https://help.solidworks.com/2020/english/solidworks/cworks/c_basic_concepts_of_analysis.htm
(Accessed: 26 March 2020).

7-Post Rig Testing Facility (2019). Available at: <http://www.arcindy.com/7-post.html> (Accessed: 16 April 2020).

Adams Car - Real Dynamics for Vehicle Design and Testing (2020). Available at:

<https://www.mscsoftware.com/product/adams-car> (Accessed: 19 April 2020).

Allcott, G. (2014) *How to be a Productivity Ninja*. London: Icon Books Ltd.

An Overview of How to Use ADAMS/Solver Chapter 1 ADAMS/Solver and MSS (no date).

Arto, K. a (1999) 'Management Across the Organisation', *International Journal Project Management*, Vol 5 No 1(ISSN 1455-4186), pp. 1–85. Available at:

<http://eu.wiley.com/WileyCDA/WileyTitle/productCd-EHEP003349.html>.

Blundell, M. and Harty, D. (2015) *The Multibody Systems Approach to Vehicle Dynamics*. Waltham.

Calver, K. (2001) *SUSPENSION - Shock absorbers (dampers), basic knowledge*. Available at:

https://www.minimania.com/SUSPENSION___Shock_absorbers___dampers___basic_knowledge
(Accessed: 14 April 2020).

Caster – Geometry Explained – Suspension Secrets (no date). Available at:

<https://suspensionsecrets.co.uk/caster/> (Accessed: 22 April 2020).

Chang, K.-H. (2019) *Motion Simulation and Mechanical Design*. Stephen Schroff.

Dempsey, M., Fish, G. and Picarelli, A. (2012) 'Using Modelica models for Driver-in-the-loop simulators'. doi: 10.3384/ecp12076571.

Edelstein, S. (2016) *The Complete Guide to Motorsports*. Available at:

<https://www.digitaltrends.com/cars/motorsports-complete-guide/> (Accessed: 19 April 2020).

Engineering Design and Technology Series An Introduction to Motion Analysis Applications with SolidWorks Motion, Student Workbook (1995).

Engineering Teamwork | Ethics Wiki | Fandom (2020). Available at:

https://ethics.wikia.org/wiki/Engineering_Teamwork (Accessed: 18 April 2020).

Essers, U. (1997) *Nutzfahrzeug-Echtzeitsimulation auf Parallelrechnern mit Hardware-in-the Loop*.

Expert Verlag. Available at:

https://books.google.co.uk/books?hl=en&lr=&id=oLU1oE8CrMgC&oi=fnd&pg=PA3&ots=JEh3fP-XWa&sig=FQKtswQ6R-kdPANDfii8YqYURCU&redir_esc=y#v=onepage&q&f=false (Accessed: 19 April 2020).

‘Formula SAE Applications using Adams/Car’ (2018).

Hahn, C. and Marquez-Brunal, E. (2020) *Vehicle Modeling Using Simscape Multibody*. Available at:

<https://uk.mathworks.com/videos/matlab-and-simulink-racing-lounge-vehicle-modeling-part-4-simscape-multibody-1508365817599.html> (Accessed: 19 April 2020).

Hart, M. *et al.* (2019) *Sports 2000 Bodywork package group project*.

Hart, M. (2020) ‘Development of a Vehicle Setup Characterisation Model for the MCR SPORS 2000 Race Car’.

Herrmann, M. (2016) *TORSIONAL STIFFNESS AND NATURAL FREQUENCY ANALYSIS OF A FORMULA SAE VEHICLE CARBON FIBER REINFORCED POLYMER CHASSIS USING FINITE ELEMENT ANALYSIS*.

Home | Base Performance (no date). Available at: <https://www.baseperformance.net/> (Accessed: 19 April 2020).

How to Calculate a Spring Constant Using Hooke’s Law - dummies (no date). Available at:

<https://www.dummies.com/education/science/physics/how-to-calculate-a-spring-constant-using-hookes-law/> (Accessed: 19 April 2020).

INDYCAR Q&A: Shaker Rig 101 (2019). Available at: <https://www.indycar.com/News/2016/01/01-08-Shaker-Rig-101-by-Pruett> (Accessed: 16 April 2020).

Introduction to Simulink with Engineering Applications - Steven T. Karris - Google Books (no date).

Available at:

https://books.google.co.uk/books?hl=en&lr=&id=L2JYZYI_I_wC&oi=fnd&pg=PT15&dq=simulink&ots=9w1mIJ60D3&sig=4Njaq3PiV315zpecWeXBVGn8Fzw&redir_esc=y#v=onepage&q=simulink&f=false (Accessed: 19 April 2020).

ISO - BSI - British Standards Institution (2020). Available at: <https://www.iso.org/member/2064.html> (Accessed: 19 April 2020).

ISO14512, B. (1999) ‘ISO 14512 - Passenger cars-Straight-ahead braking on surfaces with split coefficient of friction-Open-loop test method’, *British standards*.

ISO3888-1, B. (2018) 'ISO 3888-1 - Passenger cars — Test track for a severe lane-change manoeuvre'.

ISO3888-2, B. (2011) 'ISO 3888-2 - Passenger cars — Test track for a severe lane-change manoeuvre Part 2 : Obstacle avoidance'.

ISO4138, B. (2012) 'ISO 4138 - Passenger cars — Steady-state circular driving behaviour — Open-loop test methods'.

ISO7975, B. (2019) 'ISO 7975 - Passenger cars - Braking in a turn - Open-loop test method'.

ISO9816, B. (2018) 'ISO 9816 - Passenger cars - Power-off reaction of a vehicle in a turn - Open-loop test method'.

Lombard, M. (2013) *SolidWorks 2013 Bible*. Wiley. Indianapolis.

Longoria, R. G. (2014) *Modeling and Experimentation: Spring-Mass-Damper System Dynamics*. Austin.

MCR S2 – MCR Race Cars (2020). Available at: <https://mcrracecars.co.uk/mcr-s2/> (Accessed: 18 April 2020).

Miller, A. (2002) 'Testing a Formula SAE racecar on a seven-poster vehicle dynamics simulator', *SAE Technical Papers*, (724). doi: 10.4271/2002-01-3309.

Miller, S. and Wendlandt, J. (no date) *Real-Time Simulation of Physical Systems Using Simscape*.

Morley, A. (2007) 'Motorcycle Suspension Investigation'.

Morse, P. (2016) *Race Car Tuning with Driver-in-the-Loop Simulators*. Available at: <https://www.ansiblemotion.com/automotive-driver-in-the-loop-simulation-articles/race-car-tuning-with-driver-in-the-loop-simulators> (Accessed: 18 April 2020).

MoTeC > C187 > Overview (2020). Available at: <https://www.motec.com.au/c187/c187overview/> (Accessed: 18 April 2020).

Motorsport / rFpro (no date). Available at: <http://www.rfpro.com/motorsport/> (Accessed: 19 April 2020).

MSC Software's Adams Multi-body Dynamics & Motion Analysis software (2020). Available at: https://www.dte.co.uk/msc-adams/?gclid=CjwKCAjw7e_0BRB7EiwAIH-goLFxt4_ufF7JdPLG9DW_jeYeX6JEUpr2yxY7Z4NQ4G_Y8CtzdXwUoBoCgigQAvD_BwE (Accessed: 19 April 2020).

Noble, A. and Bridgeland, J. (2018) *Group Project Report*.

Partridge, C. (2013) 'rFactor and Base Performance Simulators', *Racing GT*. Available at: <https://www.bsimracing.com/rfactor-and-base-performance-simulators-article-by-racing-gt/> (Accessed: 19 April 2020).

Pushrod-Pullrod (2020). Available at: http://www.formula1-dictionary.net/pushrod_pullrod.html (Accessed: 17 April 2020).

Rosenhead, R. (2012) *Where is your project Gateway review? - Ron Rosenhead* Ron Rosenhead. Available at: <https://ronrosenhead.co.uk/4490/where-is-the-project-gateway-review/> (Accessed: 18 April 2020).

Schulz, C. (2020) *Conservation of Energy | HowStuffWorks*. Available at: <https://science.howstuffworks.com/innovation/inventions/newtons-cradle4.htm> (Accessed: 17 April 2020).

Simons, B. (2019) *K&C Testing and SEMA | Vehicle Dynamics*. Available at: <https://www.youtube.com/watch?v=7mpqCCMGDrg> (Accessed: 16 April 2020).

Simscape Documentation - MathWorks United Kingdom (no date). Available at: <https://uk.mathworks.com/help/phymod/simscape/> (Accessed: 19 April 2020).

Simscape Multibody - MATLAB & Simulink (no date). Available at: <https://uk.mathworks.com/products/simmechanics.html> (Accessed: 19 April 2020).

Simulink - Simulation and Model-Based Design - MATLAB & Simulink (no date). Available at: <https://uk.mathworks.com/products/simulink.html> (Accessed: 19 April 2020).

Smith, C. (1978) *Tune To Win*. Aero Publishers Inc.

SOLIDWORKS Help (2018). Available at: https://help.solidworks.com/2018/english/solidworks/motionstudies/hidd_dve_sim_springs.htm (Accessed: 19 April 2020).

Sports 2000 Duratec Championship - SRCC - SPORTS RACING CAR CLUB (2020). Available at: <https://sports2000.co.uk/championships/duratec/> (Accessed: 18 April 2020).

Still, G. K. (2017) *Crowd Safety and Risk Analysis - Simulation Principles*. Available at: <http://www.gkstill.com/CV/PhD/Chapter4.html> (Accessed: 18 April 2020).

Team MCR - University of Wales, Trinity Saint David (no date). Available at: <https://uwtsd.ac.uk/raceteams/team-mcr/> (Accessed: 20 April 2020).

Torque Formula: Moment of Inertia and Angular Acceleration (no date). Available at: https://www.softschools.com/formulas/physics/torque_formula/59/ (Accessed: 19 April 2020).

Tudor, T. and Tudor, K. (2016) 'Analysis of Load Transfer with Respect to Non-Dynamic Steer Effects'.
Using ADAMS/Solver (2002).

Watkins, E. (1997) 'The Laws of Motion from Newton to Kent'.

Wilde, J. R. *et al.* (2005) 'Experimental Evaluation of Fishhook Maneuver Performance of a Kinetic Suspension System', *SAE Transactions*. SAE International, pp. 387–396. doi: 10.2307/44725073.

Ziegler, S. and Höpler, R. (no date) *Extending the IPG CarMaker by FMI Compliant Units*.

9 Appendices

9.1 Appendix A: Data Collection

9.1.1 Tires Spring Rates

It was necessary to find the spring rate of the tires used on the MCR. This value varies depending on which tire is being used and what pressure is inside the tire. For this reason, it was not possible to gain this data through any method other than first person testing.

Since the spring rate of the tire is dependent on the vertical load per unit vertical displacement of the tire (N/m), the hydraulic press was used. A jig was created, to allow a load to be placed at the centre of the wheel, as would be the case in the vehicle. Furthermore, a set of scales was positioned under the wheel to measure the vertical contact patch load. The jig was designed to prevent the tyre from rolling away under the force of the hydraulic press. Furthermore, DTI gauges were positioned on the rig and on the scale's surface to measure the deflection of the tire, as well as ensuring that the tire remained upright compared to the press.



Figure 48: Tyre Stiffness Rig

Starting at the static height of the tire, determined by applying the static load, the change in load was measured at 5mm increments of compression and 5mm of deflection. Each measurement was taken 3 times and the average load used moving forward. Once the loads had been converted from kg to N, force deflection curves could be produced.

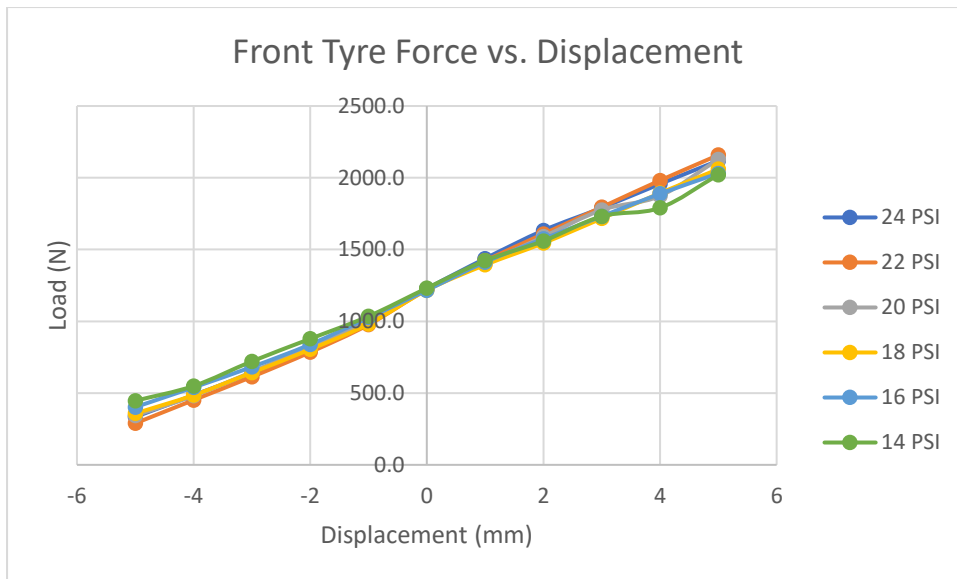


Figure 49: Front Tyre - Force Deflection Curve

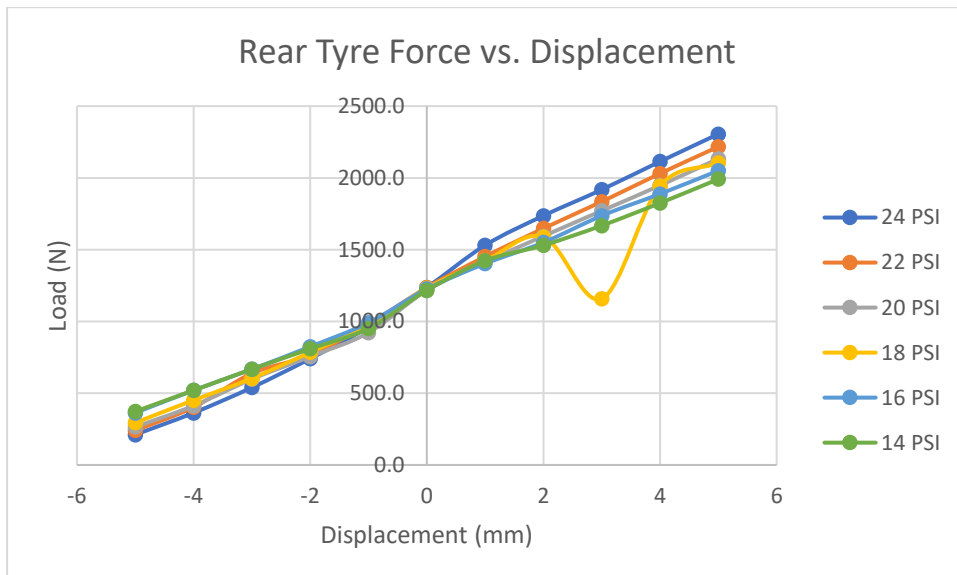


Figure 50: Rear Tyre - Force Deflection Curve

Table 5: Tyre Spring Rates

Tyre Pressure (PSI)	24	22	20	18	16	14
Front Spring Rate (N/mm)	178.542	186.8805	178.542	170.2035	162.846	157.4505
Rear Spring Rate (N/mm)	209.4435	197.6715	186.8805	180.9945	168.732	161.865

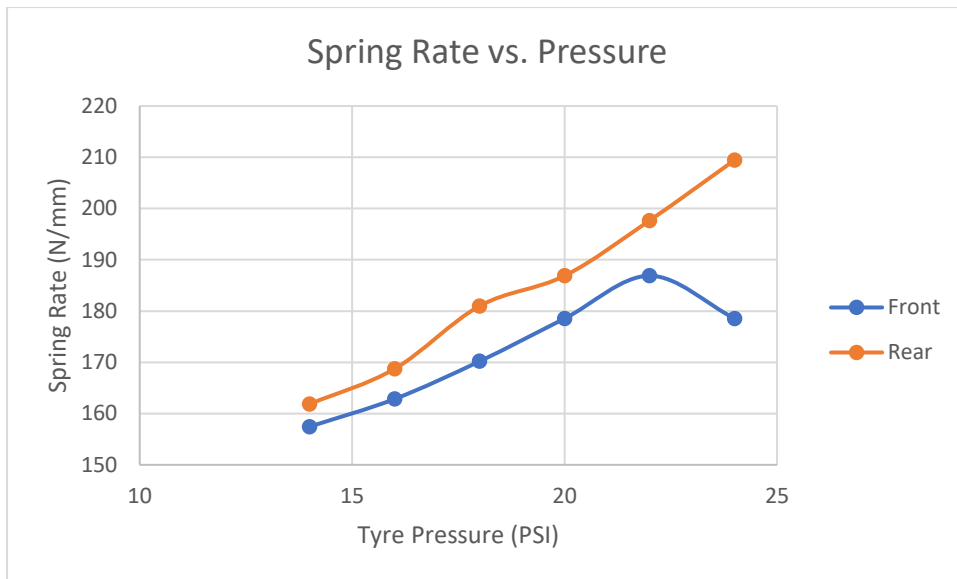


Figure 51: Spring rate at given tyre pressure

This rig had a few limitations, including the accuracy and precision of results and the data it was capable of measuring. The scales used, connected wirelessly to a laptop interface used to display the load. A significant time delay, as well as a tendency for values to fluctuate randomly, sometimes by more than 1 kg, and a rated accuracy of ± 0.01 kg made measuring loads at the correct displacements difficult. Furthermore, the rig only measured spring rate, and was not capable of recording the damping rate of the tire. Although the simulations used allowed for this information to be input, due to it not being available, tire damping was assumed to be 0 Ns/m. Finally, the rig was only capable of measuring loading for a vertical tire. The spring rate of a tire when camber is applied, as is the case in the MCR, could be significantly different.

9.1.2 Damper Rates



Figure 52: Damper Dynamometer

The damper dyno made it possible to determine the damping rates of the dampers currently being used on the MCR. Over the lifetime of a damper, wear and tear causes the damping values to change. As a result, any values for damping rates or damping coefficients given by the manufacturer cannot be trusted to be correct. The damper dyno allowed the testing of each damper at each click for bump and rebound.

The viscosity of the oil within a damper, is the primary factor in determining the damping characteristics. Since viscosity is dependent upon the temperature of the fluid, it was deemed necessary to determine the operating temperature of the dampers over the course of a race weekend.

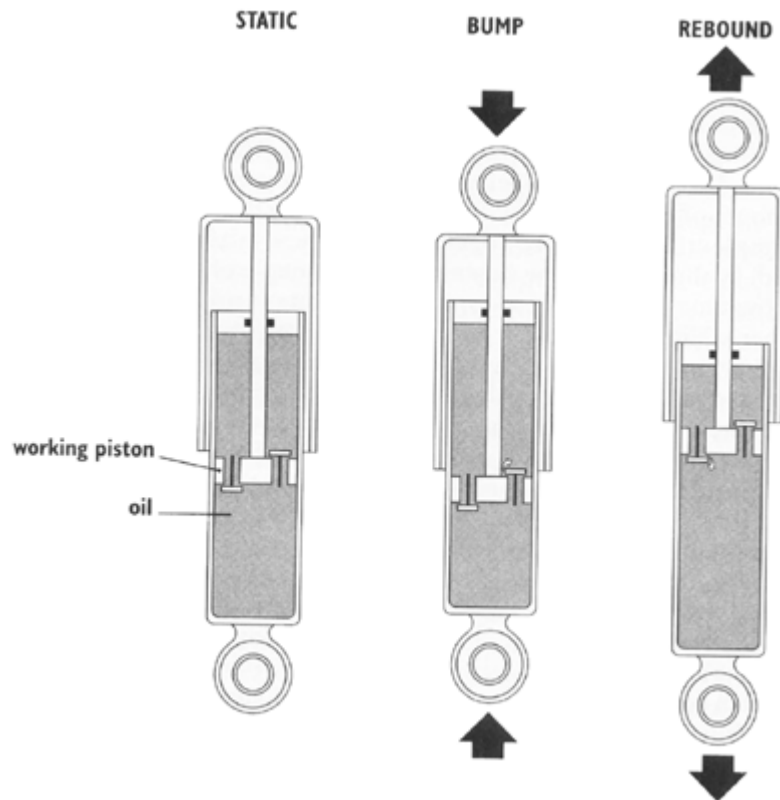


Figure 53: Damper cross section diagram (Calver, 2001)

Once this had been done, first the damp rate of each damper was tested at each setting at room temperature. In order to do this, the damper dyno records the force being applied on the damper across a sweep of damper velocities for bump and rebound. As the figure below shows, this results in a graph with four linear regions of varying gradients. Positive velocities indicate bump, whilst negative velocities indicate rebound. Bump and rebound are further split up into high, and low speed regions. The gradient in each region is the coefficient.

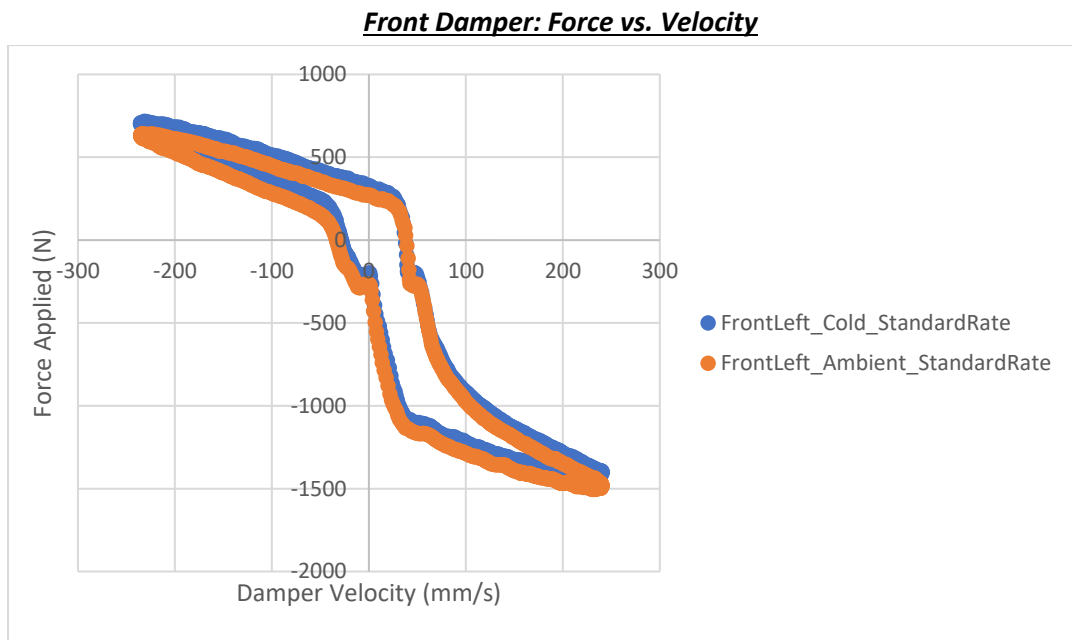


Figure 54: Front Damper Force vs. Velocity Graph

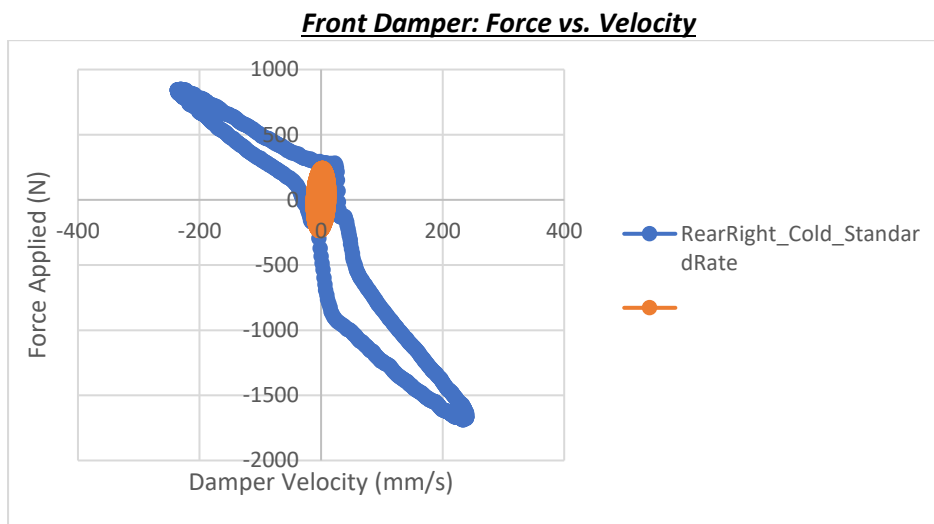


Figure 55: Rear Damper Force vs. Velocity Graph

Next, three sample damp rates at maximum, middle and minimum rebound, were measured at operating temperature and the results compared to the corresponding values at ambient temperature. As a result of this test, it was determined that the effect temperature has on the dampers was negligible compared to other uncertainties in the investigation. The front damper showed an 8% discrepancy whilst the rear showed a 29% discrepancy. As a result, in order to save time, the rates that had already been collected at ambient temperature were used.

Table 6: Ambient Temperature Damper Settings

Clicks	Front Damper Coefficient (N.s/m)				Rear Damper Coefficient (N.s/m)			
	Bump		Rebound		Bump		Rebound	
	High Speed	Low Speed	High Speed	Low Speed	High Speed	Low Speed	High Speed	Low Speed
1	2366.12	13134.33	-2315.00	-2319.40	2622.95	9656.72	-3590.82	-1402.99
2	2421.86	13252.24	-2420.30	-2635.61	2797.81	9914.18	-3628.32	-1698.00
3	2477.60	13370.15	-2525.60	-2951.82	2972.68	10171.64	-3665.82	-1973.52
4	2533.33	13488.06	-2630.90	-3268.03	3147.54	10429.10	-3703.32	-2249.04
5	2589.07	13605.97	-2736.20	-3584.24	3322.40	10686.57	-3740.82	-2524.57
6	2644.81	13723.88	-2841.50	-3900.45	3497.27	10944.03	-3778.32	-2800.09
7	2700.55	13841.79	-2946.80	-4216.66	3672.13	11201.49	-3815.82	-3075.61
8	2756.28	13959.70	-3052.10	-4532.87	3846.99	11458.96	-3853.32	-3351.13
9	2812.02	14077.61	-3157.40	-4849.07	4021.86	11716.42	-3890.82	-3626.66
10	2867.76	14195.52	-3262.70	-5165.28	4196.72	11973.88	-3928.32	-3902.18
11	2923.50	14313.43	-3368.00	-5481.49	4371.58	12231.34	-3965.82	-4177.70
12	2979.23	14431.34	-3473.30	-5797.70	4546.45	12488.81	-4003.32	-4453.22
13	3034.97	14549.25	-3578.60	-6113.91	4721.31	12746.27	-4040.82	-4728.75
14	3090.71	14667.16	-3683.90	-6430.12	4896.17	13003.73	-4078.32	-5004.27
15	3146.45	14785.07	-3789.20	-6746.33	5071.04	13261.19	-4115.82	-5279.79
16	3202.19	14902.99	-3894.50	-7062.54	5245.90	13518.66	-4153.32	-5555.31
17	3257.92	15020.90	-3999.80	-7378.75	5420.77	13776.12	-4190.82	-5830.84
18	3313.66	15138.81	-4105.10	-7694.96	5595.63	14033.58	-4228.32	-6106.36
19	3369.40	15256.72	-4210.40	-8011.16	5770.49	14291.04	-4265.82	-6381.88
20	3425.14	15374.63	-4315.70	-8327.37	5945.36	14548.51	-4303.32	-6657.40
21			-4421.00	-8643.58			-4340.82	-6932.93
22			-4526.30	-8959.79			-4378.32	-7208.45
23			-4631.60	-9276.00			-4415.82	-7483.97
24			-4736.90	-9592.21			-4453.32	-7759.49
25			-4842.20	-9908.42			-4490.82	-8035.01
26			-4947.50	-10224.63			-4528.32	-8310.54
27			-5052.80	-10540.84			-4565.82	-8586.06
28			-5158.10	-10857.04			-4603.32	-8861.58
29			-5263.40	-11173.25			-4640.82	-9137.10
30			-5368.70	-11489.46			-4678.32	-9412.63

9.1.3 Centre of Gravity

Determining the location of the centre of gravity of the overall vehicle was another piece of information that required physical testing to determine. Although SOLIDWORKS offers to calculate the centre of gravity, the results it gains are only as precise as the user's input. In order to achieve this, the user has to assign a mass or density to each component in an assembly and this information is used to find the centre of gravity. However, if certain components are only approximated, or not modelled at all, these can cause errors in the result. An example in this investigation is the entire powertrain of the MCR which was modelled as a single block. SOLIDWORKS assumed that the density throughout that block was homogeneous which looking inside the engine for example, reveals is not the case. In order to avoid these sources of potential error, tests were carried out on the real MCR.

Determining the 2D centre of mass was a simple case of placing all four wheels of the MCR on a set of scales. Comparing the percentage difference, between mass front to rear and left to right, shows the location of the centre of gravity as a percentage of the track and wheelbase.

Table 7: 2D mass distribution

Left (%)	Front (%)	Total Mass (kg)
53.125	25.987	304

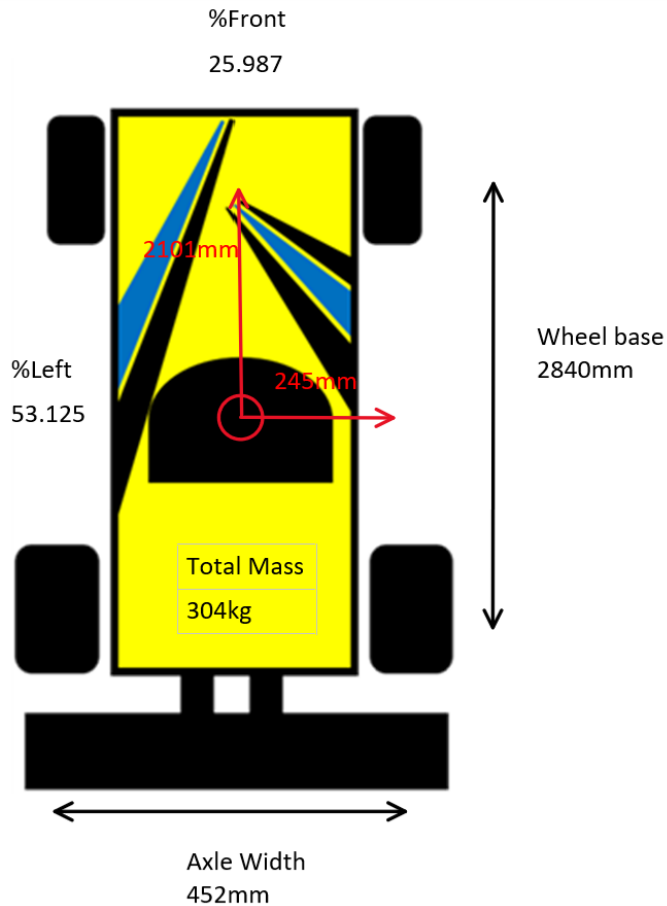


Figure 56: 2D CoG Calculation

With the 2D CoG known, it is now necessary to calculate the height of the centre of gravity above the ground. Using Equation 18 makes it possible to determine this height above the contact point to the scales. To get an accurate result, the calculation was completed at varying angles (θ) and the average of the resulting values taken.

$$CoG\ Height = \frac{Wheel\ Base \times Front\ Mass\ Change}{Total\ Mass \times \tan \theta}$$

Equation 18: Centre of Gravity Height

In order to determine the change in front mass at varying angles (θ) the rear of the vehicle was lifted using a gantry, as seen in the figure below. The wheelbase was taken to be the distance between the contact point to the scales, and the pivot point on the gantry. 0° was taken at the point where the front and rear of the chassis were at the same height above the level ground.



Figure 57: CoG Height test set-up

Table 8: CoG Height Results

Mass LF (kg)	Mass RF (kg)	Ang Lat (deg)	Ang Long (deg)	Chassis Height (cm)
88	0	0.48	0.9	0.057
86.5	0	0.6	3.62	0.184
88	0	0.82	5.83	0.295
86.5	0	1.08	8.3	0.416
83	1.5	1.1	10.84	0.540
80	2	1.06	13.62	0.672
77	2.5	1	16.12	0.791
82.5	2.5	0.93	18.66	0.907
80.5	3	1.38	21.3	1.032
78.5	4	1.42	24.15	1.158
82	3.5	1.43	26.74	1.272
86.5	3	1.38	33.52	1.566
85.5	4.5	1.34	20.66	1.003

9.2 Appendix B: Cos Bump Testing

In order to define the shape of the 1-Cos(x) shaped bump, the following code was used. For consistency, the resultant shape was used in each of the simulations, as well as in real world testing.

```
%% Given Data

h = 40; %bump amplitude (mm)
l = 2*(pi*40); %bump length ca. 2*125.66(mm)
velocity = 4; % vehicle speed m/s
% equation/shape: y=1-cos(x)*20
% MSC/Adams uses 1 frame/s
DistStep = (251.3267/200)/1000; %m

%% creating horizontal displacement column

%Presuming 200 steps at 1 step per second

counter = 1;
while counter<201
    tableXY(counter,1)=counter*((2*pi)/200); %x value
    table(counter,1)=31.8309*(tableXY(counter,1)*(1/200));

    tableXY(counter,2)=1-cos(tableXY(counter,1));
    table(counter,2)=tableXY(counter,2)*20;

    table(counter,3:21)=0.0;

    counter=counter+1;
end

% plot(table(:,1),table(:,2))
% ydisp = timeseries(table(:,2));

ydisp = zeros(600,2);

counter = 1;
while counter<=600
    counter2 = counter - 200;
    ydisp(counter,1) = velocity * (DistStep * counter);
```

```

if counter <=200
    ydisp(counter,2) = 0;
elseif counter >400
    ydisp(counter,2) = 0;
else
    ydisp(counter,2)= table(counter2,2);
end
counter = counter + 1;
end

```

```

clear counter
clear counter2
clear DistStep

```

```

% save('ydisp','-v7.3','ydisp')
% save('table','-v7.3','table')

```

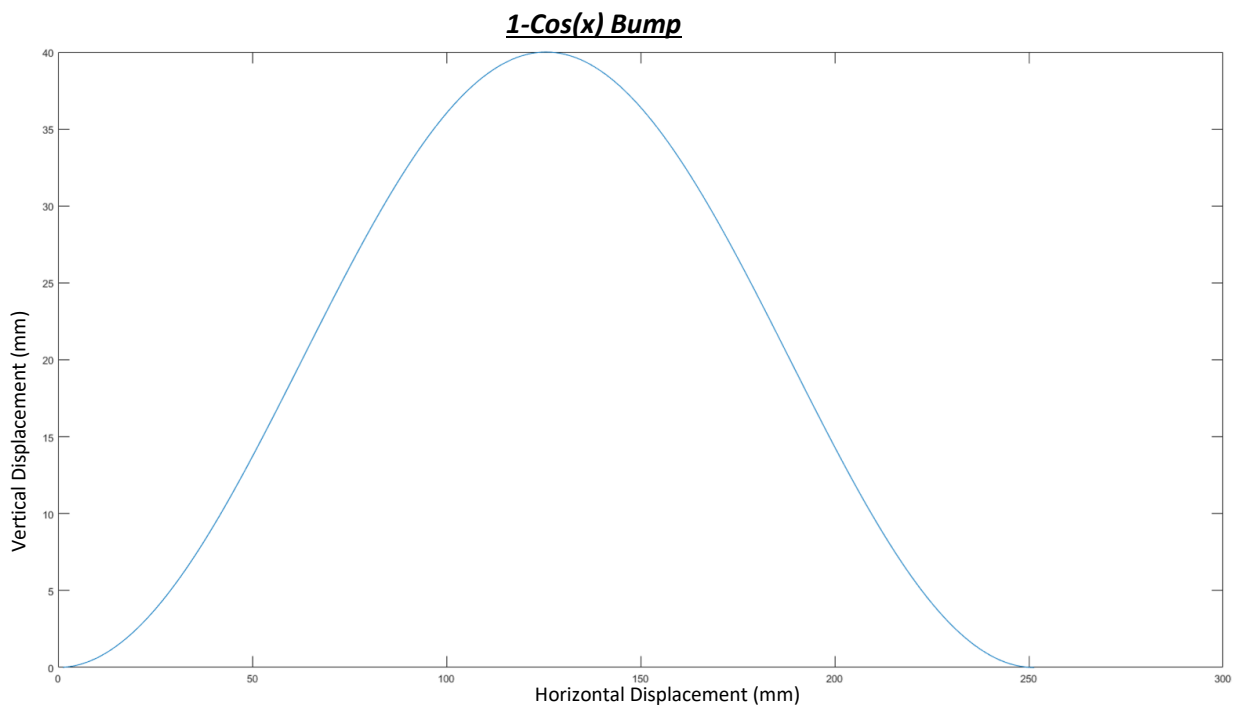


Figure 58: 1-Cos(x) Bump dimensions

Cos Bump Suspension Test Plan

Cornelius May



Figure 59: Cos Bump Test setup

Abstract

In order to correlate data received from a simulation, it is essential to have a test to which to compare your data. This test needs to be simple to run, both in real life and in simulation, easily repeatable, and results must be independent from human error.

Although other tests, such as those described in a range of ISO standards, these tend to rely, at least in part, on a consistent driver. As an alternative to these, the vehicle could be run over a standardised, calibrated bump. Due to the way in which the bump can be defined, it can be created for real life testing, and defined for an in-simulation track. This would serve as a simple and repeatable method to calibrate a suspension simulation model.

Equipment

- Vehicle
- Data Logger
- Calibrated Bump
- Steering Lock
- Flat Surface



Figure 60: 1-Cos(x) Bump

The equipment required for this testing is limited. The two most important things to have are your vehicle and the calibrated bump. It is important that as much as possible is known about these so that they can be accurately modelled in your simulation. The author recommends using a bump defined by a $1-\cos(\theta)$ curve with dimensions defined by the amplitude in a way that ensures sufficient clearance between the top of the bump and the chassis of the vehicle, so that the bump does not touch the vehicle throughout the test procedure.

The data logger needs to be capable of measuring a few things. As a minimum, damper stroke and vehicle speed need to be logged. Any other data being logged can be used as another reference point, for comparison between real world and simulation.

The steering lock, although not essential, ensures that it is not up to the driver to keep the steering wheel straight. Keeping the wheels straight, as well as having a flat surface on which to conduct your test, is essential for having consistent and repeatable results. Most types of simulation will produce a

surface and driving behaviour that are “ideal”, so imitating these as closely as possible in real life, will ensure only the calibrated bump, causes input into the suspension system.

Method

Preparation:

Before starting the test, both the test area and the vehicle need to be prepared in order to ensure a repeatable environment is maintained.

In order to prepare the vehicle for this test, a range of things need to be checked. Fuel levels, tyre pressures, battery SoC (state of charge) and wheel torque needs to be checked. Next, it is important to check the suspension set-up as this is the primary part of the vehicle under investigation. As well as these needing to be checked, for the general health of the vehicle, areas such as the fuel level or suspension setup can affect the vehicles similarity to the simulation. Once the vehicle is ready for the test, the steering needs to be locked straight ahead and the data logger started.

Preparing the test area is also important. The area should ideally be tarmac, swept as clean as possible and dry. Furthermore, since the steering is going to be locked, it is important to ensure that any potential obstacles are removed. Warning signs might be required to ensure pedestrians do not enter the test area. Next the calibrated bump needs to be positioned. A suitable run up for the vehicle to get up to speed is required, as well as a runoff braking area. These areas should also be long enough for the vehicle suspension to settle once it has reached the test speed and again after the bump. Since the steering will be locked, the run up and run off need to be straight. The bump needs to be weighed down to ensure it will not move whilst the vehicle runs over it or in between runs.

Finally, the vehicle needs to be positioned in front of the bump ready for the test to begin.

Test Procedure:

1. Accelerate the vehicle to test speed.
2. Maintain test speed as vehicle settles.
3. Run over bump.
4. Maintain test speed as vehicle settles.
5. Come to a stop.
6. Collect data from data logger.
7. Reset test.
8. Repeat steps 1-7 a total of 3 times.
9. Make required changes to test procedure.

10. Repeat.

After test procedure above can be conducted both in simulation and in “real life”. The data collected from the data logger can be compared to the data collected from the simulation. This can be used to identify areas of inaccuracy in the simulation model. In some situations, if there are multiple inaccuracies in the simulation, it is still possible to gain results that indicated an accurate simulation model. Making changes to the test procedure, such as changing the suspension setup or test speed, ensures that a false positive has not been achieved.

9.3 Appendix C: Weight Transfer

$$WeightTransfer_{Elastic} = \frac{K_w T \theta}{2}$$

Equation 19: Elastic Weight Transfer

$$WeightTransfer_{UnSprungGeometric} = \frac{m_{us} \times a_y \times r}{T}$$

Equation 20: Un-Sprung Geometric Weight Transfer

$$WeightTransfer_{SprungGeometric} = \frac{m_s \times a_y \times h}{T}$$

Equation 21: Sprung Geometric Weight Transfer

Each of the equations above is used to calculate a single component of lateral weight transfer and relates to the weight transfer across a single axle of a vehicle. The equation below is used to calculate the total weight transfer for a single axle. In order to calculate total weight transfer across both axles, the same equations need to be used for each axle and the average (mean) value calculated.

$$\begin{aligned} TotalWeightTransfer \\ = & WeightTransfer_{Elastic} + WeightTransfer_{UnSprungGeometric} \\ & + WeightTransfer_{SprungGeometric} \end{aligned}$$

Equation 22: Total Lateral Weight Transfer – Cumulative

In Simulink, the angle of the road wheels is not calculated, and as a result the weight transfer caused due to the turning of road wheels has not been discussed throughout this paper. As the road wheels turn, the caster will cause the inside front wheels load to increase, causing weight transfer to the outside rear corner (Tudor and Tudor, 2016). Whilst Adams takes this into consideration due to the entire geometry being modelled, Simulink cannot. This is another source of error in the investigation and a potential area for future improvement of the Simulink model.

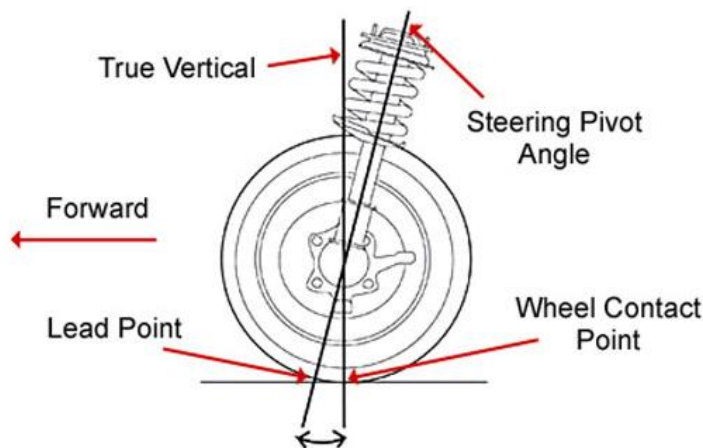


Figure 61: Caster induced Weight Transfer (Caster – Geometry Explained – Suspension Secrets, no date)

9.4 Appendix D: Adams Data Input

9.4.1 Springs

Below is the definition file for the front springs used throughout this investigation. As well as showing units and file constants, a table to define the spring force at each displacement is available. The user can define the level of precision used themselves, simply by inputting more points. Adams will use linear interpolation for values that fall in-between the defined range. Although four individual files can be used to define each spring individually, since on the MCR the springs are symmetrical left to right, only two definition files were necessary.

```
$-----MDI_HEADER
```

```
[MDI_HEADER]
```

```
FILE_TYPE   = 'spr'
```

```
FILE_VERSION = 4.0
```

```
FILE_FORMAT  = 'ASCII'
```

```
$-----UNITS
```

```
[UNITS]
```

```
LENGTH = 'meter'
```

```
ANGLE  = 'degrees'
```

```
FORCE  = 'newton'
```

```
MASS   = 'kg'
```

```
TIME   = 'second'
```

```
$-----SPRING_DATA
```

```
[SPRING_DATA]
```

```
FREE_LENGTH = 0.2057
```

```
$-----CURVE
```

```
[CURVE]
```

```
{ disp    force}
```

```
-1.0    -26000
```

```
-0.5    -13000
```

```
0.0      0.0
```

```
0.5     13000
```

1.0 26000

9.4.2 Dampers

Damper definition files work in a similar way to the files used to define the springs. The key difference to notice in this file is that forces are defined proportionally velocity, not displacement.

\$-----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}

-200 -980

-70 -280

0 0

70 280

200 440

9.5 Appendix E: Chassis Stiffness

Throughout the investigations carried out, the stiffness of the chassis was not taken into consideration. As well as other components being considered as rigid, this would have been a source of uncertainty. Whilst Adams itself can calculate chassis stiffness given the material properties, within Simulink, this would be a basic reworking of the program with lateral, longitudinal and diagonal interactions having to be considered as spring dampers, as opposed to rigid connections. Both rotational as well as translational stiffness would have to be considered.

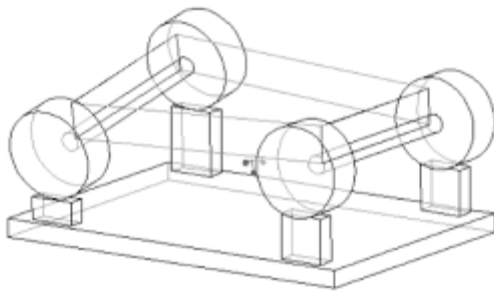


Figure 62: Chassis stiffness (Herrmann, 2016)

9.6 Appendix F: SimScape Multibody Simulation

The SimScape multibody library is designed to resolve some of the issues outlined above (*Simscape Multibody - MATLAB & Simulink*, no date) concerning rotational inertia of components. The blocks within this library are capable of modelling components in a 3D environment. A template provided by The Racing Lounge, allows users to alter the model to suit their own vehicle and use pre-defined maneuvers to investigate the assembly (Hahn and Marquez-Brunal, 2020).

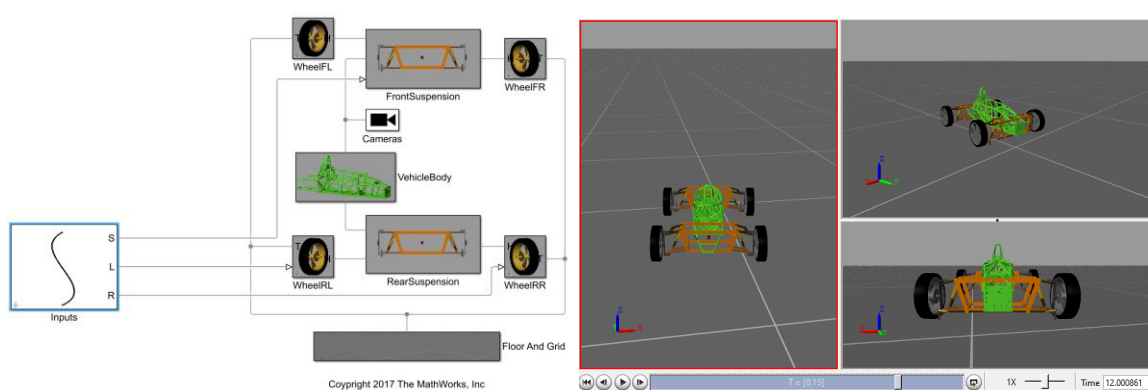


Figure 63: SimScape Multibody Vehicle Simulation

The primary issue with this simulation is that it is not designed to allow users to define their own test procedures. Furthermore, there is no driver model capable of simulating a full track simulation, which is what is the final goal of this investigation. As a result, this simulation was not used.

9.7 Appendix G: IPG CarMaker

An alternative to Adams, commonly used in the automotive industry, IPG has a number of strengths, making it a better candidate for this investigation. Its ability to link directly into Simulink would have made parallel simulations possible. This would have allowed identical inputs between the two simulations, resulting in two sets of results. Furthermore, the input of data is closer related to the interfaces used by BPS, potentially making model creating a quicker process. Ultimately, this software was not used, due to its unavailability to the university during the course of this investigation.

9.8 Appendix H: Miscellaneous

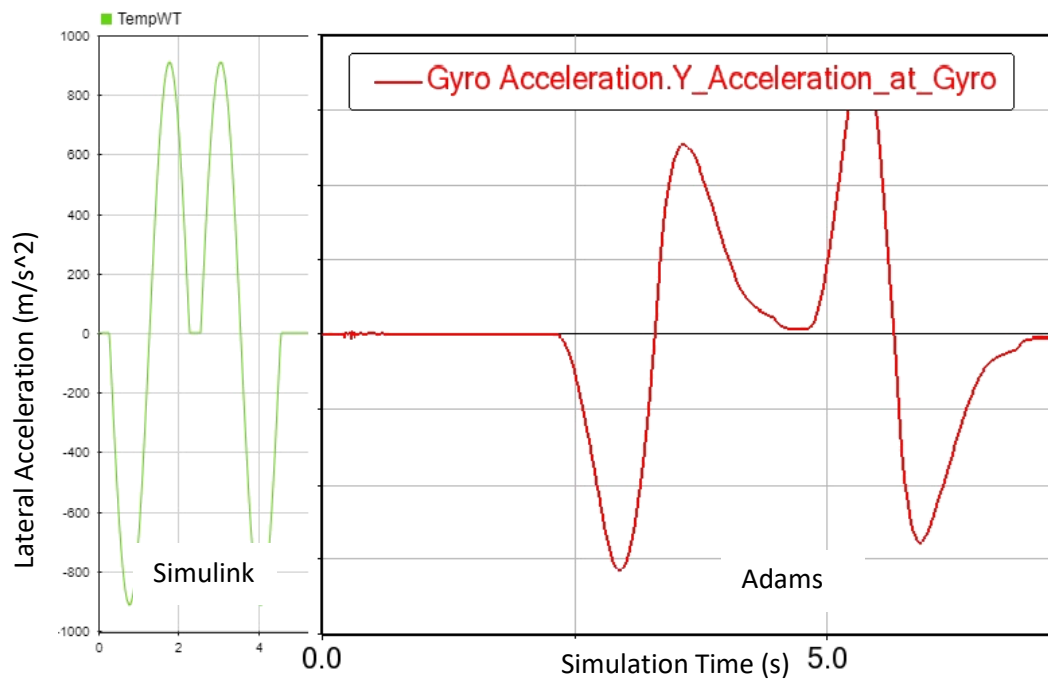


Figure 64: Double Lane Change - Lateral Acceleration

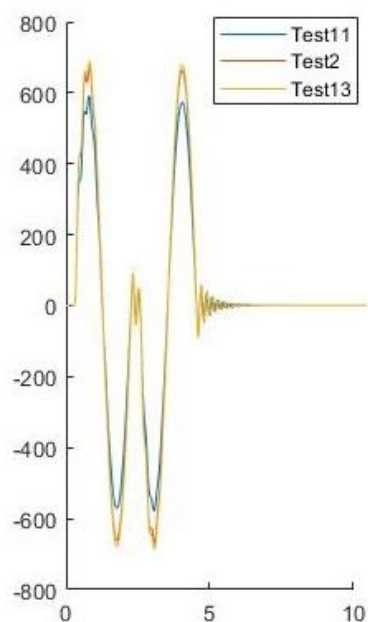


Figure 65: Simulink Double Lane Change - Roll Stiffness distribution