GreenBath Racing

GBR-02: Individual Technical Report

Vehicle Simulation

16/05/19

Author: Tom Hunt
Candidate Number: 11181
Word Count: 5197

Supervisors: Andy Green
Jos Darling

Summary

This report details the development of a simulation of a Shell Eco-marathon vehicle for the Green Bath Racing team. A model for the vehicle motion is derived which includes a powertrain model and approximates: cornering scrub, aerodynamic drag, weight, and rolling resistance. An accurate representation of an arbitrary track is presented, made by fitting a set of piecewise continuous cubic bézier splines through track coordinate data. The implementation of the numerical methods used is described in detail with great consideration taken for computational efficiency. The simulation is used to optimise a burn and coast driving strategy giving an efficiency of 179 km /kWh, an improvement of 13 km / kWh over the un-optimised case. A sensitivity analysis study is demonstrated, showing how the simulation could be used to better inform design choices and resource focus. The vehicle was found to be most sensitive to the battery efficiency.

Contents

1 Introduction

1.1 The Shell Eco-marathon

The Shell Eco-marathon is an international competition challenging students to design a vehicle to complete a track using the least amount of energy [1]. Green Bath Racing is a team based at the university of Bath aiming to compete in the urban concept category of the Shell Eco-marathon in 2020. This class of the competition is new to Green Bath Racing and so the vehicle must be designed from scratch. As the team is in the early design stage the design is relatively flexible. This represents the best time in the design process to optimise the vehicle using from fundamental engineering principles as the design isn't constrained by previous iterations.

1.2 Model Based Design

A useful tool to aid design optimisation is a physically accurate model of the vehicle. Creating a model alongside the design of the vehicle allows rapid virtual prototyping to qualitatively validate design ideas [2]. The model is continually refined as more understanding is gained about the vehicle. This approach can improve product quality and reduce development time [3].

For the purpose required by Green Bath Racing, the model should be able to be interrogated with different design points to test if they will produce a net benefit to the vehicle in terms of energy usage over the specific track that will be driven on. Additionally, the model should be used to experiment with and optimise the driving strategy employed at the competition event in order to extract the maximum out of a given vehicle.

This report details the implementation of a simulation of the vehicle and the optimisation of the driving strategy employed at the track based on the simulation.

2 Vehicle Model

2.1 Overview

The vehicle model is derived from the free body diagram shown in figure 1 where s is a scalar coordinate representing distance along the track. The vehicle is pushed along the track by a propulsive force from the driven wheel, P. The resistive forces acting on the vehicle are: weight, F_w, aerodynamic drag, F_a, rolling resistance, F_{rr}, and cornering scrub, F_c.

Resolving the forces in the direction of travel gives the vehicle equation of motion, equation 1.

$$P - F_w - F_a - F_c - F_{rr} = m\frac{d^2s}{dt^2} \tag{1}$$

2.2 Weight

The component of weight force in the direction of travel is given by equation 2.

$$F_w = mg \cdot \sin(\theta) \tag{2}$$

2.3 Aerodynamic Drag

The aerodynamic drag force is given by equation 3 [4]. The coefficient of drag, C_d, depends on the angle of the vehicle to the relative wind direction but for this model is assumed to be constant. The other variables are: air density, ρ, the frontal area of the vehicle, A, and the vehicle speed, $\frac{ds}{dt}$.

$$F_a = \frac{1}{2} \cdot \rho \cdot C_d \cdot A \cdot \left(\frac{ds}{dt}\right)^2 \tag{3}$$

2.4 Rolling Resistance

Rolling resistance is caused by hysteresis in rubber of the tyres [5]. In pneumatic tyres rolling resistance is a function of tyre pressure. For this model it is assumed that the tyre is inflated to the nominal pressure of a tyre and so the coefficient of rolling resistance, μ, is given by the tyre supplier and is constant. The model used is given by equation 4 [6].

$$F_{rr} = mg \cdot \mu \cdot \cos(\theta) \tag{4}$$

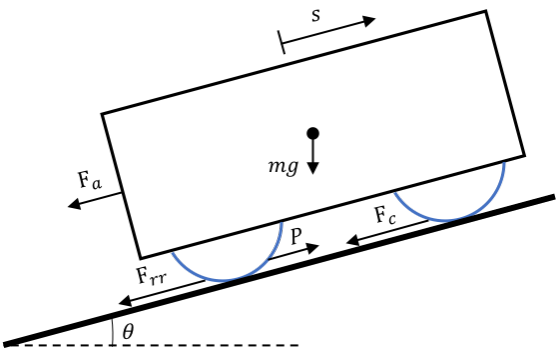

Figure 1: Free body diagram of the vehicle showing all forces considered in the vehicle model.

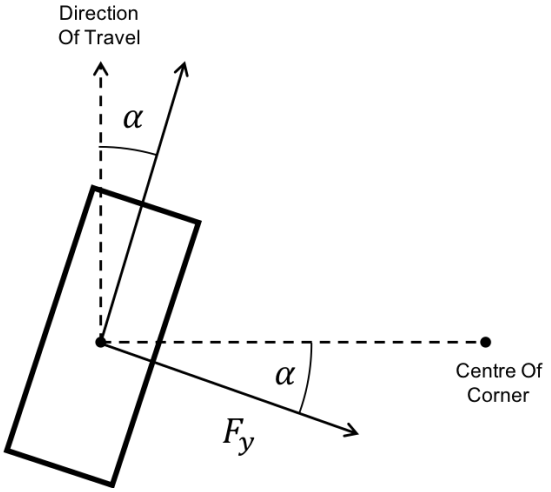

Figure 2: Diagram showing how cornering force is generated by a single wheel. F_y is the force generated and α is the slip angle of the tyre.

2.5 Cornering Forces

To turn the vehicle around corners on the track the front tyres are used create a lateral force. The force required from the steering tyres in order to turn the vehicle is approximated by the circular motion equation 5 [7]. This simplified model assumes steady steering at a constant radius, R, ignoring the yaw moment to overcome inertia.

$$F_{\text{centripetal}} = \frac{m\left(\frac{ds}{dt}\right)^2}{R} \tag{5}$$

Figure 2 is a diagram of the tyre model used to approximate the scrubbing force experienced through corners. The tyre generates a lateral force through there being a difference in angle between the steering angle and the actual wheel angle, known as the slip angle, α [8].

For a general vehicle, lateral force is generated by all four wheels [9]. Each wheel will contribute a different amount to the total lateral force. If the lateral force generated by the vehicle is not great enough for the force requirements of a manoeuvre, the vehicle will slide.

To simplify this model it is assumed that lateral force is generated only by the front wheels and that they both contribute evenly to lateral force with the same slip angle. Additionally, it is assumed that the vehicle will not exceed the grip limits of the tyres and so does not slide. The second assumption should not reduce the accuracy of the model as sliding would be detrimental to energy consumption so the vehicle should not be driven close to the limit of sliding.

Based on the first assumption, the generated lateral force, F_y, is resolved in the radial direction, giving equation 6.

$$F_y \cdot \cos(\alpha) = \frac{m\left(\frac{ds}{dt}\right)^2}{R} \tag{6}$$

Resolving in the direction of travel and rearranging gives the cornering scrub force, F_c, seen in equation 7.

$$F_c = \tan(\alpha) \cdot \frac{m\left(\frac{ds}{dt}\right)^2}{R} \tag{7}$$

The slip angle of a wheel is a function of both the tyre characteristics and the vertical load on the tyre [10]. This is a complicated relationship and is highly dependant on the vehicle suspension setup. Physical testing of tyres such as that done by Andrew Butler for the 2018 Green Bath racing vehicle [11] is required in order to find the slip angle for a given lateral force required and vertical load. For simplicity a fixed slip angle value is selected. Experimental results showed that the slip angle for a similar tyre to the one used by the 2020 team is on the order of magnitude of 2^o for lateral forces similar to those expected by the new vehicle.

3 Powertrain Model

3.1 Motor Model

A brushed DC motor was selected to power the Green Bath Racing vehicle. Figure 3 shows the equivalent electrical circuit for this motor type. The torque produced is given by equation 8, where i represents the current in the circuit and k_T is the motor torque constant which is provided in the motor specification sheet [12].

$$T_m = k_T \cdot i \tag{8}$$

The motor induces a back emf when spinning that is proportional to the motor speed, ω_m. The proportionality constant, k_ω, is given in motor specification sheet along with the terminal resistance, R, and the terminal inductance, L. A dynamic model for the motor electrical circuit is given in equation 8 where V is the voltage applied across the motor terminals [13].

$$V = iR + L\frac{di}{dt} + k_\omega \cdot \omega_m \tag{9}$$

When the vehicle is stationary, the motor is not spinning and no back emf is induced. For motors with low terminal inductance, nearly all voltage applied in this scenario would be dissipated only by

the resistance. This would cause the current to significantly increase to potentially damaging levels. To stop this a control technique known as 'soft start' is employed [14]. This slowly increases the maximum applicable voltage until the current is limited by the induced back emf. This is modelled by assuming the electrical power is limited by the peak power available from the battery, Q. Substituting the relationship $Q = i \cdot V$ into equation 9 and neglecting the inductance term gives equation 10 which is solved for V, giving the maximum safe applicable voltage.

$$V^2 - k_\omega \cdot \omega_m \cdot V - Q \cdot R = 0 \tag{10}$$

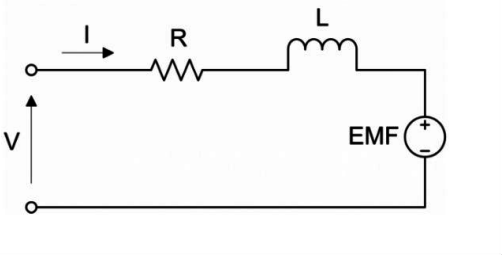

Figure 3: Equivalent circuit for a brushed DC electric motor [13].

Figure 4: An example free wheel hub [15]. The outer ratchet is driven via a chain and transfers torque to the inner race via pawls.

3.2 Transmission

Figure 5 shows the layout of the transmission used in the Green Bath Racing vehicle. A freewheel such as the one shown in figure 4 is used so that the resistive torque from the motor when no is voltage applied is not exerted on the driven wheel.

The physical freewheel engages when the speed of the driving outer race matches the speed of the inner race. To model this it is assumed that the inertia of the transmission and motor is sufficiently small such that it takes negligible time to accelerate the motor, transmission and outer race up to the speed of the inner race when power is applied to the motor. When the power is removed the same assumption is applied so that the motor comes to a stop immediately.

The transmission is modelled as one stage with an overall reduction ratio of r_t. Transmission efficiency, η_t, is assumed constant and modelled as reduction in torque transmitted betwee the input and output of the transmission. When the free wheel is engaged the torque at the wheel, T_w, is given by equation 11.

$$T_w = T_m \cdot r_t \cdot \eta_t \tag{11}$$

The linear vehicle propulsive force, P, is given by equation 12.

$$P = \frac{T_w}{r_t} \tag{12}$$

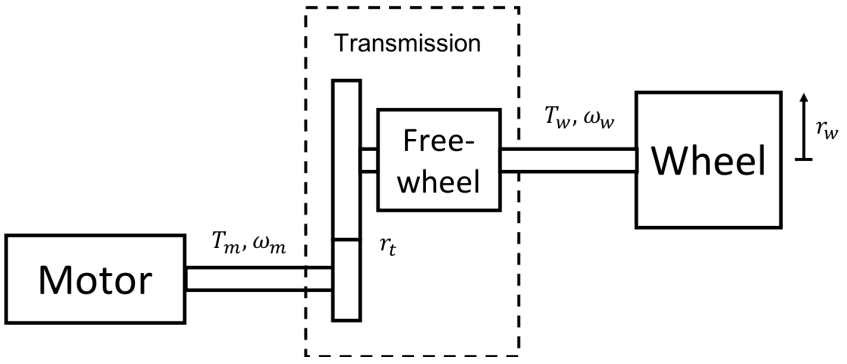

Figure 5: Schematic of the Green Bath Racing vehicle transmission.

4 Complete Model

The vehicle model is decomposed into a coupled system of first order ordinary differential equations. The state vector, $\vec{y}$, is given in equation 13. The state variables are: the scalar coordinate representing the distance travelled by the vehicle, s, the time derivative of distance representing the speed of the vehicle, $\frac{ds}{dt}$, and the motor current, i.

$$\vec{y} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} s \\ \frac{ds}{dt} \\ i \end{bmatrix} \tag{13}$$

The first order decomposition of the system is the derivative of the state vector with respect to time. This is given in equation 14.

$$\frac{d\vec{y}}{dt} = \frac{d}{dt}\begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \frac{ds}{dt} \\ \frac{d^2s}{dt^2} \\ \frac{di}{dt} \end{bmatrix} \tag{14}$$

Equation 1 is rearranged for $\frac{d^2s}{dt^2}$ to give equation 15.

$$\frac{d^2s}{dt^2} = \left(\frac{1}{m}\right) \cdot (P - F_w - F_a - F_c - F_{rr}) \tag{15}$$

Equation 9 is rearranged for $\frac{di}{dt}$ to give equation 16.

$$\frac{di}{dt} = \left(\frac{1}{L}\right) \cdot (V - iR - k_\omega \cdot \omega_m) \tag{16}$$

Equations 15 and 16 both contain terms which are functions of more than just the state variables and instead take information about the location of the vehicle on the track as input instead. For example the angle of the track to horizontal is used to calculate the weight force, F_w, in equation 15. This varies at different locations around the track. The applied motor voltage in equation 16 will depend on the driving strategy employed which again will be decided based on the location of the vehicle on the track.

5 The Track

5.1 Overview

The vehicle is modelled as a particle constrained along a path. The location of the vehicle is tracked as it moves so that the vehicle model can make calculations based on the location. This approach reduces the dimensionality of the model from three dimensional world coordinates to a single coordinate representing the distance of the vehicle along the path, s.

5.2 Representing The Track

The track is represented by a set of piecewise continuous parametric curves. A diagram of this is shown in figure 6. Each curve element is of arbitrary type, allowing complicated three dimensional paths to be easily created. The vehicle exists on only one curve element any given time, with a record of the current element kept. The position of the vehicle in three dimensional space, $\vec{x}$, is then fully defined by the curve element number and the curve parameter, λ.

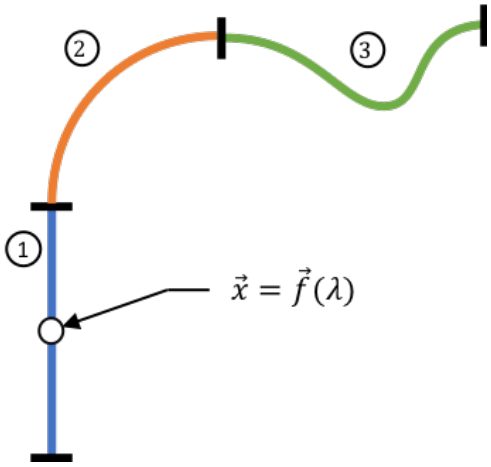

Figure 6: A path made up of a set of piecewise parametric elements.

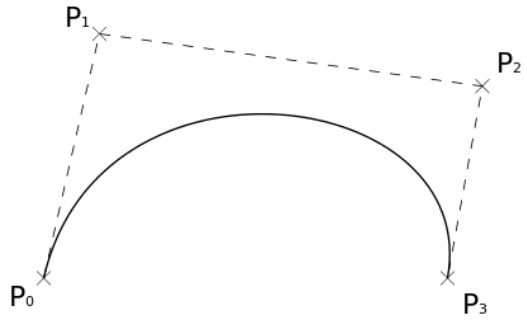

Figure 7: A sample cubic Bézier spline element [16].

To represent the Shell Eco-marathon track, cubic Bézier spline elements were used. This element type can replicate complex curvature with a relatively simple definition. Cubic Bézier splines are defined by two end points, $\vec{P}_0$ and $\vec{P}_3$, and two interior control points $\vec{P}_1$ and $\vec{P}_2$. The curve definition is given in equation 17 [17].

$$\vec{f}(\lambda) = (1-\lambda)^3 \cdot \vec{P}_0 + 3(1-\lambda)^2\lambda \cdot \vec{P}_1 + 3(1-\lambda)\lambda^2 \cdot \vec{P}_2 + \lambda^3 \cdot \vec{P}_3 \qquad (0 \leq \lambda \leq 1) \tag{17}$$

Curves of this type were fitted through known coordinates on the 2018 and 2019 tracks with second order geometric continuity using the method described in source [17]. The method assumes there is no curvature at the two ends of the track but ensures smooth curves within them. Figure 8 shows a map of the 2019 Shell Eco-marathon track to scale. The piecewise cubic Bézier spline representation of the same track is overlaid on top in figure 9 showing the effectiveness of this method at capturing complicated curves.

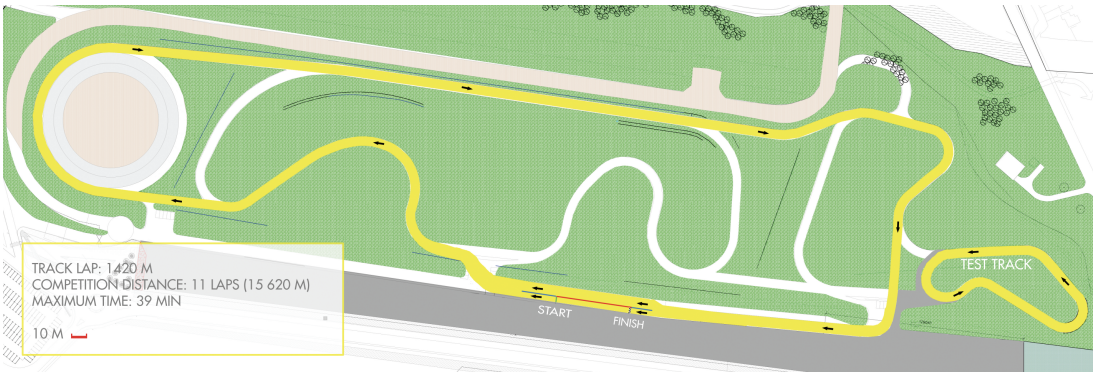

Figure 8: Track map of the Shell Eco-marathon Europe track for 2019 [18].

Figure 9: The piecewise cubic Bézier spline representation of the Shell Eco-marathon 2019 track overlaid on the track map.

5.3 Model Interface

The position of the vehicle on the track is given in terms of the element number and the curve parameter, λ. However, the vehicle model is defined in terms of a distance, s. This distance is equivalent to the arc length of a parametric curve from the start of the curve to the vehicle position. This is defined for an arbitrary parametric curve element, $\vec{f}$, and parameter value, α, in equation 18 [19] [20].

$$s = \int_0^{\alpha} \left| \frac{d\vec{f}}{d\lambda} \right| d\lambda \tag{18}$$

As s is the state variable used in the simulation, the problem of interest is finding α from s. This is stated formally in equation 19, which is solved for α [19].

$$\int_0^{\alpha} \left| \frac{d\vec{f}}{d\lambda} \right| d\lambda - s = 0 \tag{19}$$

Solving equation 19 numerically would add significant overhead to the simulation. Instead values of s at regular intervals in λ are calculated using equation 18 and stored for each curve element. These are then linearly interpolated to approximate α given s.

Note: $\frac{d\vec{f}}{d\lambda}$ is given for different parametric curve types in appendix A.

5.4 Curvature and Incline

Equations 2 and 7 from the vehicle model require the angle between the direction of travel and horizontal, θ and the radius of curvature of the track, R.

For an arbitrary curve, $\vec{f}(\lambda)$, the unit tangent vector, $\vec{T}(\lambda)$, is given by equation 20 [21].

$$\vec{T}(\lambda) = \left(\frac{1}{|\frac{d\vec{f}}{d\lambda}|} \right) \cdot \frac{d\vec{f}}{d\lambda} \tag{20}$$

θ is shown on figure 10 as the angle between $\vec{T}$ and the unit vector, $\vec{H}$, constructed from the horizontal components of $\vec{T}$ normalised. Letting $\vec{Z}$ be a vertical unit vector, θ is defined in radians by equation 21.

$$\theta = \frac{\pi}{2} - \arccos(\vec{T} \cdot \vec{Z}) \tag{21}$$

The curvature vector, $\vec{\kappa}$, of a parametric curve, $\vec{f}(\lambda)$, is defined in equation 22 [21].

$$\vec{\kappa} = \frac{\frac{d\vec{f}}{d\lambda} \times \frac{d^2\vec{f}}{d\lambda^2}}{\left| \frac{d\vec{f}}{d\lambda} \right|^3} \tag{22}$$

The radius of curvature of the track used to calculate cornering scrub, R, is taken as the magnitude of the horizontal components of κ.

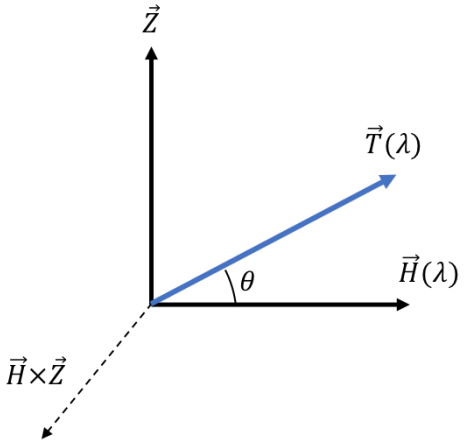

Figure 10: The angle between the direction of travel and horizontal, θ.

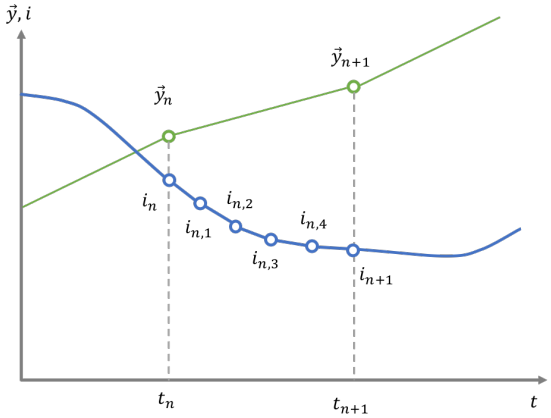

Figure 11: Diagram showing how the current is integrated using smaller inner steps.

6 Implementation

6.1 Problem Formulation

The vehicle model is expressed as a system of first order ordinary differential equations in equation 14. This forms an initial value problem of the form given in equation 23. The initial conditions, $\vec{y}_0$, are all zero assuming the track representation starts at the starting point of the real track. This system is integrated numerically in order to solve for the state. The integration is stopped when the vehicle has driven the specified number of laps or a time limit is reached.

$$\frac{d\vec{y}}{dt} = f(\vec{y}, t) \qquad \vec{y}(0) = \vec{y_0} \tag{23}$$

6.2 Integration Structure

As the track is modelled as a finite set of curve elements, special care has to be taken to handle the case when the vehicle has reached the end of a lap. To achieve this the integrator calls a function in the model after each integration step that allows the model to loop . the state variable s back to zero if the new value is beyond the limits of the track. For robustness the reverse case was also implemented. The amount of track covered by the vehicle is also updated here and a lap counter incremented accordingly so that integration can be stopped when the lap counter reaches the specified number of laps.

6.3 Verification of Track Representation

It was identified that the vehicle and track model derived is a subset of a larger group of simulations that constrain a point mass on a path. With this realisation the vehicle model was abstracted to become a general simulator of this problem type. This was tested against simple simulations with analytical solutions in order to validate the implementation of the simulator. The test cases are explained in greater detail in appendix B. The tests included: a particle undergoing only self weight falling down a vertical track and a vertical circular track, a particle with self weight and a constant propulsive force travelling up an inclined slope, a particle experiencing self weight and friction converging to the bottom of a vertical circle, and a particle in free fall experiencing aerodynamic drag and reaching terminal velocity. All tests agreed with their expected analytical results and so confidence was gained in the simulation implementation.

Note: The tests implemented are made using the unit testing framework *unittest*, in python. These are located in **GBRSim/src/Model/tests/**.

6.4 Separating Motor Equation

Through experimentation it was found that the motor state equation, equation 16, is a stiff differential equation. The implication of this is that the timestep of the numerical integration has to be prohibitively small in order for the integration to converge [22]. To improve computational efficiency, the model was split up into the vehicle model and the motor model. The vehicle model updates at a large outer timestep, whilst the motor model takes many small inner timesteps based on the values of the model at the last outer timestep. A diagram of this is shown in figure 11. The inner steps are denoted with a second subscript. The motor speed, ω_m, is calculated at state $\vec{y_n}$ and is held constant for all inner steps. The last value of current is i_{n+1} and is used to calculate the motor torque in state $\vec{y_{n+1}}$. By separating the model this way the overheads from finding the curve parameter and calculating the cornering curvature are not incurred as frequently so computation time is greatly improved. This model was found to be stable using a fourth order Runge-Kutta integration scheme with a outer timestep of 0.25 seconds and an inner timestep of 0.001 seconds.

6.5 Numerical Integration Scheme

Different numerical integration schemes were implemented in order to select the most computationally efficient for the model. These are summarised in table 1 below. All methods implemented were explicit schemes due to their relatively simple implementation. Methods with higher order truncation error can be increased in timestep whilst maintaining accuracy so fewer steps are required [23]. However, these methods require more function evaluations per step. Therefore, a balance must be found. Adaptive timestepping methods: Runge–Kutta–Fehlberg (RKF45) and Dormand–Prince, calculate solutions at two different error orders in order to approximate the truncation error of each step. The timestep is then adjusted based on the estimated error. These were implemented as it was thought that when the vehicle

accelerated from a standstill smaller timesteps would be required initially until it tends to an equilibrium where the extra calculations are not required.

Method	Type	Step Size	Order	Evaluations	Time (s)	Integration Error
Euler	Explicit	Fixed	1st	1	9.28	0.157 %
Classical Runge-Kutta	Explicit	Fixed	4th	4	9.75	0.128 %
Runge–Kutta–Fehlberg	Explicit	Adaptive	4th	6	14.43	0.004 %
Dormand–Prince	Explicit	Adaptive	4th	7	10.60	0.008 %
RK8	Explicit	Fixed	8th	10	16.12	-

Table 1: Comparison of different numerical integration schemes for the vehicle model.

The column titled 'Time (s)' in table 1 is the time a simulation of the vehicle around the 2019 track at constant throttle took to compute. The method was applied to the outer integration with a timestep of 1 second. The motor inner timestep was 0.001 seconds and the classical Runge-Kutta method was used. As expected methods with more function evaluations per step tended to take longer. The calculated energy used over the lap was compared for each method to the 8th order RK8 method to estimate the error in integration. As expected the low order Euler method performed worse. The adaptive schemes performed better than the fixed step methods. The classical Runge-Kutta scheme was chosen for optimisation as it was relatively quick with better accuracy than the Euler method.

Note: The different integrator implementations are found under **GBRSim/src/Integration/**.

6.6 Driving Strategy

Four different driving strategies were implemented in order to allow comparison of strategies.

The simplest strategy is constant throttle application. Evidently this is the quickest method of getting around the track but uses the most energy.

The second strategy was to apply full throttle whilst the speed is below a threshold. This strategy could be used to keep the motor operating at its most efficient speed.

Strategy three is to apply full throttle until the vehicle reaches a maximum speed. When reached, the vehicle is allowed to slow down to a minimum speed. Full throttle is then applied and the cycle continues. Whilst promising, this strategy is difficult if a track contains a mixture of curves and gradients since a maximum speed at one place on the track may be too fast for another place.

The final strategy that was used for optimisation, see section 7, was to apply the throttle based on the position of the vehicle on the track. This is shown diagrammatically in figure 12. The values: ds_i and T_i were implemented such that they can be optimised.

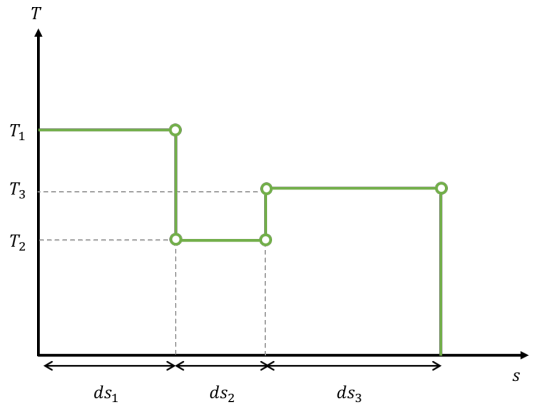

Figure 12: Graph of throttle position, T, with distance along track, s.

Note: The different driving strategies are found in **GBRSim/src/Strategy/controller.py**.

6.7 Results Explorer

To aid development a MATLAB app was created to interactively display the results of a simulation. This made troubleshooting the simulation easier as the faults could be easily identified. A screenshot of the app is shown in figure 13. On the left of the app is a three dimensional plot of the vehicle coordinates. This can be coloured by any of the tracked variables. The three plots on the right of the app show time series results of a simulation for a chosen variable. The time range plotted for the app is controlled by the slider in the bottom left.
Note: The app is located in **GBRSim/src/Results/ResultsExplorer.mlapp**.

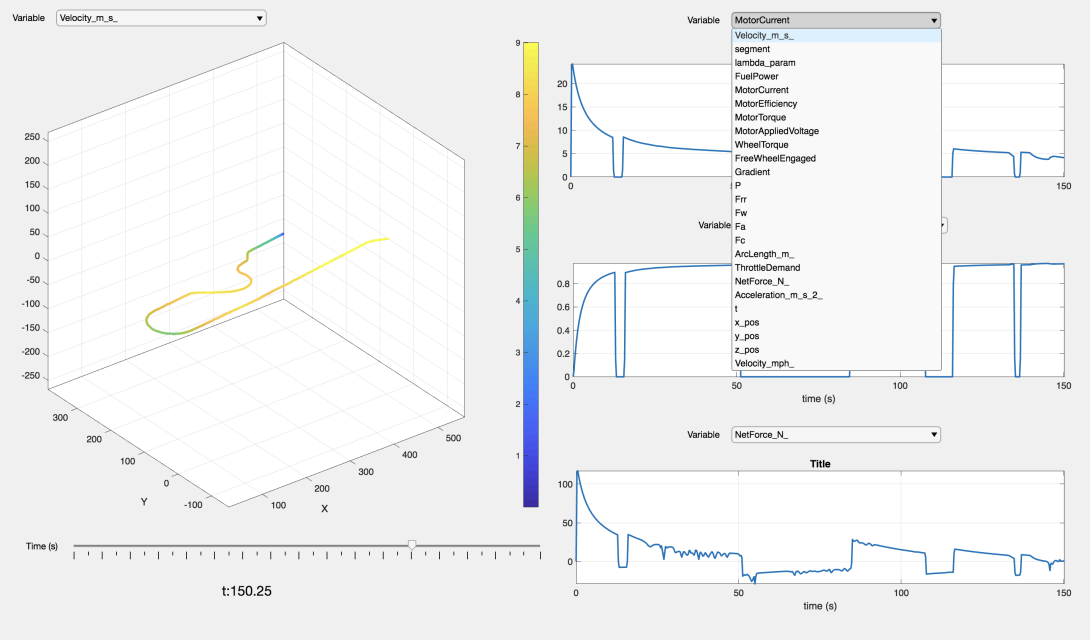

Figure 13: Screenshot of the MATLAB app developed to interactively analyse simulation results.

6.8 Programming Approach

The simulation was developed using a highly modular approach. This approach greatly improves reusability and maintainability of the code [24]. A more detailed model of any aspect of the vehicle is simple to implement in the simulator as each force is its own function so that only a single function would require changing.

The program was written in an object oriented structure. This helps improve the modularity and readability of the code [25]. The inheritance property of object oriented programming was made use of to make generalise the simulator as discussed in section 6.3.

The version control system GitHub was used whilst developing the code to keep track of every modification. If a mistake was made, previous versions could be restored. This system very commonly used in software development as it enables easy and safe collaborative working on large projects [26]. Given the modularity of the code, this property of version control would allow a member of the team to modify a certain aspect of the model and seamlessly integrate it into the master code.

Note: The complete program can be found at the url: **https://github.com/tomjhunt13/GBRSim/**

6.9 Program Structure

The main functionality of the simulation is split across six folders in a modular fashion. These are described below.

6.9.1 Integration

Contains each integrator along with unit tests to verify their implementation. The file **Integrator.py** contains the parent 'Integrator' class which defines how the individual integrator implementations are

interacted with.

6.9.2 Model

Contains all of the physical models implemented and tests to verify their implementation. The class inheritance structure for different model classes is given in figure 14. The benefit of this structure is that the functionality required to interface with an 'Integrator' instance is only implemented once in the parent class 'Model' so that child simulations can easily be implemented. The same benefit is found in the 'ConstrainedParticle' class. Child simulations can easily be produced to simulate a constrained particle. This concept is used in the unit tests for the physics model, found in **/Model/tests/Models.py**.

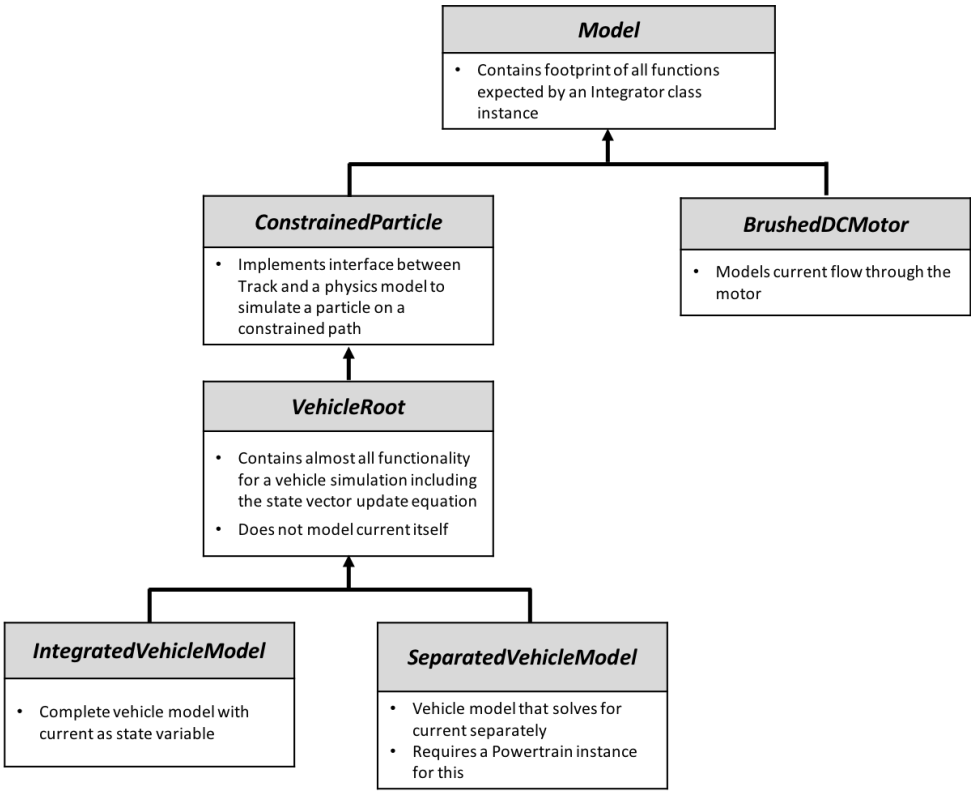

Figure 14: Class inheritance structure of the 'Model' class.

6.9.3 Optimisation

Contains abstraction layer around imported SciPy optimisers as well as the sensitivity analysis class. The 'VariableManager' parent class contains functionality to track the variables that are being optimised to simplify the creation of optimisation and sensitivity studies. With this implementation only one line of code is required to add a parameter to the study:

```
StudyObject.add_variable(variable_name, variable_pointer, minimum_value, maximum_value)
```

6.9.4 Results

Contains post processing functionality. Location of MATLAB app discussed in section 6.7.

6.9.5 Strategy

Contains specific wrapper classes around the vehicle simulation and optimisation classes for further convenient optimisation study creations. Contains scripts which were used to perform optimisation and sensitivity studies.

6.9.6 Track

Contains implementations for everything discussed in section 5.

7 Optimisation

7.1 Overview

An important part of the Shell Eco-marathon competition is the driving strategy used around the track. Teams make extensive use of 'Burn and Coast' strategies where the throttle is applied for a short time and then released for the vehicle to coast [27]. The timing of these throttle on periods is crucial to get the best out of the vehicle. This has been optimised using the simulation developed.

7.2 Problem Definition

The burn and coast throttle strategy defined in section 6.6 was used. To reduce the dimensionality of the optimisation some simplifications were made. The throttle application was fixed in a specific order so that for two burns it was: on, off, on, off. The length of the burn and spacing between burns were optimised for.

The transmission ratio was included as a variable in the optimisation as for some strategies requiring short throttle on durations more torque at low speed may be required.

Table 2 gives the vehicle parameters used for this optimisation. These would be updated with more accurate values for the Green Bath Racing vehicle when the initial design is complete.

Variable	**Value**	**Unit**
Vehicle Mass	80	kg
Driver Mass	70	kg
Transmission Efficiency	95%	-
Battery Efficiency	90%	-
Coefficient of Drag	0.25	-
Frontal Area	1.26	m^2
Driving Wheel Radius	0.279	m
Coefficient of Rolling Resistance [28]	0.0015	-

Table 2: Parameter of vehicle optimised.

The variables to be optimised are labelled $\vec{v}$. The solution of the optimisation is then defined as a set of values for $\vec{v}$ that minimise a cost function, $C(\vec{v})$.

7.3 Cost Function

The objective of the competition is to complete the track using the minimum energy. The urban concept class vehicles must come to a complete stop at the end of every lap. Therefore, the total energy consumed is equal to the energy consumed over one lap $E(\vec{v})$ multiplied by the number of laps.

The rules for the 2019 Shell Eco-marathon event state that competitors must complete 11 laps in 39 minutes [18]. This equates to 213 seconds maximum per lap which is reduced to 208 seconds to give a 5 second float. The maximum allowable time is labelled $t_{\max}$. The time taken to complete the lap is labelled t_{end}. These parameter are used to punish simulations that overrun $t_{\max}$.

In order to not allow strategies where the vehicle remains still and consumes no energy, the amount of track completed must be factored into the cost function. The total length of track is labelled L and the distance covered s_{end}.

The full cost function is given in equation 24. The parameter k is empirically tuned to scale the impact of the completion distance. The selected values of k is 3.6×10^6.

$$C(\vec{v}) = E(\vec{v}) \cdot (1 + \text{Max}(0, (t_{\text{end}} - t_{\max})) + k \cdot \text{Max}(0, (L - s_{\text{end}})) \tag{24}$$

7.4 Method

An optimiser abstraction class was created to interface between the simulation and pre-existing optimisers from the python library *SciPy*. The optimiser selected is a Dual Annealing scheme. It is a stochastic optimiser used to approximate the global optima of functions with a large number of variables. The full

method is defined in source [29].

Note: The Optimiser abstraction class is located at **GBRSim/src/Optimisation/Optimiser.py**.

7.5 Results

Figure 15 shows the optimised burn locations for up to six burns over a lap of the 2019 track. Orange is used to represent throttle application. The full results for each simulation are given in appendix C.

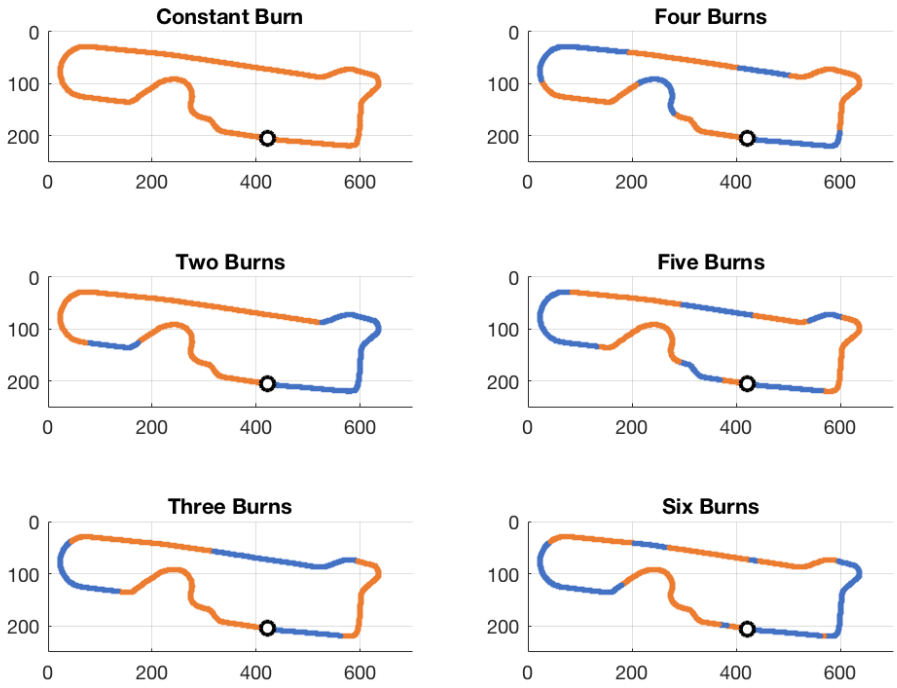

Figure 15: Optimised burn and coast strategies for up to six burns on the 2019 track. Orange represents throttle application. The circle represents the starting position of the vehicle.

Table 3 compares the efficiencies of each of the strategies. There does not appear to be a relationship between the number of throttle applications and the efficiencies.

It can be seen from the figure that the most efficient strategies are those that have a long period without throttle application at the end of the lap. This is intuitive as input energy isn't wasted on excess kinetic energy that will be lost when the vehicle has to stop at the end of the lap.

All of the optimised strategies take almost the full time allowed. Again this intuitive as excess input energy isn't spent on completing the lap too quickly.

Burns	Time (s)	Efficiency (km / kWh)
1	169.5	166
2	206.5	177
3	207.5	173
4	207.5	177
5	207.0	166
6	207.5	179

Table 3: Efficiencies and lap completion times for optimised burn and coast strategied.

8 Sensitivity Analysis

8.1 Overview

Sensitivity analysis is a tool used to prioritise future development. The simulation is interrogated to find out how sensitive the performance of the vehicle is to certain parameters so that future work can be focused on improving the most important performance parameters. The true sensitivity of the performance would consider the performance of the optimum driving strategy for each design point. A finite difference approximation is used to compute the gradient of performance. Several neighbouring design points are required to compute the approximation so including an optimisation in the gradient calculations would take an impractical amount of time. Instead the sensitivity analysis was conducted for the constant throttle driving strategy.

Three design points were analysed corresponding to the vehicles in table 4. Values not in the table are taken from table 2.

Variable	Value
Vehicle Mass	150 kg
Transmission Efficiency	70%
Battery Efficiency	80%
Coefficient of Drag	0.4

(a) Poorly performing vehicle.

Variable	Value
Vehicle Mass	100 kg
Transmission Efficiency	85%
Battery Efficiency	90%
Coefficient of Drag	0.25

(b) Averagely performing vehicle.

Variable	Value
Vehicle Mass	40 kg
Transmission Efficiency	98%
Battery Efficiency	95%
Coefficient of Drag	0.15

(c) Strongly performing vehicle.

Table 4: The three vehicles analysed.

8.2 Results

The affect of a 1% difference from the stated value of each variable on the performance of the vehicle is presented below. The variables are: the vehicle mass, m, transmission efficiency, η_t, battery efficiency, η_b, and the vehicle coefficient of drag, C_d. For all vehicles, the battery efficiency is the most sensitive parameter and the vehicle mass is the least sensitive parameter. This implies that design choices that increase mass for improvements in the other parameters will often pay off. The magnitude of sensitivity increases with the base performance of the vehicle. This represents that improvements become increasingly difficult to attain as the base vehicle becomes stronger.

Poorly Performing Vehicle

$$\begin{aligned} 1\% \text{ change in } m &\implies 0.107 \quad \text{km / kWh} \\ 1\% \text{ change in } \eta_t &\implies 0.340 \quad \text{km / kWh} \\ 1\% \text{ change in } \eta_b &\implies 0.748 \quad \text{km / kWh} \\ 1\% \text{ change in } C_d &\implies 0.170 \quad \text{km / kWh} \end{aligned}$$

Averagely Performing Vehicle

$$\begin{aligned} 1\% \text{ change in } m &\implies 0.118 \quad \text{km / kWh} \\ 1\% \text{ change in } \eta_t &\implies 0.411 \quad \text{km / kWh} \\ 1\% \text{ change in } \eta_b &\implies 1.042 \quad \text{km / kWh} \\ 1\% \text{ change in } C_d &\implies 0.184 \quad \text{km / kWh} \end{aligned}$$

Strongly Performing Vehicle

$$\begin{aligned} 1\% \text{ change in } m &\implies 0.0816 \quad \text{km / kWh} \\ 1\% \text{ change in } \eta_t &\implies 1.392 \quad \text{km / kWh} \\ 1\% \text{ change in } \eta_b &\implies 1.577 \quad \text{km / kWh} \\ 1\% \text{ change in } C_d &\implies 0.734 \quad \text{km / kWh} \end{aligned}$$

9 Group Contribution

The model based design approach allows qualitative validation of design ideas in the early phases of design. The vehicle simulation was used to generate realistic time series motor current data in order to validate the thermal capabilities of the battery. Early simulations showed the optimum transmission ratio for a given vehicle varies between tracks and driving strategies. This information was used to decide on a modular transmission design so that the transmission ratio could be easily changed. Lastly, the simulation was used to validate that the chosen motor would be suitable.

10 Future Work

10.1 Driving Line Optimisation

A proposed method for optimising the driving line alongside the throttle strategy is shown in figure 16. A track coordinate $\vec{P}_i$ is defined as being an amount, β_i, between a position on the outside of the track and the inside of the track. The track is fitted through all the coordinates as in the current simulation. β_i can then be added as a variable in the optimisation study.

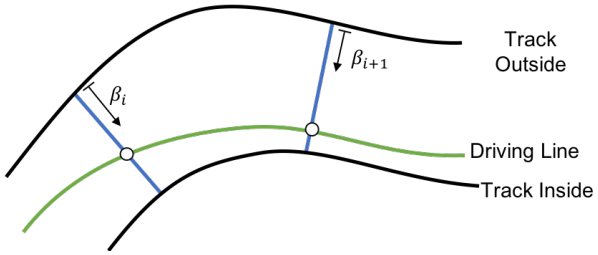

Figure 16: A proposed method to optimise the driving line.

10.2 One Wheel Drive Model

The Green Bath Racing vehicle is one wheel drive. As the driving force from the driven wheel is offset from the centre line of the vehicle, a yaw moment is exerted on the car. A steering input is required to counter this effect. This could be implemented in into the cornering scrub calculations.

11 Conclusion

This report detailed the development of a simulation of a Shell Eco-marathon vehicle. An ordinary differential equation describing the vehicle motion was derived. This included approximations of cornering scrub, aerodynamic drag, weight, and rolling resistance forces. The motor and transmission of the vehicle were also modelled and a complete coupled system of equations modelling the full system produced. An accurate representation of an arbitrary track was created by fitting a set of piecewise continuous cubic bézier splines through track coordinate data. The accuracy of the fitting method was visually verified against a map of the 2019 Shell Eco-marathon track. The model was computationally optimised by breaking up the state update equation and solving for current with inner timesteps. A comparison of the performance of different numerical integration schemes was performed and the classical 4th order Runge-Kutta scheme selected. The simulation was used to optimise a burn and coast driving strategy. A strategy was found that gave an efficiency of 179 km /kWh. This was 13 km / kWh more efficient than a constant throttle application. A sensitivity analysis study was executed to show how the simulation can be used to inform design choices and resource focus. The vehicle was found to be most sensitive to the battery efficiency. As the Green Bath Racing vehicle is developed further, so should the simulation so that every design iteration can be qualitatively validated before expensive resources are committed with increasing confidence.

References

[1] "Shell eco marathon," 2019. [Online]. Available: https://www.shell.com/make-the-future/shell-ecomarathon.html

[2] K. CHAUHAN, "Why is model-based design important in embedded systems?" 2018. [Online]. Available: https://www.einfochips.com/blog/why-is-model-based-design-important-in-embedded-systems/

[3] "Why use model-based design?" 2019. [Online]. Available: https://uk.mathworks.com/solutions/model-based-design.html

[4] "Rocket aerodynamic forces." [Online]. Available: https://www.grc.nasa.gov/www/k-12/rocket/presar.html

[5] "What is tire rolling resistance?" 2019. [Online]. Available: https://www.bridgestonetire.com/tread-and-trend/tire-talk/tire-rolling-resistance

[6] "Rolling friction and rolling resistance," 2008. [Online]. Available: https://www.engineeringtoolbox.com/rolling-friction-resistance-d_1303.html

[7] "Circular motion and satellite motion - lesson 1 - motion characteristics for circular motion," 2019. [Online]. Available: https://www.physicsclassroom.com/class/circles/Lesson-1/Mathematics-of-Circular-Motion

[8] J. Darling, "Vehicle dynamics lecture notes: Tyres," 2019.

[9] H. B. Pacejka, "Tyre and vehicle dynamics," 2006. [Online]. Available: https://doi.org/10.1016/B978-0-7506-6918-4.X5000-X

[10] "How do race car tyres/tires work? slip angle, contact patch deformation etc," 2015. [Online]. Available: https://www.youtube.com/watch?v=y5Y-w4zGW00t=999s

[11] A. Butler, "Eco-marathon handling and rolling resistance," 2018.

[12] "Dc motor speed: System modeling," 2017. [Online]. Available: http://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeedsection=SystemModeling

[13] "Dc motor speed: Voltage and torque relationships," 2019. [Online]. Available: https://www.precisionmicrodrives.com/content/dc-motor-speed-voltage-and-torque-relationships/

[14] L. Zongwang and S. Fuyan, "Soft-start circuitry for dc motor," 2011. [Online]. Available: https://ieeexplore.ieee.org/document/6057599

[15] "Oversimplified: Freewheels," 2015. [Online]. Available: https://www.missionbicycle.com/how-do-freewheels-work

[16] M. Sigler, "Sample image of a bézier curve," 2006. [Online]. Available: https://commons.wikimedia.org/wiki/File:Bezier_curve.svg

[17] "Smooth bézier spline through prescribed points," 2012. [Online]. Available: https://www.particleincell.com/2012/bezier-splines/

[18] Shell, "Shell eco-marathon europe 2019 track map," 2019. [Online]. Available: https://www.facebook.com/groups/semeurope/

[19] M. WALTER and A. FOURNIER, "Approximate arc length parametrization," *SIBGRAP*, pp. 143–150, 1996. [Online]. Available: https://www.visgraf.impa.br/sibgrapi96/trabs/pdf/a14.pdf

[20] P. Dawkins, "Arc length with parametric equations," 2018. [Online]. Available: http://tutorial.math.lamar.edu/Classes/CalcII/ParaArcLength.aspx

[21] "Curvature formula," 2019. [Online]. Available: https://www.khanacademy.org/math/multivariable-calculus/multivariable-derivativescurvature

[22] "Stiff differential equations," 2003. [Online]. Available: https://uk.mathworks.com/company/newsletters/articles/stiff-differential-equations.html

[23] T. Young and M. J. Mohlenkamp, "Lecture 31: Higher order methods," 2017. [Online]. Available: http://www.ohiouniversityfaculty.com/youngt/IntNumMeth/lecture31.pdf

[24] "Modular approach in programming," 2019. [Online]. Available: https://www.geeksforgeeks.org/modular-approach-in-programming/

[25] R. Half, "4 advantages of object-oriented programming," 2017. [Online]. Available: https://www.roberthalf.com/blog/salaries-and-skills/4-advantages-of-object-oriented-programming

[26] "What is version control," 2019. [Online]. Available: https://www.atlassian.com/git/tutorials/what-is-version-control

[27] M. Love, "Running on empty: on the road with the shell eco-marathon," 2017. [Online]. Available: https://www.theguardian.com/technology/2017/may/28/shell-eco-marathon-fuel-endurance-racing-prototype-london-martin-love

[28] "Tire uc 95/80r16 - hs code:401110," 2013. [Online]. Available: http://www.eshopsem.com/boutique/product.php?id_product=75

[29] "scipy.optimize.dual_annealing," 2019. [Online]. Available: https://scipy.github.io/devdocs/generated/scipy.optimize.dual_annealing.html

A Arc Length Formulation

The arc length of a parametric curve between the start of the curve at $\lambda = 0$ and a point on the curve at $\lambda = \alpha$ is given in equation 18. It is repeated below:

$$s = \int_0^\alpha \left| \frac{d\vec{f}}{d\lambda} \right| d\lambda \tag{18}$$

This has been analysed for line elements, circle elements and cubic bézier spline elements.

A.1 Line Element

The parametric definition of a line is given in equation 25.

$$\vec{f} = \vec{P_0} + \lambda \cdot (\vec{P_1} - \vec{P_0}) \tag{25}$$

The derivative of this with respect to λ is given in equation 26.

$$\frac{d\vec{f}}{d\lambda} = (\vec{P_1} - \vec{P_0}) \tag{26}$$

Defining the magnitude of this vector as L, the solution to the definite integral in equation 18 becomes equation 27.

$$s = L \cdot \alpha \tag{27}$$

A.2 Circle Element

A horizontal circle is parametrically defined in equation 28.

$$f(\vec{\lambda}) = r \cdot \begin{bmatrix} \cos(\lambda \cdot 2\pi) \\ \sin(\lambda \cdot 2\pi) \\ 0 \end{bmatrix} \tag{28}$$

The derivative with respect to λ of a parametric circle is given in equation 29.

$$\frac{d\vec{f}}{d\lambda} = 2\pi r \cdot \begin{bmatrix} -\sin(\lambda \cdot 2\pi) \\ \cos(\lambda \cdot 2\pi) \\ 0 \end{bmatrix} \tag{29}$$

The magnitude of this vector is $2\pi r$. The solution to equation 18 for a circle element is given in equation 30.

$$s = 2\pi r \cdot \alpha \tag{30}$$

A.3 Cubic Bézier Spline Element

A cubic bézier spline is parametrically defined in equation 17, repeated below:

$$\vec{f}(\lambda) = (1-\lambda)^3 \cdot \vec{P_0} + 3(1-\lambda)^2\lambda \cdot \vec{P_1} + 3(1-\lambda)\lambda^2 \cdot \vec{P_2} + \lambda^3 \cdot \vec{P_3} \qquad (0 \leq \lambda \leq 1) \tag{17}$$

The derivative of this with respect to λ is given in equation 31.

$$\frac{d\vec{f}}{d\lambda} = -3(1-\lambda)^2 \cdot \vec{P_0} + 3(1-4\lambda+3\lambda^2) \cdot \vec{P_1} + 3(2\lambda-3\lambda^2) \cdot \vec{P_2} + 3\lambda^2 \cdot \vec{P_3} \tag{31}$$

Unlike line and circle elements, solving equation 18 analytically for the bézier curve is not straightforward and instead is solved numerically. To do this equation 18 is rearranged into the first order ordinary differential equation given in equation 32 which can be solved between $0 \leq \lambda \leq \alpha$ with the initial condition $s(0) = 0$.

$$\frac{d\vec{s}}{d\lambda} = \left| \frac{d\vec{f}}{d\lambda} \right| \tag{32}$$

B Validation Cases

The following are some of the implemented unit tests in **GBRSim/src/Model/tests** used to validate the 'ConstrainedParticle' class. More complicated tests are found in the specified directory.

B.1 Particle dropped from top of vertical track

Variable	Value
m	1 kg
g	10 m/s^2

The Track: Four line elements stacked vertically. Total height of 5m.

The Test: Particle dropped from top of track.

Analytical Solution: From constant acceleration equations:

$$\begin{aligned} s &= ut + \frac{1}{2}gt^2 \\ u &= 0 \\ \implies t &= \sqrt{(}2\frac{s}{g}) \\ s &= 5 \\ a &= 10 \\ \implies t_{\text{end}} &= 1 \end{aligned}$$

Assertion: Assert $s_{\text{end}} = 5$

B.2 Particle popping up from end of vertical track

Variable	Value
m	1 kg
g	2.5 m/s^2

The Track: Four line elements stacked vertically. Total height of 5m.

The Test: Particle with initial velocity upwards of 5 m/s upwards from bottom of track.

Analytical Solution: From constant acceleration equations:

$$\begin{aligned} v &= u + at \\ v(t_{\text{end}}) &= -u \\ \implies t_{\text{end}} &= 2\frac{u}{a} = 4 \end{aligned}$$

Assertion: Assert $s_{\text{end}} = 4$, Assert particle reaches top of track.

C Individual Burn and Coast Solutions

C.1 Constant Throttle Application

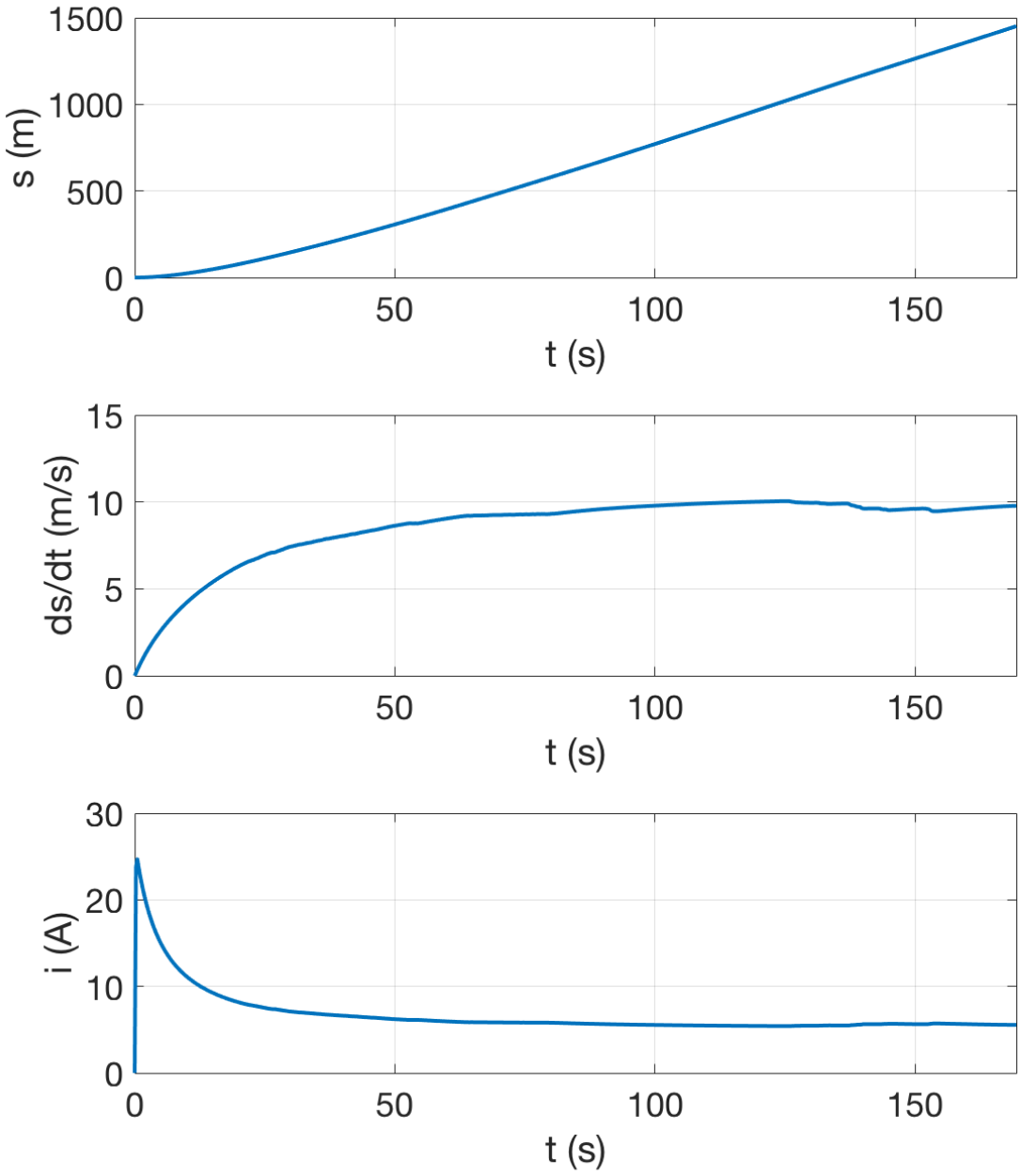

Figure 17: Simulation result for continual throttle application on the 2019 track.

C.2 Two Burns

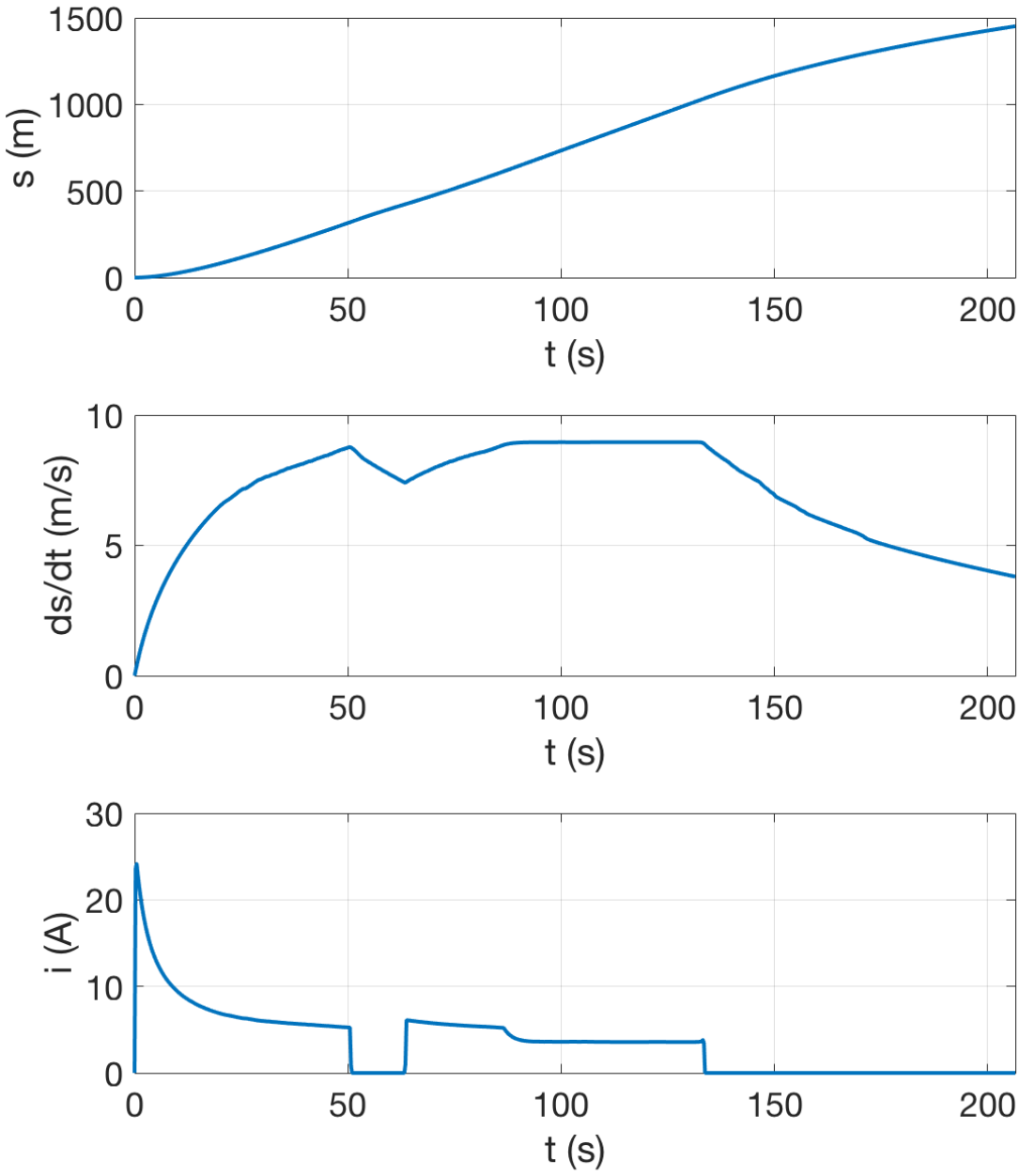

Figure 18: Simulation result for a two burn strategy on the 2019 track.

C.3 Three Burns

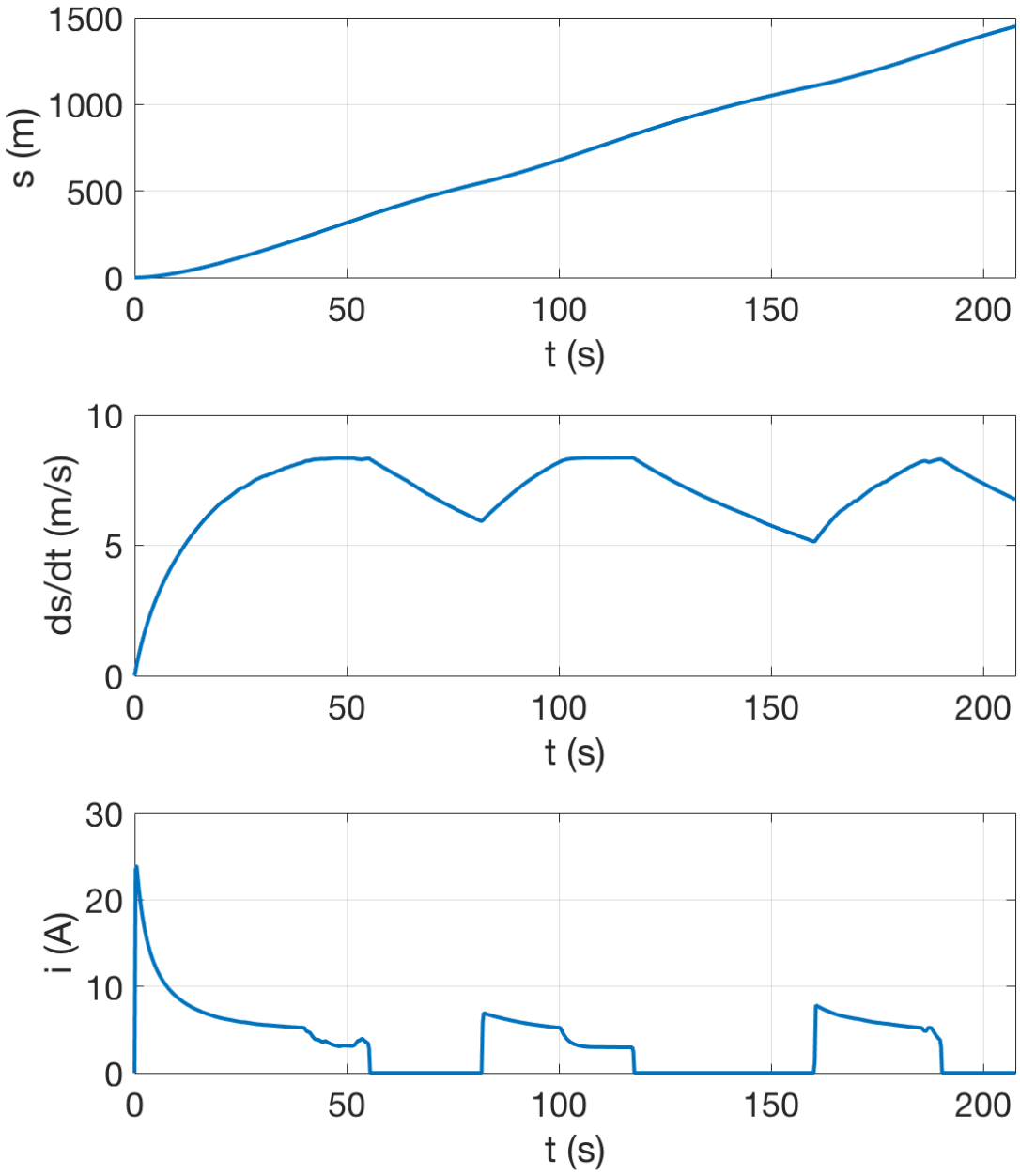

Figure 19: Simulation result for a three burn strategy on the 2019 track.

C.4 Four Burns

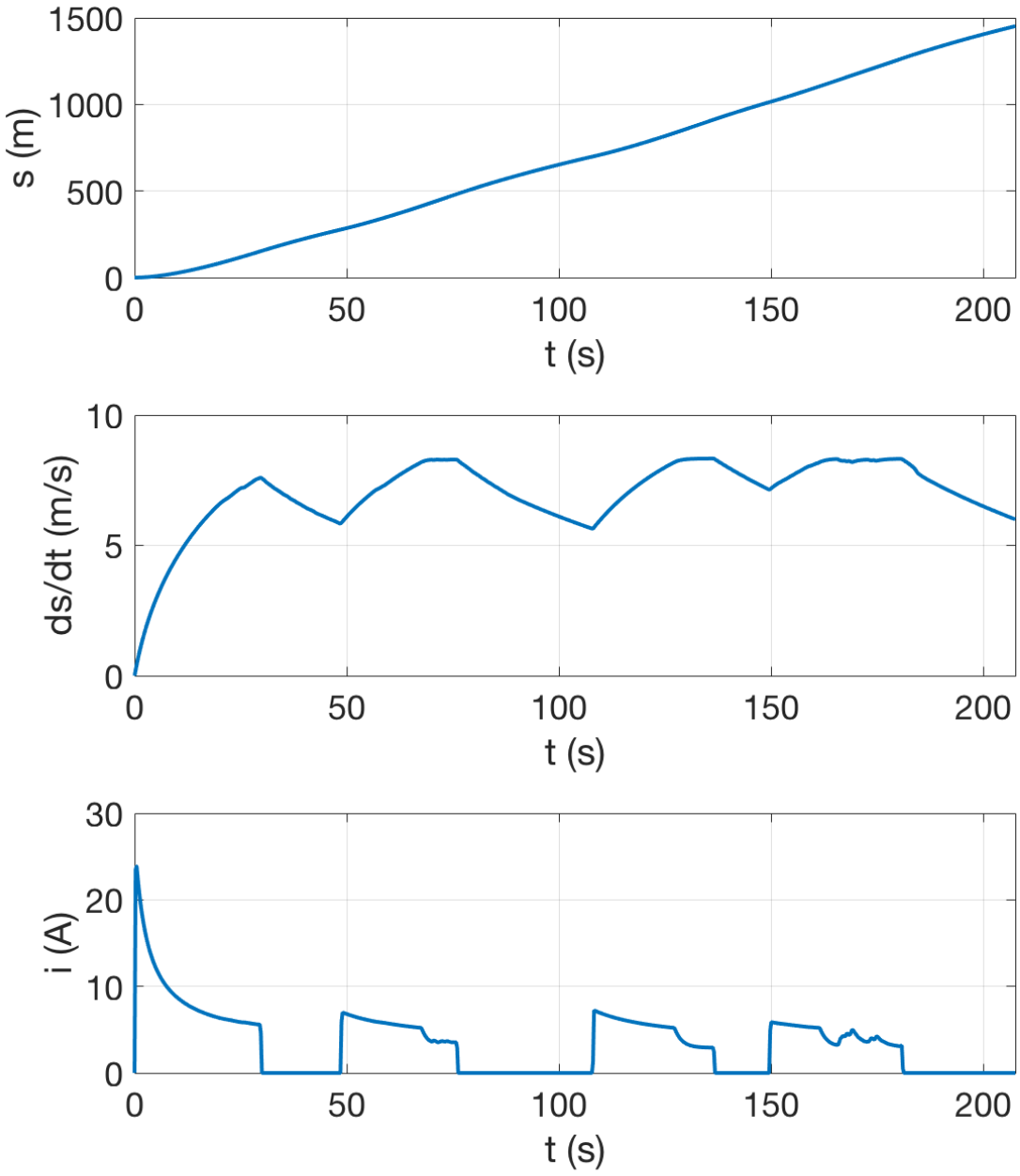

Figure 20: Simulation result for a four burn strategy on the 2019 track.

C.5 Five Burns

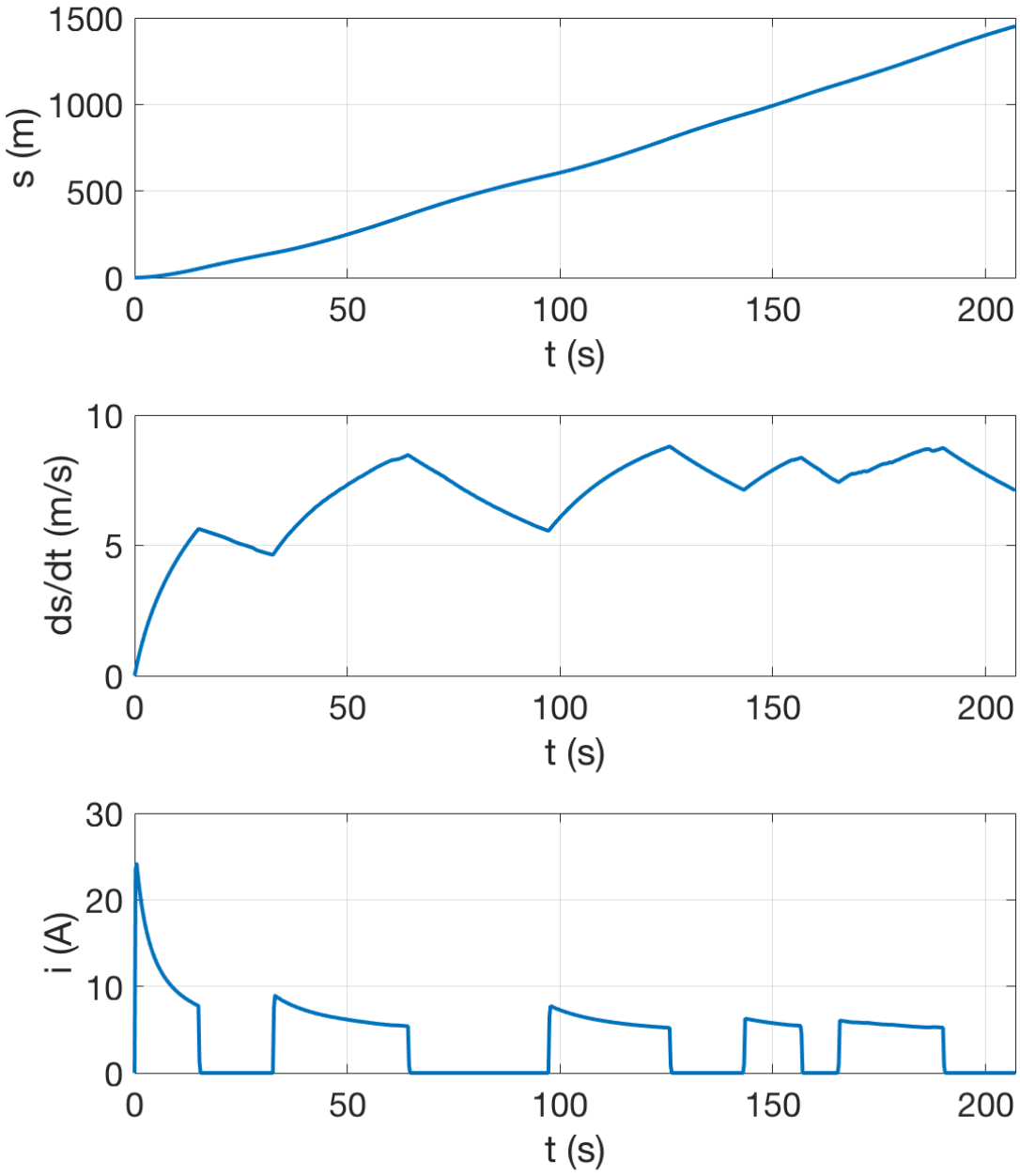

Figure 21: Simulation result for a five burn strategy on 2019 track.

C.6 Six Burns

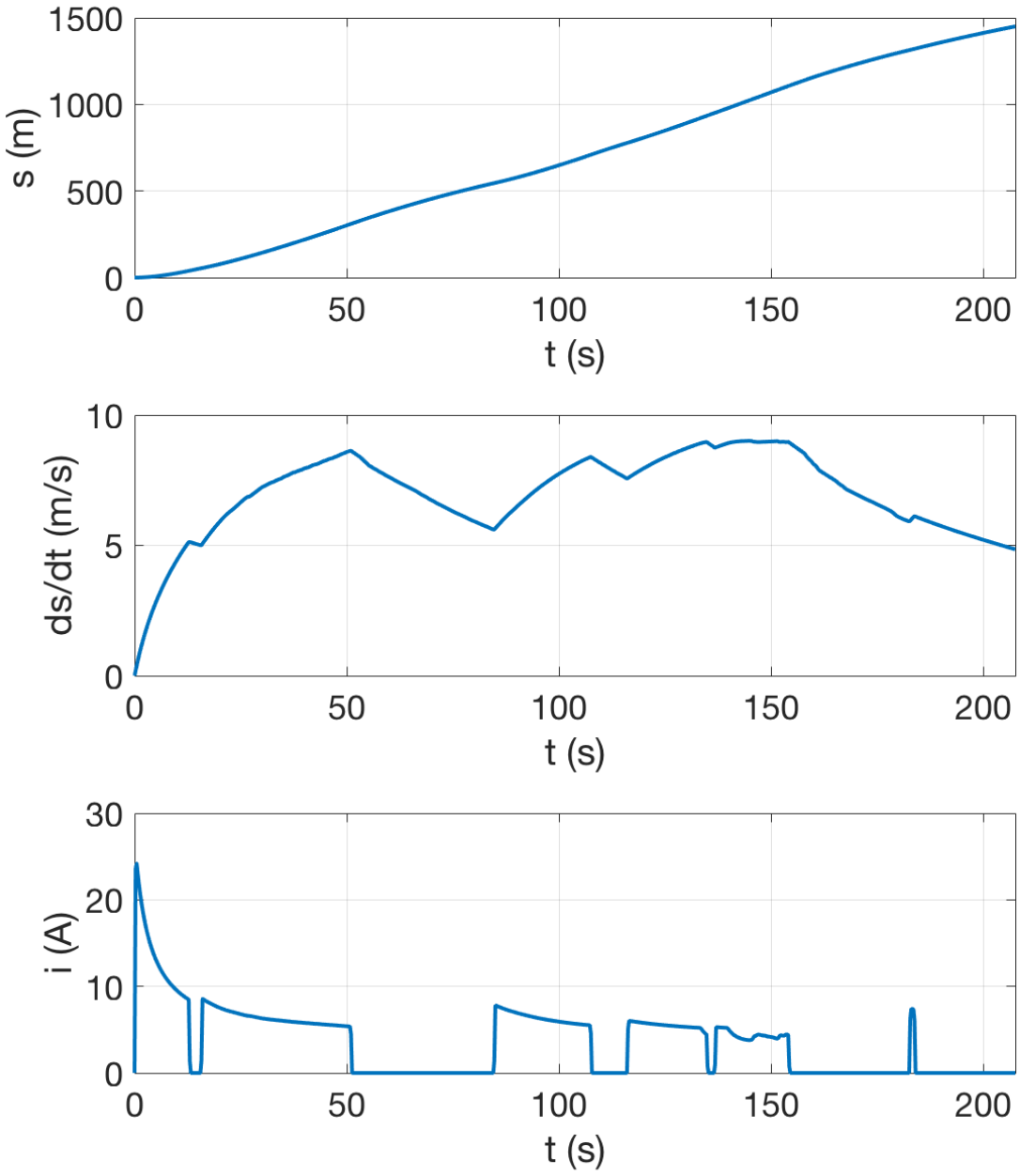

Figure 22: Simulation result for a six burn strategy on the 2019 track.