



**MONASH MOTORSPORT**  
**FINAL YEAR THESIS COLLECTION**

**Vehicle Simulation for Concept Level Design  
and Specification of a Formula SAE Racecar**

*Cameron Warne - 2016*

The Final Year Thesis is a technical engineering assignment undertaken by students of Monash University. Monash Motorsport team members often choose to conduct this assignment in conjunction with the team.

The theses shared in the Monash Motorsport Final Year Thesis Collection are just some examples of those completed.

These theses have been the cornerstone for much of the team's success. We would like to thank those students that were not only part of the team while at university but also contributed to the team through their Final Year Thesis.

The purpose of the team releasing the Monash Motorsport Final Year Thesis Collection is to share knowledge and foster progress in the Formula Student and Formula-SAE community.

**We ask that you please do not contact the authors or supervisors directly, instead for any related questions please email [info@monashmotorsport.com](mailto:info@monashmotorsport.com)**

# VEHICLE SIMULATION FOR CONCEPT LEVEL DESIGN AND SPECIFICATION OF A FORMULA SAE RACECAR

CAMERON STUART WARNE 23408405

SUPERVISED BY: DR SCOTT WORDLEY



Figure 1 M15 Competing at the Australasian competition.

## **SUMMARY**

Previous work done by Russouw, Bett, and Geist has produced a tyre model, competition simulator and a vehicle model. These three projects have previously been disjointed and not significantly influenced the concept level design of Monash Motorsport Formula SAE racecars. This project developed and extended these three tools into a completion simulator is built on a four wheeled, 3 degree of freedom vehicle model using a tyre model that has been fitted to available tyre testing data. This simulator was developed to include a graphical user interface, so that unfamiliar users will find it intuitive, easy and quick to use. The final part of this project is to apply the developed program and start using it to drive concept level design and specification of the 2017 Monash cars. One of the investigations was energy management and aerodynamics of the 2017 electric car. The simulator was able to demonstrate that effective techniques could be used to reduce the battery consumption by up to 40% and that even with an aerodynamics package, reasonable energy usage levels can be achieved.

## TABLE OF CONTENTS

<b>1.0</b>	<b><i>NOMENCLATURE</i></b>	<b>5</b>
<b>2.0</b>	<b><i>INTRODUCTION</i></b>	<b>5</b>
	<b>2.1</b> <b><i>AIMS</i></b>	<b>6</b>
<b>3.0</b>	<b><i>TYRE MODEL</i></b>	<b>6</b>
<b>4.0</b>	<b><i>VEHICLE MODEL</i></b>	<b>6</b>
	<b>4.1</b> <b><i>CONSTANT RADIUS SIMULATION</i></b>	<b>9</b>
<b>5.0</b>	<b><i>DEVELOPMENT OF COMPETITION SIMULATOR</i></b>	<b>9</b>
	<b>5.1</b> <b><i>CREATING LIMIT SURFACE</i></b>	<b>10</b>
	<b>5.2</b> <b><i>ACCELERATION SIMULATION</i></b>	<b>10</b>
	<b>5.3</b> <b><i>SKIDPAD SIMULATION</i></b>	<b>11</b>
	<b>5.4</b> <b><i>AUTOCROSS AND ENDURANCE SIMULATION ``</i></b>	<b>12</b>
	<b>5.5</b> <b><i>EFFICIENCY SIMULATION</i></b>	<b>13</b>
<b>6.0</b>	<b><i>DEVELOPMENT OF GRAPHICAL USER INTERFACE (GUI)</i></b>	<b>14</b>
	<b>6.1</b> <b><i>MAIN MENU</i></b>	<b>15</b>
	<b>6.2</b> <b><i>TYRE MODEL GENERATOR</i></b>	<b>16</b>
	<b>6.3</b> <b><i>VEHICLE MODEL GENERATOR</i></b>	<b>16</b>
	<b>6.4</b> <b><i>POWERTRAIN MODEL GENERATOR</i></b>	<b>18</b>
	<b>6.5</b> <b><i>TRACK GENERATOR</i></b>	<b>18</b>
	<b>6.6</b> <b><i>ADVANCED SIMULATOR</i></b>	<b>19</b>
	<b>6.6.1</b> <b><i>SENSITIVITY ANALYSIS</i></b>	<b>21</b>
<b>7.0</b>	<b><i>VALIDATION</i></b>	<b>22</b>
<b>8.0</b>	<b><i>USABILITY</i></b>	<b>24</b>
<b>9.0</b>	<b><i>APPLICATION</i></b>	<b>25</b>
<b>10.0</b>	<b><i>FUTURE WORK</i></b>	<b>27</b>
<b>11.0</b>	<b><i>CONCLUSION</i></b>	<b>28</b>
<b>12.0</b>	<b><i>REFERENCES</i></b>	<b>29</b>

## **1.0 NOMENCLATURE**

$A_x$  = Longitudinal acceleration ( $m/s^2$ )

$C_n$  = Yaw acceleration ( $rad/s^2$ )

$F_y$  = Longitudinal Force (N)

$M_x$  = Overturn Moment (Nm)

$A_y$  = Lateral acceleration ( $m/s^2$ )

$F_x$  = Lateral Force (N)

$F_z$  = Vertical Force (N)

$M_z$  = Self Aligning Moment (Nm)

## **2.0 INTRODUCTION**

Formula SAE is the largest student engineering competition in the world with over 500 universities participating. The competition is assessed on a point's basis with 375 out of 1000 points allocated to static events where engineering design, cost effectiveness and a business presentation are assessed. The remaining 625 points are allocated to dynamic events which are composed of a straight line acceleration test, a cornering test, a one lap time trial and a 22km endurance event, where both speed and efficiency are assessed. The competition is also split into two categories, combustion cars which use four stroke internal combustion engines with up to 610cc displacement, and Electric cars with an 80 kW power limit. Previously the Monash Motorsport team has competed in just the combustion category but are currently developing an electric car to compete in 2017. This project is focused on developing tools to analyse the performance of the car in the dynamic events at a concept level for both the combustion and electric cars.

Previous team members have developed vehicle dynamic models and lap simulators to investigate the performance effects of different vehicle concepts. Steven Webb (2012), developed a point mass competition simulator that, given basic vehicle parameters such as mass, power, grip and aerodynamic coefficients, would simulate a car performing all five dynamic events. This was a basic but very useful tool for analysing broad vehicle concepts. Further final year projects expanded on this to develop more complex tools that could be used to better investigate vehicle parameters. Martin Bett (2015) developed a more complex competition simulator that incorporated additional vehicle parameters such as yaw acceleration, and vehicle acceleration limit surfaces. Marc Russouw (2014) developed a non-dimensional tyre model for the Goodyear D2704s, which was used to better understand the tyre and its effect on car performance. Finally Michael Geist (2015) developed a four wheel, three degree of freedom vehicle model that simulates transient and steady state vehicle behavior, using Russouw's tyre model. Even though the knowledge of vehicle dynamics within the Monash FSAE team, has greatly improved due to the development of these tools they have not had a substantial effect on Monash Motorsport's concept level design and vehicle specification.

The development and implementation of a combined simulator that uses all three tools would allow current and future members, to evaluate concept level design decision with great depth and accuracy. This program consists of a new, more versatile version of Russouw's tyre model that can be quickly updated with new tyres, this feeds into Geist's Vehicle model which will be used to create a performance envelope of the car. The performance envelope is then used by Bett's competition simulator to determine the points effect of concept level design decisions.

## **2.1 AIMS**

The aim for this project is to develop and implement a competition simulator that is easy to use for current and future Monash Motorsport team members for the purpose of making concept level design decision and specifications for the Formula SAE racecar. To achieve this, previous vehicle simulation tools will be used and expanded upon to meet the needs of current and future designers.

## **3.0 TYRE MODEL**

Based on Russouw's Tyre Model (2014), a new tyre model was successfully developed, that substantially improved upon the original code in both implementation and feature set. Since this project is designed to be a tool for future FSAE team members to use and adapt easily, understandability and a logical process with good comments were essential to allow future users to modify the code efficiently.

The Pacejka "Magic Formula" (Pacejka, 2006) tyre model was chosen because of its simplicity and computational efficiency. This is the same as what was used in Russouw's original tyre model. Russouw's tyre model produced a well-fitting model but had a ridged structure such that adaptation for different tyres and different sets of data is very time consuming and need a thorough knowledge of the code. The new tyre model is structured so that slip angle and slip ratio sweeps, camber settings, and pressure settings can be easily changed for different sets of tyre data, making it much more versatile and able to create a tyre model from any data.

The detail of the tyre model has also been improved upon compared to Russouw's tyre model. Russouw's model took inputs of normal load, camber, slip angle and slip ratio, and calculated the resultant  $F_x$  and  $F_y$ . The new tyre model now also takes into account tyre pressure, and rim width, and in addition to  $F_x$  and  $F_y$  outputs the Moments  $M_x$  and  $M_z$ , also known as the overturn moment and self-aligning moment. These moments can contribute up to 10% of the vehicle's weight transfer and yaw moment (Milliken & Milliken, 1995), so will be beneficial in improving the accuracy of the vehicle model.

TTC Data was used to calculate the lateral relaxation length which is defined as the distance the tyre travels to get to within 36% of the steady state  $F_y$  after a step input (Pacejka, 1995). The relaxation length is not used in steady state simulations but will be used in transient vehicle simulations.

## **4.0 VEHICLE MODEL**

Once the tyre model was completed it was able to be implemented in to Geist's steady state, three degree of freedom vehicle model. Since the tyre model function was in the a similar format as the previous one, the new tyre model was able to work with Geist's original Vehicle model with little modification to the code. The biggest improvements that were able to be made with the model was by improving the efficiency of the code. The original code took 48 minutes to create one  $C_n$  vs  $A_y$  plot, this created a number of issues with the practicality of the code. In order to create a limit surface that would be needed for the lap simulator multiple  $C_n$  vs  $A_y$  plots are needed to be generated at different  $A_x$  values. This would take up to 8 hours to complete, further if a sensitivity study for mass was wanted to be undertaken multiple days of computation time would be required compared to 15 minutes with Martin Bett's competition simulator (Bett, 2015). Improvements were able to made through the use of MATLAB's profiler, the main contribution to computing time was the number of calls made for the tyre model. Through use of vectorization and inbuilt optimization

functions in MATLAB a solve time of 3.5 minutes was achieved. This would still result in a sensitivity study taking 6 hours. The resulting vehicle model solver was called “Super Speed”. Further improvements required parallelization and restructuring of the solver using the pseudo code shown below in figure 2

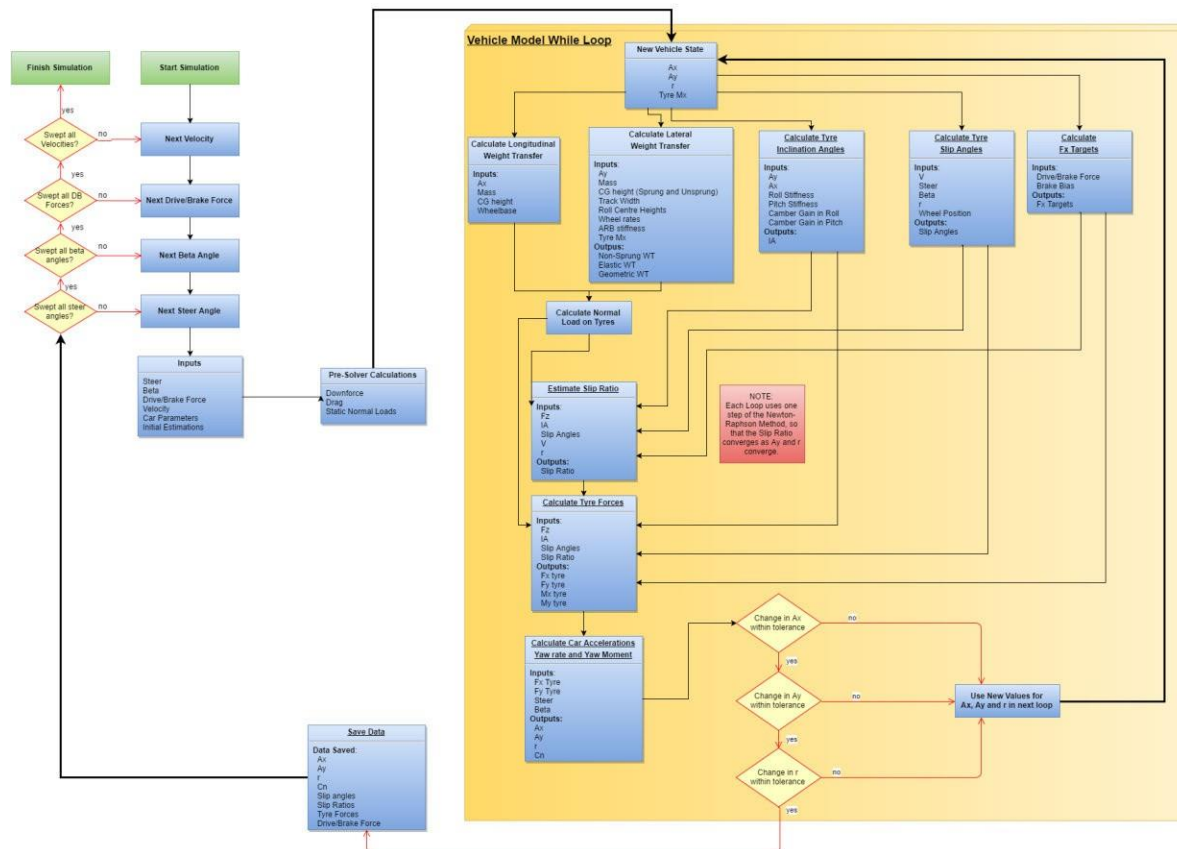


Figure 2: Vehicle model pseudo code

Modern microprocessors nowadays commonly have two or four cores and are able to handle up to 8 threads, hence parallelization of the code can yield substantial benefits. This will be particularly beneficial when large sensitivity studies are done, there is a computer available to undergraduates that has an Intel Xeon with 12 cores and 24 threads, the full power of which can only be leveraged through parallelization.

This restructuring of the code lead to further, substantial performance increases, decreasing the solve time for a single Cn-Ay plot to 10 seconds, a similar sensitivity study could now be done in 30 minutes, 15 minutes for computing the limit surfaces and 15 minutes for the competition simulator to solve. The restructured and parallelized code was called “Hyper Speed”, a comparison of the performance of these three codes are shown in figure 3. These improvements have made it possible to use the vehicle model in conjunction with the competition simulation, when making concept level design decisions. It also opened up the ability to add a velocity dimension to the limit surface to better capture the effects of speed on the performance of the car.

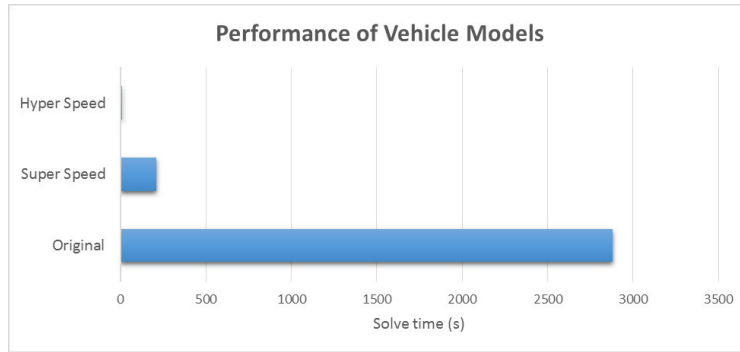


Figure 3: Performance of Vehicle Models

Another significant modification to the vehicle model is a spool drivetrain in addition to an open differential. This has provided some insight into the effect of driveline setup. An open differential allows the two driven wheels to spin at different speeds since the inner wheel tracks a shorter path than the outer wheel around a corner. The spool fixes the rotation of one wheel to the other, hence they have to spin at the same speed. This results in the open diff having to apply the same torque but different speeds to each of the driven wheels whereas the spool forces the wheels to have the same speed but can have different torque. As shown in the Cn-Ay plot in figure 4, the spool is able to achieve greater lateral acceleration since it is not limited by having to apply equal torque to each of the driven wheels. The Cn-Ay plot also shows that the spool has higher level of stability but less controllability which are proportional to the slope of the constant steer and constant chassis slip lines respectively. Monash Motorsport's 2016 car, M16, currently employs a limited slip differential (LSD) which allows a speed differential but induces a torque bias. An LSD could be implemented in future work.

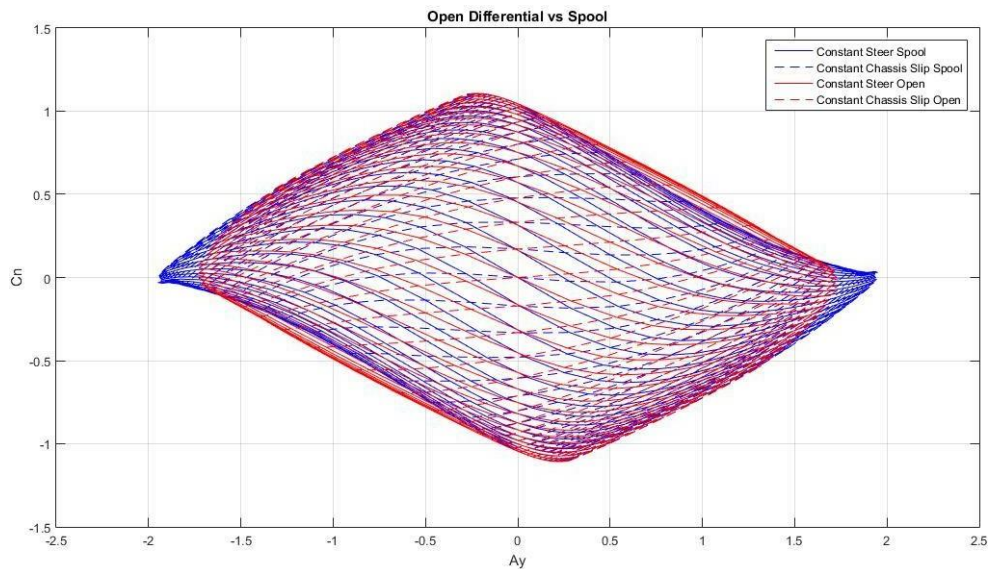


Figure 4: Cn-Ay diagram of spool and open diff



#### 4.1 CONSTANT RADIUS SIMULATION

A critical function that needed to be developed was to solve for the highest speed around a given radius. For this problem there are two independent variables ( $\delta, \beta$ ) and two dependent variables ( $C_n$ , internal radius) that are being solved for. This was achieved through use of a three-dimensional application of the Newton-Raphson method, demonstrated in figure 6. The fastest speed was then determined through a bisection method.

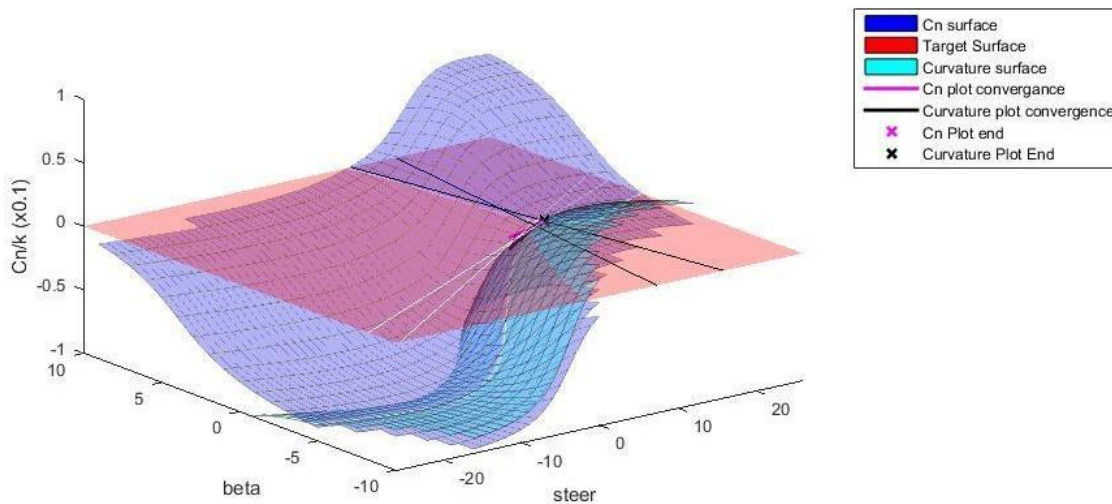


Figure 5: Solving for Constant radius

The ability to find the maximum steady state speed, around a given radius will be used for the skidpad event.

#### 5.0 DEVELOPMENT OF COMPETITION SIMULATOR

Once the vehicle model, had been developed it was necessary to use the model to simulate the car performing a number of manoeuvres, to complete the dynamic events at competition. There were two methods that were considered in order to achieve this: live solving or solving the vehicle model while simulating events or pre-solving the vehicle model to create a limit surface function which will then be used in simulation. Both techniques were used for different events.

Live solving was chosen for the skidpad event using the constant radius simulation described in 3.1 this allows for more control over the accuracy and is not subject to discretization issues when solving limit surfaces. This process takes around 3s per car, compared to 0.1s for a pre-solved method, the increase in solve time is acceptable and will have minimal effect on the usability of the program.

Acceleration event is a good candidate for a transient simulation since it is in a straight line and time dependent effects such as throttle response, shift times, damping and geometric weight transfer greatly affects the performance of the car. However this is outside the scope of this project so a simple pre-solved method was used until a transient method has been developed.

Using live solving for Autocross or Endurance events, where the car has to navigate a circuit, is much more complicated and computationally intensive. It would require the constant radius simulation to solve for each of the “critical points” on the racetrack where curvature is highest, then solve, at each distance step, the lowest  $A_x$  braking backwards from the critical points and the highest  $A_x$  accelerating forwards from the critical points. A typical Endurance/Autocross track is 1.0-1.5 km long, a distance step of 0.5m would result in 2000-3000 points to solve, and this would take 3-4 hours to solve per car. Taking most of a day to solve for one car is unreasonable and impractical for the number of simulations that are required; hence a pre-solved method is used.

### **5.1 CREATING LIMIT SURFACE**

A pre-solved method consists of a limit surface function which represents the performance envelope within which the car can operate. The surface consists of a roughly convex, continuous 4D surface which specifies the maximum of any combination of  $A_x, A_y, C_n$  and  $V$  that is achievable. This is constructed by solving the vehicle model through sweeps of steered angle, chassis slip angle, Drive/Brake force and Velocity. This creates a cloud of values of in the  $A_x, A_y, C_n, V$  space, the extreme values are used to create a linearly interpolated function, using the scatteredInterpolant Matlab function as shown in (figure 6). This function has the form  $(A_x) = \text{function}(A_y, C_n, V)$ . The drive and brake functions are separated, where  $DBforce \geq 0$  and  $DBforce < 0$  and have the form  $(A_x) = \text{function}(A_y, C_n, V)$ . A  $V_{limit}$  function ( $V = V_{limit}(\theta, k)$ ) is also generated so that speeds at critical points can be calculated. Additional functions of steered angle, chassis slip angle and  $DBforce$  are generated so that the vehicle model inputs can be reverse engineered once the simulation has solved.

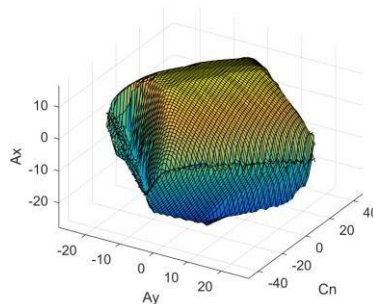


Figure 6: Cn-Ay-Ax Limit Surface

### **5.2 ACCELERATION SIMULATION**

Acceleration event is a simple test of the vehicle’s straight-line performance. This consists of a 75m straight where the time taken to complete the distance is used to calculate the scores. This event favours cars with good power to weight ratios, longitudinal grip and low drag. Since it is a straight line run it makes it one of the easier events to simulate.

As stated previously the acceleration simulation is a good candidate for a transient solver since it is simple to solve and may benefit from the addition of time dependent effects such as throttle response, shift times, damping and geometric weight. However, this is outside the scope of this project, so a simple pre-solved method was used until a transient method has been developed. The 75m

acceleration run is split up into 0.1m distance increments, where the acceleration of the car is determined and the speed for the next distance step is calculated assuming constant acceleration.

The calculation of vehicle acceleration is determined though comparing the grip available from the tyres and power available from the motor. The acceleration available from the grip of the tyres is given by the drive surface function. At each distance step the power available is determined by the powertrain model where power for a given speed is already calculated. The acceleration potential provided by the motor is determined by equation 1 below

$$a_x = \frac{p}{m \times v_x}$$

Equation 1

The velocity of the next distance step is calculated given a constant acceleration assumption, using the equation 2 below.

$$v_{x(i+1)} = \sqrt{v_{xi}^2 + 2 \left( a_x - \frac{1}{2} \times \rho \times CdA \times \frac{v_{xi}^2}{m} \right) * d_{step}}$$

Equation 2

Scoring of the acceleration event is given by the official FSAE rules in equation 3, where Tmin is the time of the fastest car, Tmax is 1.5xTmin, and Tyour is the time of the car in question:

$$Acceleration\ Score = 95.5 \times \frac{\frac{T_{max}}{T_{your}} - 1}{\frac{T_{max}}{T_{min}} - 1} + 4.5$$

Equation 3

### **5.3 SKIDPAD SIMULATION**

The Skidpad event is a simple test of the car's cornering ability. It consists of a figure 8 with two circles of radius 7.625m as shown in figure 7. The driver drives around one loop twice then the next and the average of the 2<sup>nd</sup> and 4<sup>th</sup> loops are taken as the time. This can be reasonably simplified to modelling the car's speed at a constant radius. This event favours light cars with high levels of lateral grip, high downforce and a good suspension setup.

This event is a good candidate for two different methods: live solving and pre-solving. Since it is a steady state manoeuvre a transient simulation would not be a good candidate. Pre-solving is very quick and easy. The Vlimit function that is created when the vehicle model is pre-solved will return the maximum speed that the car is able to travel around the given radius. This process is very quick and will solve in under 0.1s compared to 3 seconds for live solving. Live solving using the constant radius simulation where an upper and lower limit is set, and a bisection method is used. For example: a lower

limit of 10 m/s and an upper limit of 50 m/s is set and, the bisector 30 m/s is tested. If the car can complete a skidpad at 30m/s the upper bisector is chosen 40m/s, if the car cannot corner at this speed the lower bisector is chosen: 35 m/s, etc. This process continues for a set number of iterations until the desired tolerance is achieved. Live solving allows skidpad accuracy to be higher than that required for general track purposes.

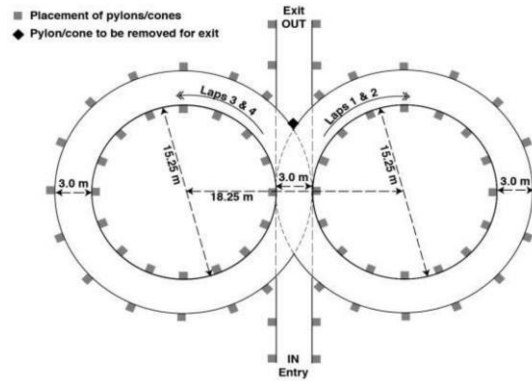


Figure 7: Skidpad Track (SAE International, 2016)

Scoring for the Skidpad event is proportional to the lateral acceleration of the car hence the squared terms in equation 4 below, where  $T_{min}$  is the time of the fastest car,  $T_{max}$  is  $1.25 \times T_{min}$ , and  $T_{your}$  is the time of the car in question.

$$Skidpad\ Time = \frac{2\pi \times 7.625 \times Car\ Width/2}{v_x}$$

Equation 4

#### 5.4 AUTOCROSS AND ENDURANCE SIMULATION

Autocross is a one lap test of the car's short distance speed, compared to Endurance which is a test of the car's speed and reliability of the car. These events tend to favour a car that is well rounded with high lateral and longitudinal grip, light weight, efficient aerodynamics and moderate power. These simulations are too complex to use live solving since it could take 3-4 hours to solve for one car, hence the pre-solving method is used.

The starting point for the lap simulation is calculating the maximum speed that the car is able to achieve at each point on the track only given the curvature and change in curvature using the  $V_{limit}$  function created in the pre-solving section. This produces a speed profile where the local minima are the "critical points" as shown below.

The critical points are then selected and the program calculates the amount of lateral and yaw acceleration required to follow the curvature, then determines the maximum amount of longitudinal acceleration available using the brake, linear interpolation function created in the pre-solve stage. The program then accelerates forward from the critical points using the same techniques as braking

backwards. The transition between accelerating and braking are where accelerating forward and braking backwards speed traces meet.

Both autocross and endurance events are scored based on similar formulas shown below (equation 5 and equation 6) where maximum score is 125 and 225 respectively, and are relative to the fastest time to complete the event.  $T_{min}$  is the time of the fastest car,  $T_{max}$  is  $1.45 \times T_{min}$ , and  $T_{your}$  is the time of the car in question.

$$Autocross\ Score = 118.5 \times \frac{\frac{T_{max}}{T_{your}} - 1}{\frac{T_{max}}{T_{min}} - 1} + 6.5$$

Equation 5

$$Endurance\ Score = 200 \times \frac{\frac{T_{max}}{T_{your}} - 1}{\frac{T_{max}}{T_{min}} - 1} + 25$$

Equation 6

### **5.5 EFFICIENCY SIMULATION**

The efficiency event is run during the endurance event, where fuel consumption and average lap time is calculated for the same run. This event tests the car's ability to complete the endurance event with reasonable lap times while using minimal fuel, hence cars with low weight, small engines and very efficient aerodynamics packages tend to do well.

The powertrain model not only returns the amount of power produced by the engine but also the fuel or electrical energy used, this is used to calculate piecewise to determine the amount of fuel or electrical energy used during the event.

The scoring for this event is based off efficiency factors where the relative amount of  $CO_2$  is compared to the time taken. The score is then calculated based on the efficiency factors as shown in equation 7, where  $CO_{2min}$  is the lowest amount of carbon dioxide emitted by other competitors,  $T_{min}$  is the fastest time during endurance, and  $Eff. Factor_{min}$  is the Fuel Efficiency factor for a car that uses 60.06 kg  $CO_2/100km$  and gets a time of  $1.45 \times T_{min}$ .

$$Eff. Factor = \frac{T_{min}}{T_{your}} \times \frac{CO_{2min}}{CO_{2your}}$$

Equation 7

$$Eff. Score = 100 \times \frac{\frac{Eff. Factor_{min}}{Eff. Factor_{your}} - 1}{\frac{Eff. Factor_{min}}{Eff. Factor_{max}} - 1}$$

Equation 8

## 6.0 DEVELOPMENT OF GRAPHICAL USER INTERFACE (GUI)

All of the code mentioned so far was developed using Matlab scripts, and with good formatting, layout and comments, the entire sim could be used just within the script environment. Even a user with moderate Matlab experience could use these scripts an example of which is shown in figure 8. This would also have made it easier for people to customise the code, extract specific information from it and build on additional features in the future. However, since one of the main aims of the project is to produce a product that anyone can learn and use easily then development of an integrated graphical user interface (GUI) system was necessary.

```

51 % Vehicle Parameters
52 %----- Car Properties -----
53 car.L = 1.55; % L = Wheel base (m)
54 car.CG = 0.55; % CG = Ratio of CG location (CG=0.6, CG is further to rear)
55 car.CG_s_h = 0.30; % CG_s_h = CG height of sprung mass (m)
56 car.CG_ns_h = 0.25; % CG_ns_h = CG height of unsprung mass (m)
57
58 car.m = 280; % m = Vehicle mass (kg)
59 car.m_ns = 12.5; % m_ns = Vehicle unsprung mass (kg)
60
61 car.FTrack = 1.1; % FTrack = Front track width (m)
62 car.RTrack = 1.05; % RTrack = Rear track width (m)
63
64 % Front
65 car.FRollCen_h = 0.08; % FRollCen_h = Front roll centre height (m)
66
67 car.K_f = 22; % K_f = Front corner spring stiffness - per corner (N/mm)
68 car.F_ABB = 80; % F_ABB = Front ABB roll stiffness (Nm/deg)
69
70 % Rear
71 car.RRollCen_h = 0.016; % RRollCen_h = R roll centre height (m)
72
73 car.K_r = 26; % K_r = Rear corner spring stiffness - per corner (N/mm)
74 car.R_ABB = 100; % R_ABB = Rear ABB roll stiffness (Nm/deg)
75

```

Figure 8: Cn-Ay-Ax Limit Surface

The goal of the GUI was threefold: Improve usability of the system, tie in all the functions that have been developed and create a sense of a coherent, and continuous system. The GUI system provides a clear layout with clearly labelled buttons headers and graphs which quickly display information to the user and help them navigate through the tasks and settings required to run the simulator. One example of this is the dynamic event settings panel in the Advanced Sim window. Initially it clearly shows that a track has not been selected and prompts the user to input a points breakdown, after this is done the graphic and list displays the information provided.

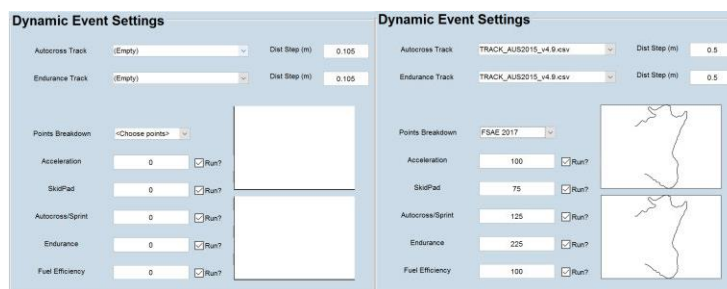


Figure 9: Dynamic Event Settings panel before and after user completion

To tie in all the functions a unifying main menu was made as well as a colour scheme and layout that persisted throughout the program. The background is Monash Motorsport blue at 70% opacity so

that it does not overpower the foreground items, edit boxes, drop down menus and figures all have white backgrounds with interactive buttons having grey back grounds. The interface is split up into panels which have large, bolded headers, titles and labels are bolded and data or statistics have a regular typeface. These small details help the flow of information to the user and allow them to quickly identify important information.

### **6.1 MAIN MENU**

The aim of the main menu is to integrate all aspects of the simulator and to provide a starting place for a new user. It is a simple GUI that acts as a gateway to the various aspects of the Simulator. The first four items generate inputs the inputs for the simulator: tyre model, vehicle model, powertrain model and track. These four components can be used through all of the simulators, this provides a homogeneous work environment and continuity between the simulators.

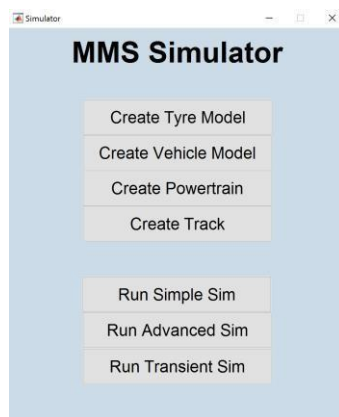


Figure 10: Main Menu GUI

The next three items on the menu launch the different simulators: Simple Sim, Advanced Sim, and Transient Sim. Advanced Sim is the focus of this paper, it is a quasi-steady state competition simulator that employs a four-wheel, 3 degree of freedom vehicle model. This simulator allows a lot of car parameters to be investigated hence needs a lot of care when being set up to ensure that the results are valid, it also takes a not insignificant amount of time to solve. Hence the Simple Sim will also be available, this will employ a quasi-steady state competition simulator that uses a point mass model. This simulator will allow large amounts of runs to be performed in a short amount of time at significantly lower complexity. This will provide a good comparison to the Advanced Sim and allow rough concept specifications and casual users to get good results. The Transient Sim leads to a simulator created by Michael Geist (2015), this is not a competition simulator and will not provide a points estimate. It will, however, provide insight into the time dependent effects of the car, such as rise times and settling times of the vehicle for a step steer input. Both the Simple Sim and Transient Sim are outside the scope of this project and will be completed in future works.

## 6.2 TYRE MODEL GENERATOR

This is the first stage in creating a Vehicle model, creating a tyre model from Tyre Test Consortium data (TTC). One of the challenges of creating a GUI for generating tyre models was creating a way to easily view the data so that the appropriate section can be fitted. This must be done manually since tests like warm up procedures, are difficult to identify with code. Depending on the tests to be analysed different data channels will need to be viewed and identified, as looking at varied scales. It was decided that two plots would be sufficient, each of which could display one of eight channels. These plots could then be controlled with two sliders at the bottom, one for zoom and one for pan. These allow the user to easily select the data of interest and view the two channels simultaneously in the same scale and location.

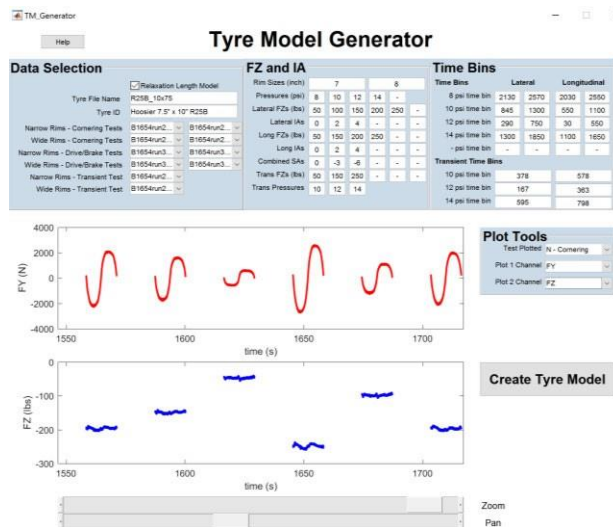


Figure 11: Tyre Model Generator GUI

The user is required to enter three types of data: file names for the tests, values of parameters swept through, and the time bins for the various pressure sweeps. Once this is done the tyre model can be created and automatically saved to the tyre model folder. This will typically take a few minutes. Tyre models should be tested to ensure good fit with the data using the tyre vs data program.

## 6.3 VEHICLE MODEL GENERATOR

This is where the majority of the vehicle specification is done, and the pre-solved limit surface functions are created. There are a large number of parameters to specify at this stage 43 vehicle parameters and 23 solver options. This required them to be split up into logical groups to make it easier for the user to navigate, these groups consist of Vehicle Geometry, Suspension setup, Kinematics, Aerodynamics, Tyres and Transient Model Inputs, for the Vehicle parameters and Vehicle model inputs, Vehicle model options and Vehicle Parameter sweep for the solver options.



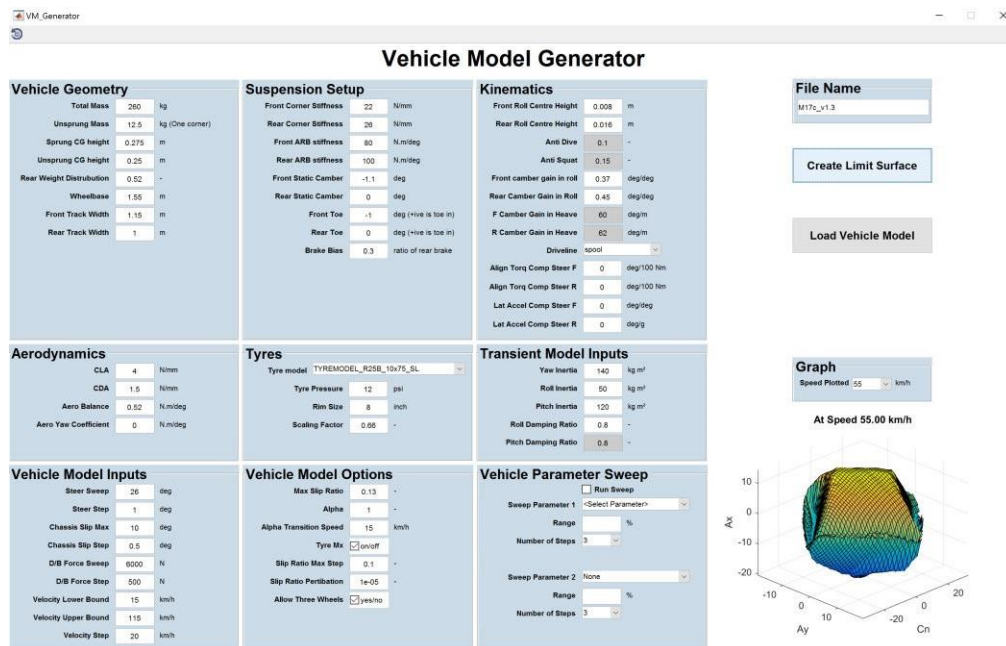


Figure 12: Vehicle Model Generator GUI

Once the vehicle and solver options have been specified, the pre-solved limit surface can be created. This brings up a progress bar which shows how far the program is through the process. This helps give the user a sense of time and expectation as to how long until completion. The progress bar also includes a cancel button so that if the solver is taking too long or the user decides to re-run the solver it is quick and safe to do.

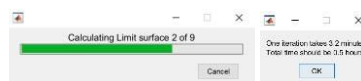


Figure 13: Progress bar

Parameter sweeps with up to two parameters can also be done, once the sweep parameters, range and number of steps have been specified then the solver will create one vehicle model file per step, this can then be batch run in the Advanced Sim. Since this sim is quite complicated and computationally intensive it is better suited to specific investigations rather than broad parameter sweeps, however sometimes a run of incrementally changing parameters can be useful to guide concept decision making.

Since each vehicle model in the standard configuration takes 10-15 minutes to run a large sweep, like a 21x21 sweep could take three days to solve. In most cases this will not be necessary, and either the number of sweeps or vehicle model inputs can be reduced in order to have a more reasonable solve time. For example an 11x11 sweep with half the DB force sweeps will take 8.6 hours, and can be run

overnight. This will be adequate for most applications of this simulator. Specific concepts where several parameters are considered will only need 3-5 runs and can be done at full resolution in around an hour.

#### 6.4 POWERTRAIN MODEL GENERATOR

Often motor specification come in the form of torque vs speed and efficiency vs speed of the motor to transform this into something that is more easily computable for the sim: power vs car velocity and efficiency vs car velocity. This also take into account gearing and calculates the optimal gear to engage for each speed.

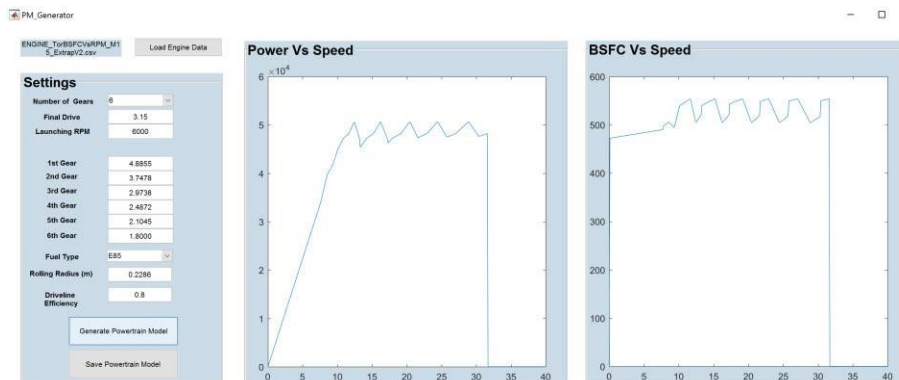


Figure 14: Powertrain Model GUI

The user first loads motor data in the form of a CSV containing motor RPM, Torque and Brake Specific Fuel Consumption (Combustion) or Efficiency (Electric). Then the number of gears in the gearbox, final drive and launch RPM are chosen, these parameters will change between motors and car setup. Next the individual gears ratios are chosen, gears ratios higher than the specified number of gears are ignored. In Formula SAE there are three types of fuel available: 98 RON petrol, E85 Ethanol/Petrol Mix, and electric. 98 RON and E85 have different energy densities and different CO<sub>2</sub> emissions per litre, this is needed for calculating efficiency points. Internal combustion engines have efficiency measured in BSFC as opposed to electric powertrains measure efficiency in the form of work out per electrically energy in. Lastly the rolling radius of the tyre and driveline efficiency are required to calculate power at the wheels and effective gearing.

Once all the specifications are loaded the user can generate the powertrain model. This code takes less than a second to complete and displays the resultant power and BSFC or efficiency vs car speed on the two plots provided. Once the user reviews the result and is happy with it they can save the model to the Powertrain Model folder.

#### 6.5 TRACK GENERATOR

Tracks used for the sim are in the form of an array of curvature at a number of distance steps. This can either be created manually or by using on track data from the car. The Monash Formula SAE car has Bosch 3-axis yaw rate centre that measures yaw rate (Mz), lateral acceleration (Ay), and longitudinal acceleration (Ax). These sensors combined with vehicle speed are enough to re-create a track. The

track generator can either reconstruct the track using lateral acceleration and speed or yaw rate and speed.

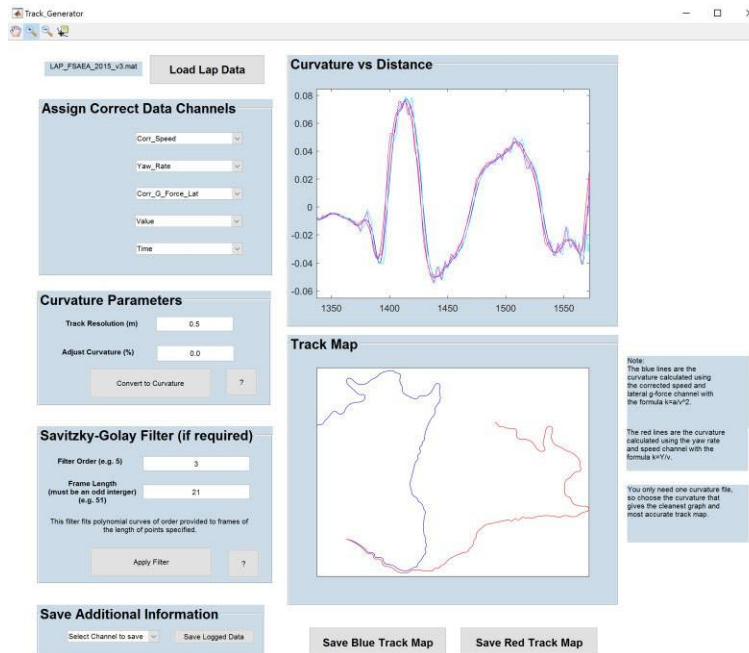


Figure 15: Track Generator GUI

After loading the track data, the user is prompted to select the channels for speed, yaw rate and lateral acceleration, and value and time variables. Once this is done the desired track resolution and curvature adjustment is specified before the data is converted to two sets of curvature: one based off lateral acceleration and one based off yaw rate. When converted, the curvature profiles is displayed in the top plot and the track maps on the lower plot.

An extra, optional process is to filter the data. Two techniques were considered for filtering: a Butterworth filter, and a Savitzky-Golay filter. The Butterworth filter is a low pass filter that attenuates high frequencies, whereas the Savitzky-Golay filter fits a series of polynomials to the data using ordinary least squares to reduce noise. Both filters performed reasonably well however Savitzky-Golay filters were better able to filter out high frequency noise and unwanted data spikes while still maintaining curvature peaks and sharp transitions. Hence the Savitzky-Golay filter was chosen for this part.

The final step is for the user to choose which track map they want to keep. Either the blue track based of Lateral Gs or the red track based of yaw rate.

## 6.6 ADVANCED SIMULATOR

Once the user has created a track, vehicle model, and powertrain model they can now use the Advanced Sim to calculate potential points for required for the chosen concept. The workflow goes starts in the top left corner and proceeds from left to right. First item is selecting a track, as with most of the drop down menus in the sim the first few options are recently selected items for quick access, then there is a load option to load previously created tracks, and finally an option that will open up the appropriate

generator, in this case the track generator. Once the tracks are selected they are displayed on the corresponding plots. Next the points schedule is selected, this can be different depending on competition amendments to the rules.

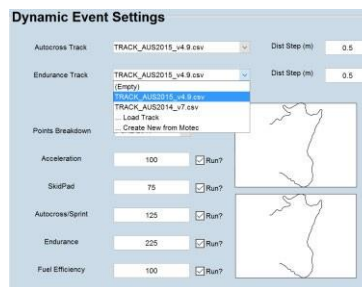


Figure 16: Drop down menu in GUI

The next process involves loading powertrain and vehicle models to create cars. They can either be loaded individually as saved as a new car, or a previously created car file can be loaded which contains the relevant vehicle and powertrain models. Up to three primary cars can be loaded and six additional cars, it is important that there is a range of concepts that can perform well at each event since the points scored is calculated based off the best time or efficiency factor. The primary cars have a summary presented that contains key statistics such as mass, CDA and CLA, a plot showing power and efficiency as well as a small plot of the limit surface.

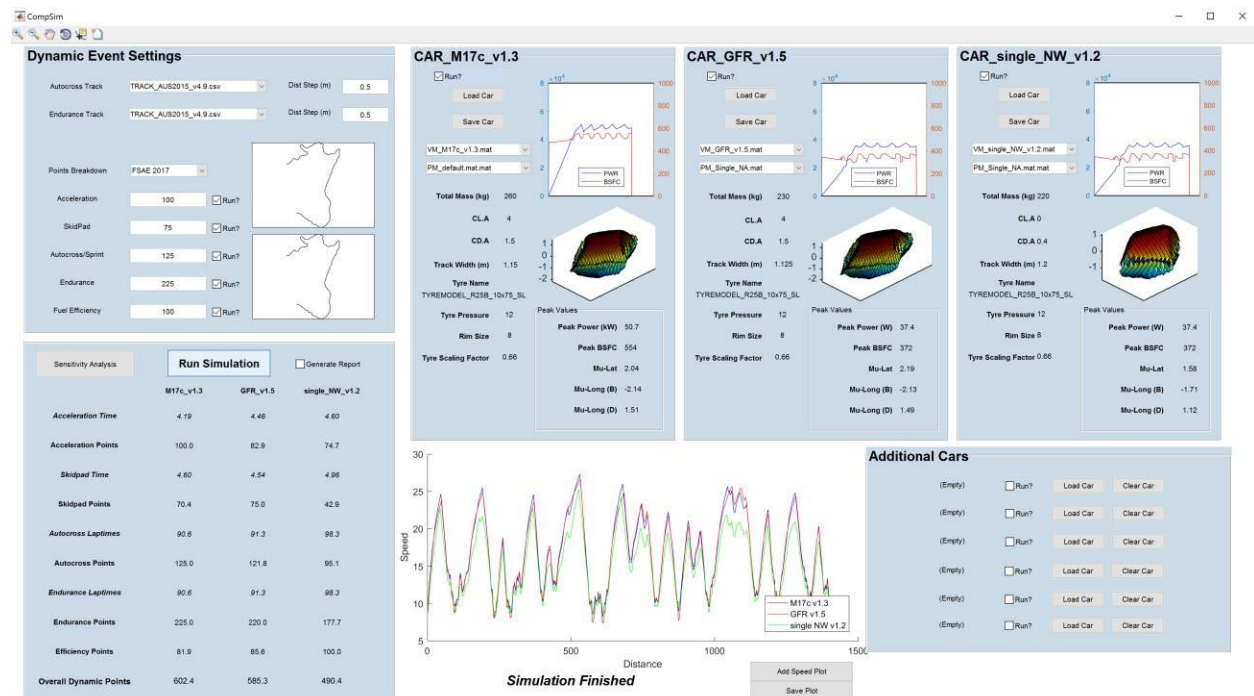


Figure 17: Advanced Sim GUI

Now that the track and cars have been loaded the user can run the simulation, with three cars shown above in figure 17, it takes 15-20s to solve. The results are displayed in the bottom left panel. This show

the time taken for each event and the associated points scored for the three primary cars. To the right of this is a speed trace for the three primary cars for the endurance event for a quick insight into where each concept is gaining or losing time.

For a more detailed investigation the user can choose to generate a report before running the simulation. This creates exports data from the endurance run to an excel spreadsheet, this includes data such as steered angle, chassis slip, lateral acceleration and throttle position. Another option that the user has is to easily compare endurance speed traces between the three primary cars by clicking “Save Plot” underneath the speed plot.

Time	Distance	Speed	Engine RPM	G Force Lat	G Force Long	Yaw Acceleration	AccuPower	Steered Angle	Beta Angle	Dbforce	Throttle Pos
s	m	m/s	rpm	G	G	mm	W	deg	deg	N	%
0.055930749	0.5	8.9396264	0	9.460565079	1.163063263	-14.34182411	4549.734968	19.88020723	1.974935553	511.40246	0.111925
0.055529575	1	9.0042108	0	2.253015764	3.131273485	-11.48477366	4154.988086	1.16290489	-5.404012613	484.78319	0.101597
0.053866679	1.5	9.382176	0	-2.09779426	5.510593794	-6.836119711	12627.70983	-1.638588781	-1.750031874	1402.4227	0.306916
0.052146314	2	9.5884054	0	-4.59660008	6.006816351	-3.454790972	14773.35698	-6.171077592	-2.108624473	1591.5834	0.354805
0.050511071	2.5	9.8988201	0	-5.74779826	6.008533377	-1.099406708	15734.78465	-6.461776194	-1.3110942	1641.022	0.366872
0.049083031	3	10.18682	0	-5.97930225	5.863822141	0.331140134	15498.10971	-5.544704607	-0.634478055	1565.6522	0.350606
0.047770316	3.5	10.466751	0	-5.71997466	5.885174965	0.958933964	15930.64197	-4.526750131	-0.062837894	1563.8484	0.351154
0.046546734	4	10.741892	0	-5.3150135	5.974922522	0.978246121	16522.59036	-3.741616089	0.201022739	1578.5787	0.357877
0.045388301	4.5	11.016055	0	-5.01784424	6.111024448	0.597517595	17351.6746	-3.245560122	0.275310898	1615.3276	0.369419
0.044287363	5	11.289902	0	-4.9781518	6.203671383	0.280856092	18265.40563	-3.027888128	0.270517143	1658.0714	0.384669
0.043257999	5.5	11.559144	0	-5.24140225	6.19676872	-0.453394969	18907.82141	-3.178933446	0.144185206	1674.7551	0.39602
0.042309097	6	11.820078	0	-5.67658926	6.113612957	-0.913575554	19293.39515	-3.468582973	0.081739617	1669.1025	0.400807
0.04142162	6.5	12.070991	0	-6.40958525	5.969683139	-0.835673177	19514.1533	-3.97791296	-0.016878315	1650.9328	0.398559
0.040617953	7	12.309828	0	-6.75271426	5.845707158	-0.017899741	19489.74837	-4.65953272	0.005448467	1614.5939	0.392982
0.039867683	7.5	12.541486	0	-6.66716095	5.814232816	0.653867836	19718.46906	-3.922943635	0.393211092	1601.8477	0.390127

Figure 18: Excel Report

### 6.6.1 SENSITIVITY ANALYSIS

Once the vehicle model has created a sweep it was necessary to create a streamlined way of running them through the Advanced Sim. From within the Advanced Sim profile the Sensitivity Analysis window can be launched, the user is required to select a Vehicle model and Powertrain model to use in the sweep. Sweeps can be performed by a combination of vehicle model parameters, power and efficiency scaling. Vehicle model parameters are embedded within the vehicles models produced when the VM generator performs it’s sweeps, this cannot be changed. Power and efficiency scaling however can be specified within the Sensitivity Analysis window without any pre-solving.

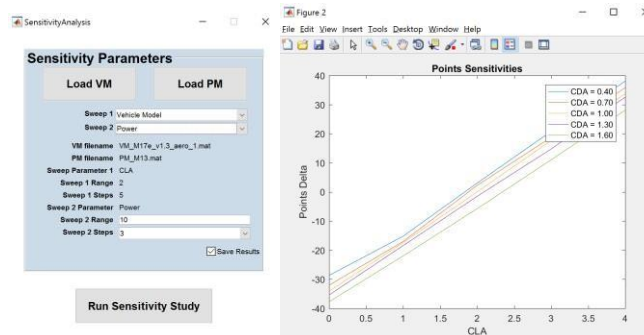


Figure 19: Sensitivity analysis GUI

Once the sensitivity analysis has run through all the parameters, it displays the results a points sensitivity plot as shown in figure 19. If more detailed information is desired the results can be saved to an excel spreadsheet that summarises, the points deltas, individual times and points per event and speed traces for acceleration, autocross and endurance events.

## 7.0 VALIDATION

A very important part of any simulation is validation of the model compared to real life data. In this case it is very important that the sensitivities of various parameters reflect what will actually benefit the car. For example, if power was overstated in the sim, then additional power will be simulated to have a smaller effect than in real life. Also since the car is grip limited more of the time then the sensitivity of longitudinal grip will be inflated, this could lead to the incorrect conclusion that a lighter, underpowered car would be beneficial. Initially there are six basic validations that are needed to ensure the model is performing adequately, furthermore, there are 43 vehicle parameters that can be changed, each one ideally should be validated to ensure correct results. Due to limitations in time, human, and financial resources this would require a large number of tests on track, some of which are impractical to perform. So, investigation of these parameters must rely on a good foundation for the sim and sound engineering and programming understanding.

The first of the parameters to be validated is the lateral grip of the car. There are a lot of controlled parameters that affect performance metric such as: tyres, mass, CG height, track width, suspension setup and a number of uncontrolled parameters: driver performance, track grip, tyre temperature, tyre wear etc. Hence the aim is to get the lateral grip close to data, but some error is acceptable. Low speed cornering is particularly indicative of lateral grip rather than high speed cornering where aerodynamic effects play a large roll. As can be seen in figure 20 there is an excellent correlation between on track and simulated data, the only significant deviations are during the slaloms between 200 and 250 meters. This was found to be the result of slower entry speeds due to poor throttle application of the driver as discussed later on.

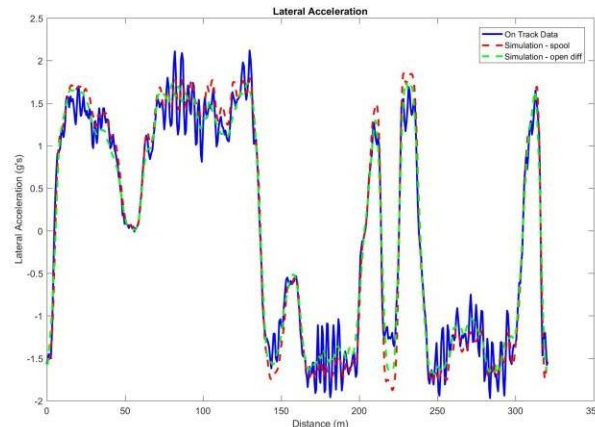


Figure 20: Lateral Acceleration Trace

The next parameters to be investigated was longitudinal grip and power. These two parameters are interrelated such that it is imperative to faithfully represent the balance between the two. At low speeds the powertrain can provide enough torque to overcome the grip of the tyres, hence the car is traction limited, as the speed increases, torque provided to the wheels decreases and downforce increases the tractive potential of the tyres until the car transitions into the power limited region. Matching the car's low speed and high-speed acceleration is the key to validating longitudinal grip and power.

As shown in figures 21 there are some significant deviations in longitudinal acceleration and speed between the data and sim. One region where the simulation does correlate well is in the straight-line section at 50m, the acceleration of the car is consistent with the data through both low speed and high speed regimes, and this suggests that the power and grip of the car are accurately represented. One aspect of note is where the car shifts, this can be seen as a momentary dip in acceleration, since the sim does not have any time dependent features it cannot simulate the lag caused by shifting, however this will have little effect on the usability of the Sim. Another area of interest is the constant radius corner between 150 and 200 meters, the sim accelerates and brakes in rapid succession rather than a smoother acceleration out of the corner. This is again an effect of the simulator not simulating time variant phenomena, in reality quick application of throttle and brake like this generally not possible and a smoother input is used. This affect is also observed through other parts of the track and has little effect on the fuel consumption and speed of the car. The third and most concerning deviation to the data is the increased longitudinal acceleration out of corners particularly at 130m and 200m. This is partially due to the car being fitted with a limited slip differential which is much more complicated to model than a spool or open diff as modelled in the simulator. Another possible explanation, since the car is exiting one corner immediately entering a corner in the opposite direction, that there are significant transient, or stability/control effects that inhibit the driver from reaching the simulator's potential.

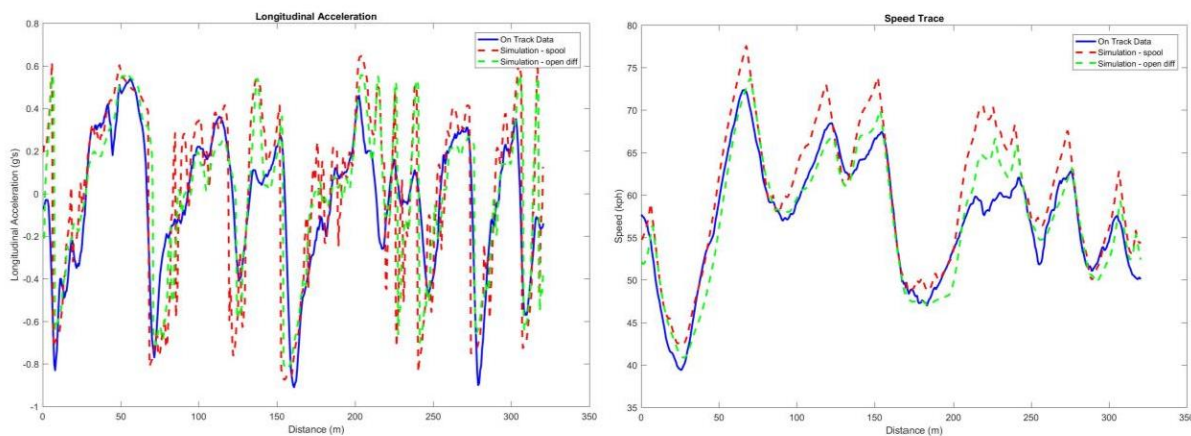


Figure 21: Longitudinal and speed trace

Due to the scope of the project only two driveline options were developed, an open differential where the left and right driven wheels must react the same torque, and a spool where the driven wheels must have the same rotational velocity. In reality the Monash Motorsport car has a limited slip differential

which allows for a difference in speed of the left and right wheels and a difference in torque up until the torque bias limit. An LSD essentially performs somewhere between an open diff and a spool, both of which were tested against the on-track data. The spool was able to start accelerating out of corner earlier than the on-track data and achieved higher speeds before the end of the straight, on the other hand, the open had much more limited grip for combined cornering and drive, this resulted in the car accelerating much later than the actual car where the curvature is lower.

The aerodynamic properties of the car,  $C_{l.A}$  and  $C_{d.A}$  are essential to validate compared to data, the effect of these can be seen in high-speed corners compared to low speed corners and the decrease of longitudinal acceleration at high speed. Both of which are consistent with the data as seen in figures 20 to 21.

The final essential parameter to be validated is the track map. The track maps are based of measured lateral acceleration or yaw rate of the car. Both of which can have substantial noise due to vibrations from the engine and the road. This results in sharp variations in the curvature of the track which is not representative. Filtering can reduce these variations, however over-filtering can cause sharp curvature transitions, such as during slaloms to be softened, or peaks in curvature to be reduced, both of which “open up” the track and favor cars that perform better at high speeds. Hence fine control on the filtering parameters is necessary, it was found that with a distance step of 0.5m a filter of order 3, and a frame length of 13, was the most filtering that could be applied before decreases in curvature maxima and curvature derivatives became significant.

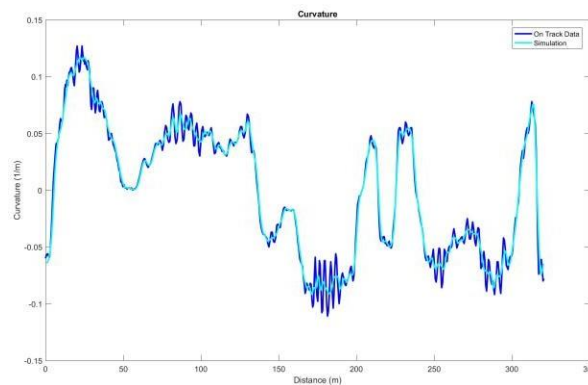


Figure 22: Inverse Corner Radius Trace

## **8.0 USABILITY**

To test if the effectiveness of the layout and graphical user interface of the system several tests were carried out to see how well people with little or no contact with the simulator were able to navigate the menus and perform tasks. Four tasks were asked of six new users, who were timed and compared to an experienced user.



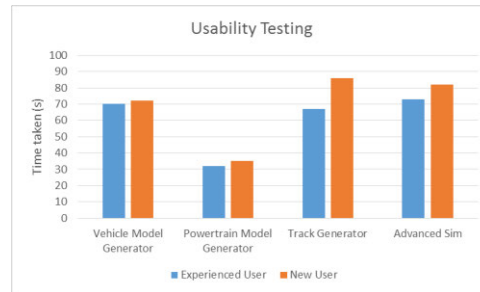


Figure 23: Usability study results

In general, the new users did very well compared to the experience user, especially in the Vehicle Model and Powertrain model where they were the inexperienced users were only 2% and 9% slower respectively. There were also no issues with finding items, loading files and creating models. The users started to have some issues with the Track Generator and Advanced Sim. The Advanced Sim notification window that would inform the user that it was busy loading models or running simulations was not very noticeable. This resulted in some participants not being aware that the program was busy and started attempting the next task too quickly, resulting in confusion and some incorrect clicks. The Track Generator slowed down some of the new users, especially when they needed to select a channel, for lateral G, yaw rate and speed. Further confusion occurred when they were asked to zoom in on a particular part of the curvature plot. They found it difficult to find the zoom tools in the top left corner of the window and those who were less familiar with MATLAB did not draw boxes around the area to be zoomed in rather they clicked on areas of interests which is less efficient.

## **9.0 APPLICATION**

The aim of project is not only to develop a simulator but also to implement it to make concept level specifications for the Monash Formula SAE car. The simulator has started to be used to investigate concepts for the 2017 cars: M17e and M17c, the electric and combustion FSAE cars. Uses so far have included combustion engine selection for M17c, tyre selection for both cars, Aero dynamic balance for the combustion car, energy conservation methods for M17e and an investigation into an aerodynamic package for M17e. In this report the investigation into energy conservation methods and aerodynamics for M17e will be discussed.

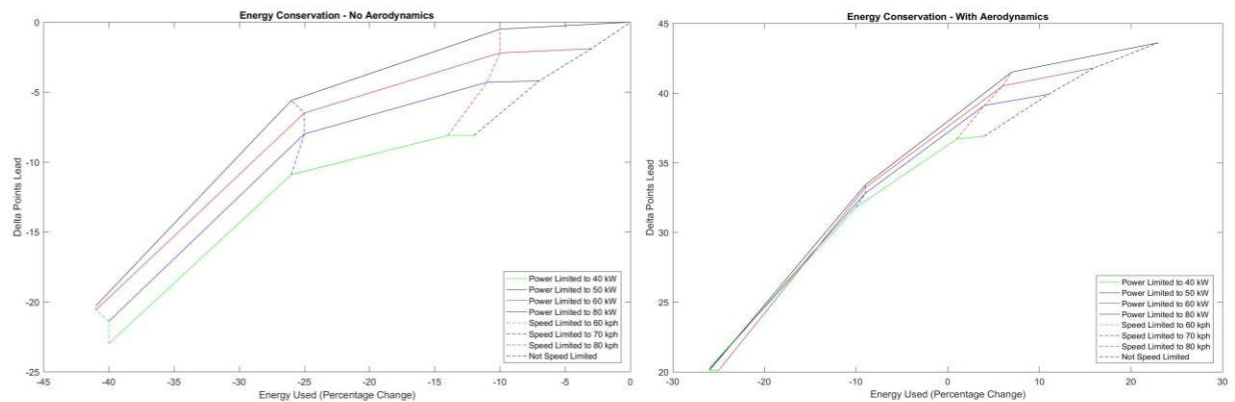


Figure 24: Energy conservation results for No Aero and with Aero

A large part of the mass of an electric formula SAE car comes from the battery or accumulator which can account for as much as 35% of the mass of the car, hence controlling how much energy is used and finding the right balance between speed and economy is crucial. The Advanced Sim has been used to investigate the methods that can be used to reduce the amount of energy used while still scoring well in efficiency and endurance. Two methods were proposed to be investigated: limiting power and limiting top speed. This can be represented by the map shown in figure 24, where a combination of power limiting, and speed limiting has been plotted on axis of percentage change in energy usage and delta points lead. The baseline car is one with no aero and full power (80kW), significant reduction of energy usage, (up to 40%) can be made by limiting the top speed to 60 kph. The plot also shows that the most efficient way to reduce energy consumption for the least number of points is by limiting the top speed of the car. The efficient frontier of this plot is the solid black line representing constant 80kW powertrain with changing power limit. This is where the efficient frontier is defined as the line of points where there are no other options that are unconditionally preferred, i.e. more points for same power consumption or less power consumption for the same points.

The same procedure was done on a model with aero an aerodynamic package based off the M16 combustion car. It was known that adding an aerodynamic package would increase the amount of energy used keeping all else constant since it increases weight and drag, however it was hypothesised that through reducing the power or top speed of the car, the same amount of energy could be used as the baseline but with more points gained. This ended up being correct as shown in figure 25 where it can be shown that reducing the top speed to 70 kph a car with wings would use 10% less energy while scoring 33 more points than the baseline car. If even more energy saving is required, the car limited to 60 kph would use 25% less energy and still make 20 points over the baseline. This provides strong evidence that making an aerodynamic package is a worthwhile concept that has significant gains while still maintaining an energy safety margin.

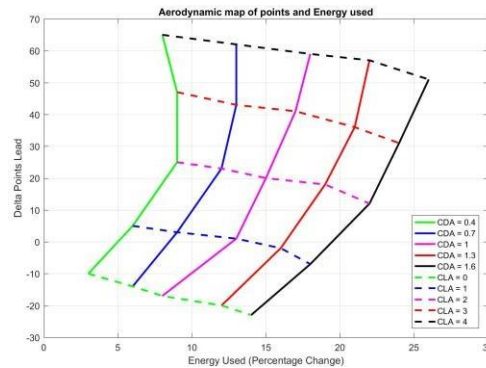


Figure 25: Lateral Acceleration Trace

Another aspect that has been investigated with the Advance Sim, using the vehicle model sweep function, is the effect of lift vs drag of the aerodynamic package as shown in figure 25. This parameter sweep is done compared to an 80 kW, non-limited speed, with no aero baseline, and other than the changes to CLA and CDA the aero concepts carried an additional 15 kg. This particular plot is useful in specifying the aerodynamic package. A limit on energy usage can be set which corresponds to a specific set of CLA and CDA combinations. For example: if a 15% extra energy use limit has been set and an L/D of 2 is achievable then a package of CLA 2 and CDA of 1 is required, if it is found later that 20% extra energy use is acceptable then a CLA of 2.5 and CDA of 1.25 is the target.

## 10.0 FUTURE WORK

There is a large amount of future potential for this simulation system, it has been designed with future expansion and development in mind. Some parts of the simulator are already setup for future work such as the Simple and Transient Sims.

Due to the complexity and computational intensity of the simulator it would be beneficial to have a more robust, easy to use, and faster to solve sim for broad sweeps and as another point of reference for the Advanced Sim. This simulator could be simplified to a point mass or bicycle model, yaw performance could be ignored, and friction ellipses could be estimated using shapes can be manipulated to fit the data. This could reduce the parameter set from 43 to less than ten. Michael Giest's Transient Sim should also be integrated at the same time. Both Sims would need to be coded so they work seamlessly with the existing code and feature set.

As stated earlier the inclusion of a limited slip differential model would greatly improve the fidelity of the output and give the use the ability to investigate the effect of diff pre-load. Since accurate energy usage estimation is integral to the design of an electric car, the powertrain model could be developed with the addition of torque and efficiency relative to throttle response and accumulator discharge. This could also be used in the combustion powertrain model.

Additional Vehicle model feature could also be added for systems that are likely that the team will be investigating in the future. Drag Reduction System (DRS) is where the flaps on the front and rear wing open to reduce drag on the straight away improving straight line speed and efficiency, this could be

implemented into the vehicle model setup. Four-wheel drive is a concept that will be explored once the team has gained more experience with the electric car, this would require modification to the vehicle model generator and investigation into methods of distributing power to each of the four wheels. The last major concept that could get implemented in the vehicle model is four-wheel steer, where the rear wheels do some of the turning, allowing for tighter turns and greater response.

## **11.0 CONCLUSION**

The aim of this project was to develop a competition points simulator that was easy to use and could be used by future Monash Motorsport team members. Building on the work of Russouw, a tyre model that now is able to work with a range of tyres, and test data as well as fitting a function for the moments  $M_x$ ,  $M_z$  and relaxation length. The new tyre model was then taken and implemented in a vehicle model based of the work of Geist, it was then thoroughly overhauled to achieve a 300x improvement in solve time. With the use of Davis Taylor's limit surface pre-solving function, and lap time solver the vehicle model was used in conjunction with Bett's code to create a competition simulator.

An integrated GUI environment was then created to tie together all the aspects of the simulator and make it easy to use and intuitive for future team members. The usability has been tested and was found to be extremely good with new users only 10-20% slower than an experienced user.

Finally, the system was implemented to provide useful data for concept level decision making and has been used to investigate specifications that up until now the Monash motorsport team has not had tool to be able to do so.

## **12.0 REFERENCES**

- Bett, M (2015). Development of a dynamic event simulation tool for formula-SAE. Monash University
- Geist, M (2015). Development and analysis of a vehicle model for formula-SAE. Monash University
- Milliken, W., & Milliken, D. (1995). Race Car Vehicle Dynamics. Warrendale: Society of Automotive Engineers.
- Pacejka, H. (2012). Tyre and Vehicle Dynamics. Delft: Butterworth-Heinemann
- Patton, C. (2013). Development of Vehicle Dynamics Tools for Motorsports. Corvallis: Oregon State University.
- Russouw, M. (2014). Development of a vehicle dynamics simulation for formula-SAE. Monash University
- SAE International (2016). 2017-18 Formula SAE rules
- Smith, C. (1978). Tune to Win. Fallbrook, CA: Aero Publishers.
- Webb, S. (2012). Vehicle Simulation to drive Formula SAE design decisions