



Article

Simultaneous Optimisation of Vehicle Design and Control for Improving Vehicle Performance and Energy Efficiency Using an Open Source Minimum Lap Time Simulation Framework

Alberto Jiménez Elbal [†], Adrián Zarzuelo Conde ^{*,†} and Efstathios Siampis ^{*}

Advanced Vehicle Engineering Centre (AVEC), Cranfield University, College Road, Cranfield MK43 0AL, UK; ajelbal@gmail.com

^{*} Correspondence: adrian.zarzuelo.97@gmail.com (A.Z.C.); efstathios.siampis@cranfield.ac.uk (E.S.); Tel.: +44-(0)-1234-754654 (E.S.)

[†] These authors contributed equally to this work.

Abstract: This paper presents a comprehensive framework for optimising vehicle performance, integrating advanced simulation techniques with optimisation methodologies. The aim is to find the best racing line, as well as the optimal combination of parameters and control inputs to make a car as fast as possible around a given track, with a focus on energy deployment and recovery, active torque distribution and active aerodynamics. The problem known as the Minimum Lap Time Problem is solved using optimal control methods and direct collocation. The solution covers the modelling of the track, vehicle dynamics, active aerodynamics, and a comprehensive representation of the powertrain including motor, engine, transmission, and drivetrain components. This integrated simulator allows for the exploration of different vehicle configurations and track layouts, providing insights into optimising vehicle design and vehicle control simultaneously for improved performance and energy efficiency. Test results demonstrate the effect of active torque distribution on performance under various conditions, enhanced energy efficiency and performance through regenerative braking, and the added value of including parameter optimisation within the optimisation framework. Notably, the simulations revealed interesting behaviours similar to lift-and-coast strategies, depending on the importance of energy saving, thereby highlighting the effectiveness of the proposed control strategies. Also, results demonstrate the positive effect of active torque distribution on performance under various conditions, attributed to the higher utilization of available adherence. Furthermore, unlike a simpler single-track model, the optimal solution required fine control of the active aerodynamic systems, reflecting the complex interactions between different subsystems that the simulation can capture. Finally, the inclusion of parameter optimisation while considering all active systems, further improves performance and provides valuable insights into the impact of design choices.

Keywords: minimum lap time problem; trajectory optimisation; electric powertrain; active aerodynamics; active torque distribution; optimal control; simulation; CasADi; ipopt



Citation: Jiménez Elbal, A.; Zarzuelo Conde, A.; Siampis, E. Simultaneous Optimisation of Vehicle Design and Control for Improving Vehicle Performance and Energy Efficiency Using an Open Source Minimum Lap Time Simulation Framework. *World Electr. Veh. J.* **2024**, *15*, 366. <https://doi.org/10.3390/wevj15080366>

Academic Editor: Ayman EL-Refaie

Received: 30 June 2024

Revised: 31 July 2024

Accepted: 2 August 2024

Published: 13 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

This study focuses on developing a simulation framework to maximise a vehicle's speed on the track while studying the effects of dynamic controls such as active torque distribution, active aerodynamics, and electric energy recovery and deployment. This problem is commonly known as the Minimum Lap Time Problem (MLTP). Engineers have been trying to tackle the problem of minimum lap time for long and there is extensive literature about it. The first references date back to the 1950s [1], where the state and control trajectories were reconstructed from the maximum speed at the apex of the corners. Since then, many different techniques have been explored. A detailed overview of these methods can be found in [2,3].

To support the approach of optimising vehicle performance and energy efficiency, recent studies have demonstrated the efficacy of different control strategies for torque distribution and energy management in hybrid and electric vehicles [4–7].

The MLTP can be considered a path planning problem, where the goal is to find a trajectory—i.e., a succession of states of the dynamic system, in this case, the vehicle—that satisfies certain criteria, such as minimising the lap time while staying within the limits of the track. As highlighted in [2], one of the main constraints of the system comes from the saturation of the tyre forces. In this work, additional criteria are considered such as the utilisation of electric energy. In conclusion, the optimal racing line is determined for a given circuit by the physical properties of the system, and any change in the vehicle configuration results in a different optimal path, hence the optimal controls must be found as well. Note that both the racing line and the controls are part of the solution to the same offline optimisation problem, which is a different problem compared to online path tracking as highlighted in [8,9].

The evolution of computational technologies has expanded the capabilities to solve optimisation problems and, nowadays, practically every approach relies on simulation and numerical methods. A more recent work [10] presents a comprehensive review of the most relevant techniques that are currently being used, and these are classified into four types: quasi-steady-state simulations, optimal control, driver model-based, and evolutionary algorithm-based.

The limitations of the quasi-steady-state assumptions are demonstrated in [2] and a comparison with the optimal control approach is made in [11], showing a considerable difference in the solution for the Barcelona circuit. On the other hand, the approaches that use a driver model start by splitting the problem into two: trajectory *planning* and *tracking*; solutions found this way are sub-optimal because they depend on the driver model that is used [10]. In [2], this method is applied to find an initial estimate of the controls, which is then used to solve the problem using optimal control techniques. Finally, the optimal control approach is capable of solving a problem that takes into account the transient behaviour of complete dynamic models, simultaneously finding the optimal race line and controls.

Optimal control techniques are usually classified into direct and indirect methods. A third category would be dynamic programming, however, there are scarce applications found in the literature [10]. Once the MLTP is formulated as an OCP (optimal control problem), indirect methods aim to find the necessary conditions for optimality, whereas direct methods transform it into a discrete minimisation problem, also known as NLP (non-linear program). An in-depth general comparison between the direct and indirect formulation is presented in [12] while [10] compares both types of methods applied to the MLTP.

Both approaches have been used successfully to solve a variety of optimisation problems. The various methods that fall into either of these two categories have differences between them, and each has its advantages and drawbacks. However, the general caveats of indirect methods are precisely outlined in [12]: analytical expressions for the optimality conditions have to be derived, which is not an easy task for complex dynamic systems, and would be a major barrier when making modifications to the models e.g., by introducing new subsystems; accurate initial guesses are required for the adjoint variables that may not have an intuitive physical meaning; finally, including inequality constraints that are a function of the continuous states and inputs (known as path constraints) present some difficulties. In contrast, direct methods do not require initial guesses for the adjoint variables; instead, the state and control variables are directly adjusted by the solver when looking for the solution, therefore it is more intuitive to provide adequate initial guesses. Furthermore, path constraints can be easily included and the formulation of the problem can be extended to optimise parameters of the vehicle model like in [3,13]. Historically, the main obstacles found when applying direct methods were the solution of large-scale sparse NLPs resulting from the transcription of the OCP and the calculation of the first and second derivatives of

the equations of the system needed by the solver [14], but significant progress in those areas have been made in later years. For these advantages, direct methods are used in this work.

There are various techniques to implement the direct approach. Shooting methods rely on the integration of the dynamic equations. For example, single-shooting works by integrating the whole trajectory of the states by making an initial guess of the inputs but a major drawback is the sensitivity to the initial guesses [14], making it only suitable for problems with very simple controls or when a good initial guess is known [15]. Multiple-shooting methods reduce the integration horizon by dividing the problem into a finite number of intervals to mitigate this effect, nevertheless, all shooting methods suffer from some of the same problems as indirect methods that make the inclusion of path constraints a difficult task. Direct collocation methods do not suffer from this impediment [12]. These work by replacing the dynamic equations of the OCP by defect (or collocation) constraints in the NLP in a discrete number of points [14]. They are also more robust and can still produce good approximations when using fewer discrete points [16]. The collocation constraints are based on the idea that the change in the state must be equal to the integral of the dynamics. An alternative explanation of direct collocation is given in [15]. If the system dynamics and the inputs are approximated by polynomials, then the integral of the dynamics—i.e., the trajectory of the states—can be calculated explicitly using quadrature formulas, which only need the value of the states and the inputs in a discrete number of points. Therefore, another way of expressing the collocation constraints is by matching the derivatives of the system with the derivatives of the approximating polynomials in the collocation points. Both ideas are equivalent, as shown in [16].

Orthogonal collocation is a flavour of collocation that uses high-order orthogonal polynomials to approximate the trajectory. The discretisation is done in a particular set of selected points, not equally spaced. Orthogonal polynomials are known to have some desirable properties: interpolation formulas are well conditioned, especially when using a high number of interpolation points, as opposed to schemes using uniformly spaced points [17]; quadrature formulas evaluated in k points approximate the trajectories of the dynamics and inputs exactly if they are polynomials of order $2k$ [16]; and if the solution is sufficiently smooth, the convergence rate is exponential [18] (even if it is not smooth, the convergence is still faster than with Runge-Kutta methods [19]). The method to solve the OCP developed in this study makes use of orthogonal collocation, using a Lagrange basis representation for the approximating polynomials that benefits from barycentric interpolation to efficiently compute the quadrature formulas, the derivatives of the approximating polynomials, and the interpolation of the state's trajectory. The implementation takes advantage of the freely available NLP solver ipopt [20], and the open-source automatic differentiation framework Casadi [21], which has been a popular combination in recent works.

In summary, this project aims to demonstrate the suitability and applicability of trajectory optimisation techniques combined with algorithmic differentiation to solve the minimum lap time problem, finding the optimal combination of inputs, and the optimal value of either or both design and set-up parameters. The simultaneous optimisation of these parameters and the minimisation problem is a novel aspect of this work. Sections 2.2 and 2.3 introduce the modelling strategy for the track and vehicle; while Section 3 provides a detailed description of the OCP formulation and its transcription into a NLP. Finally, the developed software is used in Section 4 to present an optimisation study.

The intent is not to find an optimal configuration for any particular vehicle or track, but rather to explore and understand the use of these techniques with a focus on electric powertrains, vehicle controls, and the compromise between efficiency and performance. More specifically, the objectives of this piece of work include:

- Developing an appropriate vehicle dynamics simulation model, including non-linear tyres, wheel dynamics, longitudinal and lateral load transfer, aerodynamic forces, and a hybrid powertrain model that comprises an ICE, an electric motor and four clutch mechanisms used for active torque distribution.

- Creating a simulation environment for testing the developed vehicle model around a virtual track, aiming to have a modular architecture that has the flexibility to include other subsystems of the car.
- Finding the optimal combination of parameters and controls for minimising lap times on different tracks representative of all possible manoeuvres, studying the benefit of energy recovery, active torque distribution and active aerodynamics to strike the right balance between performance and efficiency.

2. Track, Vehicle and Powertrain Modelling

2.1. Materials and Methods

The following diagram (Figure 1) showcases the proposed simulation framework and how this is presented in this paper.

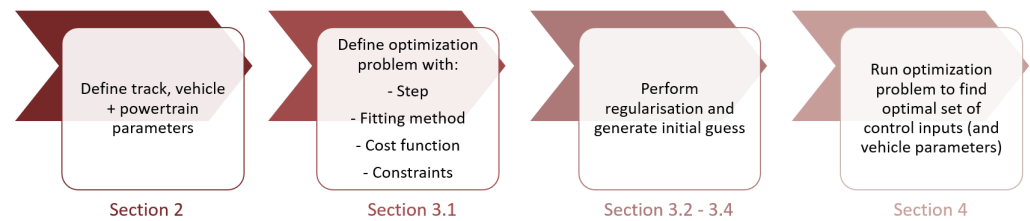


Figure 1. Logical framework diagram.

2.2. Track Model

The track model includes the track definition and its limits, as well as the vehicle positioning along the track (Figure 2). Amongst all the possible track definitions, a 2D map where the centre line is defined by the curvature (κ) and space (s) is chosen for this model. The width at each point (w) defines the track limits, within which the vehicle must stay at all times.

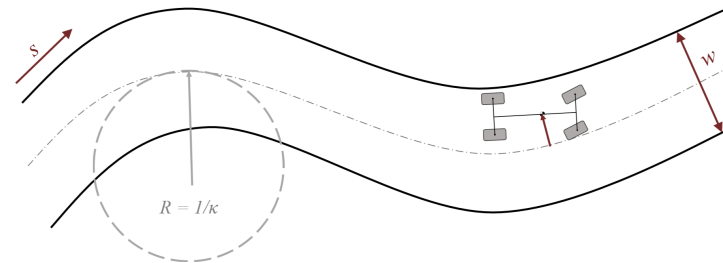


Figure 2. Coordinate system.

This definition of the track facilitates using the distance s as the independent variable instead of time, a widespread approach that is beneficial for several reasons. Firstly, the OCP is easier to discretise because time is unknown a priori [22]. Secondly, the curvature and width of the track are defined explicitly at every discretisation point. Lastly, there is one less state needed for the vehicle model, reducing the computational effort.

Similar to Equations (6)–(8) in [3], the vehicle's dynamic equations can be expressed as:

$$\frac{d}{dt}s = \frac{v_x \cos \epsilon - v_y \sin \epsilon}{1 - n\kappa} \quad (1)$$

$$\frac{d}{dt}n = v_x \sin \epsilon + v_y \cos \epsilon \quad (2)$$

$$\frac{d}{dt}\epsilon = \dot{\epsilon} - \kappa \frac{ds}{dt} \quad (3)$$

where:

- v_x : Vehicle's longitudinal speed;
- v_y : Vehicle's lateral speed;

r : Yaw rate;

n : Distance to the centre line;

ϵ : Angle of the vehicle respect to the tangent to the centre line.

A change of variable must be performed to express the dynamic equations as a function of the distance along the track, s , instead of time, t . If the dynamic equations are

$$\frac{d}{dt} \mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (4)$$

Being \mathbf{x} the vector of states and \mathbf{u} the vector of inputs, the change of variable is performed by multiplying each side of the previous equation by the inverse of (1), named as S_f :

$$\frac{d}{ds} \mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \cdot \frac{dt}{ds} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \cdot S_f \quad (5)$$

2.3. Vehicle Model

The vehicle is modelled using a double-track model with lateral and longitudinal load transfer, non-linear tyres and active aerodynamic forces (see Figure 3). Using the same nomenclature as in the previous section, the equations that describe the model's dynamics along with the Equations (2) and (3) are derived from two force balances along and torque balance:

$$\frac{d}{ds} v_x = (F_{xrl} + F_{xrr} + (F_{xfl} + F_{xfr}) \cdot \cos(\delta) - (F_{yfl} + F_{yfr}) \cdot \sin(\delta) - F_{drag} + M \cdot v_y \cdot r) \cdot S_f / M \quad (6)$$

$$\frac{d}{ds} v_y = (F_{yrl} + F_{yrr} + (F_{yfl} + F_{yfr}) \cdot \cos(\delta) + (F_{xfl} + F_{xfr}) \cdot \sin(\delta) - M \cdot v_x \cdot r) \cdot S_f / M \quad (7)$$

$$\begin{aligned} \frac{d}{ds} r = & (-F_{yrl} \cdot l_r - F_{yrr} \cdot l_r - F_{xrl} \cdot \frac{t}{2} + F_{xrr} \cdot \frac{t}{2} + (F_{yfl} \cdot \cos(\delta) + F_{xfl} \cdot \sin(\delta)) \cdot l_f + \\ & + (F_{yfr} \cdot \cos(\delta) + F_{xfr} \cdot \sin(\delta)) \cdot l_f - (F_{xfl} \cdot \cos(\delta) - F_{yfl} \cdot \sin(\delta)) \cdot \frac{t}{2} + \\ & + (F_{xfr} \cdot \cos(\delta) - F_{yfr} \cdot \sin(\delta)) \cdot \frac{t}{2}) \cdot S_f / I_z; \end{aligned} \quad (8)$$

where:

$F_{k,i,j}$: Tyre forces in the tyre local frame (k : longitudinal(x)/lateral(y), i, j : tyre location);

F_{drag} : Aerodynamic drag force;

δ : Steering angle;

l_f : Longitudinal position of the centre of gravity respect to the front axle;

l_r : Longitudinal position of the centre of gravity respect to the rear axle;

t : Vehicle's track;

M : Vehicle's mass;

I_z : Vehicle's moment of inertia around it's vertical axis.

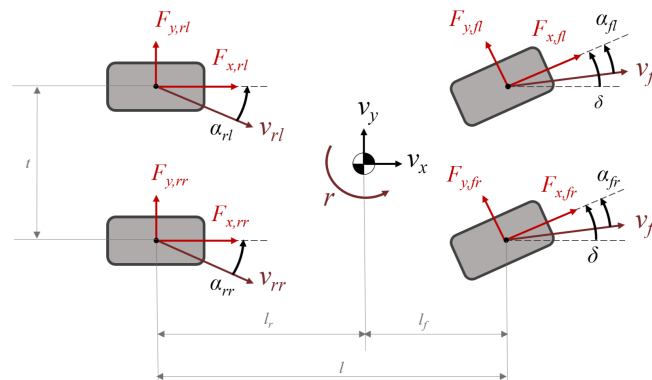


Figure 3. Double track model.

The wheel's dynamics are also included, which means that four extra states are used to consider their rotation. The equation is derived from a torque balance along the wheels' rotational axis:

$$\frac{d}{ds} \Omega_{i,j} = (T_{i,j} - F_{x,i,j} \cdot R_{w,i}) \cdot S_f / J_w \quad (9)$$

where:

- $\Omega_{i,j}$: Angular velocity of the wheel;
- $T_{i,j}$: Torque applied at the wheel;
- $F_{x,i,j}$: Longitudinal force produced by the tyre;
- $R_{w,i}$: Wheel radius;
- J_w : Wheel rotational inertia.

2.3.1. Tyres

The forces acting on the tyres have three components, vertical, lateral and longitudinal. To calculate the lateral and longitudinal forces, the vertical force acting on each tyre needs to be determined first. This vertical force (F_z) is calculated using the static weight distribution, the aerodynamic lift force calculated in Section 2.3.3 (F_{lift}), and the lateral and longitudinal weight transfers calculated in Section 2.3.2 (Γ_y and Γ_x respectively):

$$F_{z,ij} = M \cdot g \cdot \frac{l_i}{l} \cdot \frac{1}{2} + \frac{F_{lift}}{4} \pm \frac{\Gamma_x}{2} \pm \frac{\Gamma_y}{2} \quad (10)$$

- M : Vehicle mass;
- g : Acceleration of gravity;
- l : Wheelbase;
- l_i : Distance from the centre of gravity to each axle;
- F_{lift} : Aerodynamic vertical force;
- Γ_y and Γ_x : Lateral and longitudinal weight transfers.

The remaining tyre forces (lateral and longitudinal) are calculated using Pacejka's magic formula [23]:

$$F_{x,ij} = D_{x,ij} \cdot \sin(C_x \cdot \text{atan}(B_x \cdot s_{x,ij} - E_x \cdot (B_x \cdot s_{x,ij} - \text{atan}(B_x \cdot s_{x,ij})))) - F_{roll,ij} \quad (11)$$

$$F_{y,ij} = D_{y,ij} \cdot \sin(C_y \cdot \text{atan}(B_y \cdot \alpha_{ij} - E_y \cdot (B_y \cdot \alpha_{ij} - \text{atan}(B_y \cdot \alpha_{ij})))) \quad (12)$$

The coefficients C_k , B_k and E_k are constant, while $D_{k,ij}$ depends on F_z and the individual wheel's coefficient of friction μ_{ij} , which depends also on F_z to account for the load sensitivity.

$$D_{k,ij} = \mu_{ij} \cdot F_{z,ij} \quad (13)$$

$$\mu_{ij} = \mu_0 + P_{D2} \cdot \frac{F_{z,ij} - F_{z0}}{F_{z0}} \quad (14)$$

where μ_0 is the coefficient of friction at the nominal vertical load F_{z0} , and P_{D2} is a negative coefficient that dictate the tyre load sensitivity.

The combined behaviour is included in the model as a constraint with a friction circle (assuming the same μ in both longitudinal and lateral directions), where the total available adherence on each tyre must be greater or equal to the used adherence in that tyre:

$$\mu_{ij}^2 - \frac{F_{x,ij}^2 + F_{y,ij}^2}{F_{z,ij}^2} \geq 0 \quad (15)$$

The slip ratios and slip angles can be calculated as a function of the states.
Slip angles:

$$\alpha_{fj} = \delta - \operatorname{atan}\left(\frac{r \cdot l_f + v_y}{v_x \pm r \cdot t/2}\right) \quad (16)$$

$$\alpha_{rj} = \operatorname{atan}\left(\frac{r \cdot l_f - v_y}{v_x \pm r \cdot t/2}\right) \quad (17)$$

$$v_{k,ij} = \cos(\alpha_{ij}) \cdot \sqrt{(v_y \pm r \cdot l_f)^2 + (v_x \pm r \cdot t/2)^2} \quad (18)$$

The last term remaining to be explained is the rolling resistance (F_{roll}). For simplicity reasons it is assumed to be constant by neglecting the load transfer:

$$F_{roll,i} = f \cdot F_{z0} = f \cdot M \cdot \frac{l_i}{l} \quad (19)$$

where f is the rolling resistance coefficient.

2.3.2. Load Transfer

The load transfer is calculated in a quasi-steady state form, as the suspension and other chassis dynamics are not modelled. To avoid creating an algebraic loop, new decision variables are introduced for the load transfer using the equations below as path constraints:

$$\Gamma_x - ((F_{x,fl} + F_{x,fr}) \cdot \cos(\delta) - (F_{y,fl} + F_{y,fr}) \cdot \sin(\delta) + F_{x,rl} + F_{x,rr} + F_{drag}) \cdot \frac{h_{CG}}{l} = 0 \quad (20)$$

$$\Gamma_y - ((F_{y,fl} + F_{y,fr}) \cdot \cos(\delta) + (F_{x,fl} + F_{x,fr}) \cdot \sin(\delta) + F_{y,rl} + F_{y,rr}) \cdot \frac{h_{CG}}{t} = 0 \quad (21)$$

where h_{CG} is the centre of gravity height.

2.3.3. Aerodynamics

The aerodynamic forces are introduced as quadratic functions that depend on the longitudinal speed and are applied in the centre of gravity. Active aerodynamics are introduced with variable coefficients that are computed by interpolating linearly between a minimum and maximum value, with the control variable λ taking a value between 0 and 1. Note that the value of both coefficients is coupled by this control variable that allows the aerodynamic package to shift between minimum drag and maximum downforce configurations. In the equations, κ refers to either drag or lift:

$$F_{drag} = \frac{1}{2} \cdot \rho \cdot C_d \cdot A \cdot v_x^2 \quad (22)$$

$$F_{lift} = \frac{1}{2} \cdot \rho \cdot C_l \cdot A \cdot v_x^2 \quad (23)$$

$$C_\kappa = C_{\kappa,min} + \lambda \cdot (C_{\kappa,max} - C_{\kappa,min}) \quad (24)$$

where:

ρ : Air density;
 C_d and C_l : Drag and lift coefficients;
 A : Vehicle frontal area.

2.3.4. Brakes

The brake torque is a control variable. Also the electric motor and the ICE described later have braking capabilities. The braking torque T_{brake} is the average of all four wheels, so it can be expressed:

$$T_{fj} = 2 \cdot T_{brake} \cdot brk_B \quad (25)$$

$$T_{rj} = 2 \cdot T_{brake} \cdot (1 - brk_B) \quad (26)$$

where T_{brake} is the control braking torque and brk_B the front to rear brake balance.

2.3.5. Powertrain

There are many approaches to simulating hybrid vehicle powertrains, however, most involve complex dynamic models that are out of the scope of this work. The aim of these papers is usually to understand or analyse internal aspects of the powertrain, not so much its influence on the performance of the whole vehicle. Refs. [24–26] are some examples of papers that model hybrid powertrains, but their approaches are not practical for this application. A specific model representative of a high performance sports car has been created for this study for that reason. More concretely, it consists of an electric hybrid powertrain with full active torque distribution and a nine-speed gearbox. The active torque distribution is achieved with four slip clutches located at each wheel. It is powered by an electric motor and an ICE (Internal Combustion Engine); both run through the gearbox and other transmission components. The layout of these components can be seen in Figure 4.

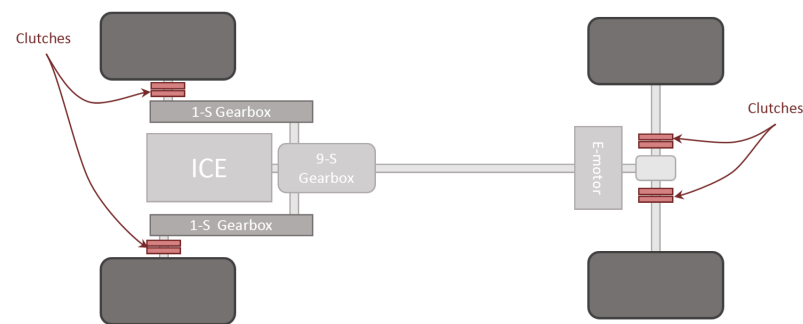


Figure 4. Powertrain schematic.

There are also gear ratios between the ICE and the motor, the gearbox and the rear wheels, and between the motor and the front wheels.

The ICE is modelled using a torque-speed map approximated by three straight lines and included as constraints limiting the maximum ICE torque. A reference for the specifications of a similar engine can be found in [27]. The ICE braking torque is included in the simulation as a negative lower bound for the torque for simplicity, but a braking torque curve could be implemented following the same strategy as for the driving torque.

$$T_{ICE} - (a_i \cdot \Omega_{ICE} + b_i) \leq 0 \quad (27)$$

Being:

T_{ICE} : ICE driving torque;

Ω_{ICE} : Engine speed;

a_i and b_i : Coefficients for each straight segment.

The electric motor is modelled with a torque and a power limits. The maximum braking and driving torques are included directly as input limits (bounds for the decision variables), and the maximum power is added as an extra constraint that limits the regenerative torque as well. To cover both positive and negative torque demands and make the power limit symmetric about the speed axis (i.e. the x-axis), we introduce the following constraint:

$$-P_{max} \leq T_{motor} \cdot \Omega_{motor} \leq P_{max} \quad (28)$$

where:

T_{motor} : Motor's driving torque;

Ω_{motor} : Motor speed;

P_{max} : Maximum power.

There are four control variables that have not been explained yet. Three of them control the actuation of the clutches; these are combined to get the final values of each individual wheel's clutch (which goes from zero to one). The sum of all wheel clutch values must always add to one, so the power coming out of the engine and motor is not lost. The individual clutches control the torque distribution from left to right in each axle, and the torque distribution from front to rear (Cl_f , Cl_r and FR_{dist}). The relationship between these and the three control variables is described by:

$$Cl_{fr} = (1 - Cl_f) \cdot FR_{dist} \quad (29)$$

$$Cl_{fl} = Cl_f \cdot FR_{dist} \quad (30)$$

$$Cl_{rr} = (1 - Cl_r) \cdot (1 - FR_{dist}) \quad (31)$$

$$Cl_{rl} = Cl_r \cdot (1 - FR_{dist}) \quad (32)$$

The remaining input variable controls the gearbox. It represents directly the gear number. The inclusion of a discrete variable in the formulation of the problem as an NLP presents some difficulties. Since it is not the focus of this study, in this case the gear number can take any real value between one and nine, and it is used to evaluate a cubic function fitted to the gear ratios. Knowing all the powertrain control variables, the torque distribution can be defined. The torque at each wheel can be expressed as follows:

$$T_{fl} = (T_{motor} + T_{ICE} \cdot gear_{rat} \cdot gear_{rat-mid}) \cdot gear_{rat-f} \cdot Cl_{fl} + 2 \cdot T_{brake} \cdot brk_B \quad (33)$$

$$T_{fr} = (T_{motor} + T_{ICE} \cdot gear_{rat} \cdot gear_{rat-mid}) \cdot gear_{rat-f} \cdot Cl_{fr} + 2 \cdot T_{brake} \cdot brk_B \quad (34)$$

$$T_{rl} = (T_{motor} / gear_{rat-mid} + T_{ICE} \cdot gear_{rat}) \cdot gear_{rat-r} \cdot Cl_{rl} + 2 \cdot T_{brake} \cdot (1 - brk_B) \quad (35)$$

$$T_{rr} = (T_{motor} / gear_{rat-mid} + T_{ICE} \cdot gear_{rat}) \cdot gear_{rat-r} \cdot Cl_{rr} + 2 \cdot T_{brake} \cdot (1 - brk_B) \quad (36)$$

where:

$gear_{rat}$: 9-S gearbox gear ratio;

$gear_{rat-mid}$: Gear ratio between the 9-S gearbox and the motor;

$gear_{rat-f}$: Gear ratio between the motor and the front wheels;

$gear_{rat-r}$: Gear ratio between the 9-S gearbox and the rear wheels.

The ICE and motor speeds are calculated:

$$\Omega_{ICE} = (\Omega_{fl} + \Omega_{fr}) / 4 \cdot gear_{rat-f} \cdot gear_{rat-mid} \cdot gear_{rat} + (\Omega_{rl} + \Omega_{rr}) / 4 \cdot gear_{rat-r} \cdot gear_{rat} \quad (37)$$

$$\Omega_{motor} = (\Omega_{fl} + \Omega_{fr}) / 4 \cdot gear_{rat-f} + (\Omega_{rl} + \Omega_{rr}) / 4 \cdot gear_{rat-r} / gear_{rat-mid} \quad (38)$$

The following images have been included in order to illustrate different powertrain working situations. In all three images the vehicle is turning left. In Figure 5a the vehicle is braking while starting to turn, so the inner wheels are producing a greater braking torque. The image Figure 5b is a typical situation accelerating in a corner, where the front wheels require less driving torque as they are using part of their adherence generating lateral acceleration, so part of the motor torque is being transferred to the rear wheels. Finally, Figure 5c represents a corner exit, where the front wheels can use more torque, so they are using part of the one from the ICE.

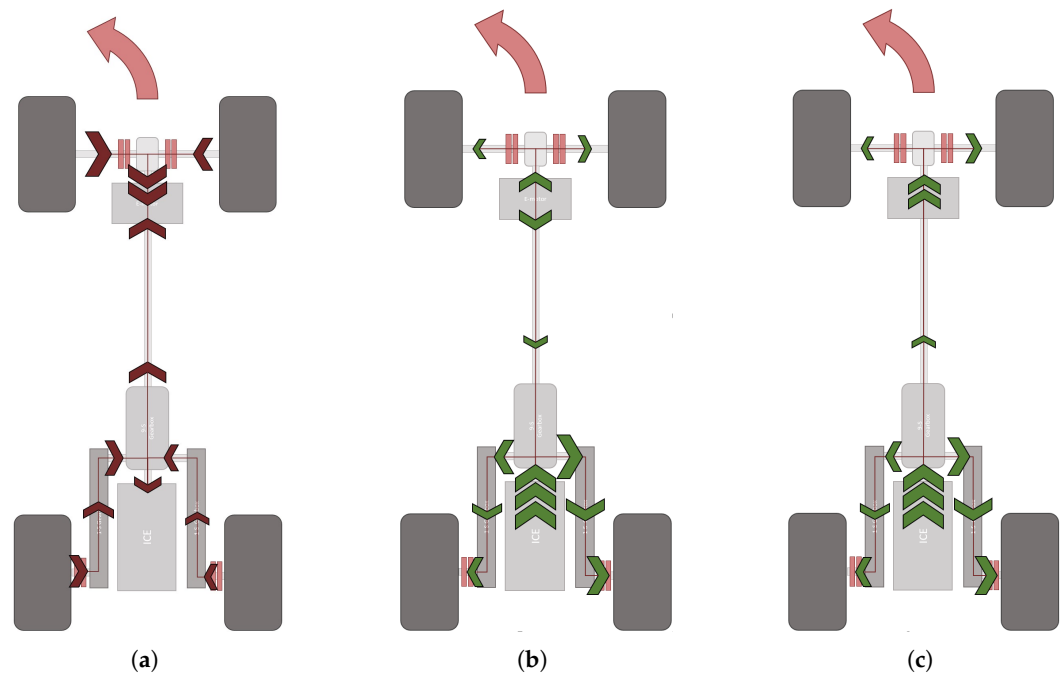


Figure 5. (a) Braking; (b) Driving situation 1; (c) Driving situation 2.

3. Trajectory Optimisation

As introduced before, the MLTP problem can be described as an OCP, that can be transformed into a discrete NLP using direct orthogonal collocation. This process is known as transcription. The final implementation is described in this section. A general formulation of an OCP can be found in [28]. A simpler formulation is presented in [15], focused on a single-phase continuous-time problem, which is a problem where the dynamics of the system do not change over time. In essence, the OCP aims to minimise a cost function while satisfying a set of constraints; it is special in that the optimisation variables include continuous functions.

In the case of the MLTP problem, when using time as the independent variable, the integrand of the Lagrange term in the objective function as presented in [15] would be equal to 1. In the present work, time is replaced by distance as the independent variable, hence the integral objective function is defined by

$$J(w) = \int_0^{t_f} 1 \cdot dt = \int_0^{s_f} S_f(w) \cdot ds \quad (39)$$

where t_f and s_f are the final time and distance, and w contains the optimisation variables. Typically the optimisation variables are the states x , the inputs u , and a finite number of static parameters p . Here, an additional continuous variable is included, y , that accounts for auxiliary continuous variables that are neither states nor inputs, such as the load transfer. In other related studies, the load transfer was included by overloading the input vector. This new formulation introduces more flexibility in the treatment of the corresponding decision variables of the NLP, for example, by approximating y with higher order polynomials than the inputs.

The constraints of the OCP can be expressed as

$$\frac{d}{ds}x = f(w) \quad (40)$$

$$h(w) \leq 0 \quad (41)$$

$$b(w_0, w_f) \leq 0 \quad (42)$$

The system dynamics are enforced by (40); f includes (2), (3), (6)–(9). h represents the path constraints, which can be equality or inequality constraints, including (15), (20), (21) and 5 other equations to avoid brake throttle overlapping and to define the motor and ICE's torque curves. Boundary constraints are defined by b and can be used to define the initial and final states, for example, specifying an initial longitudinal speed, or the orientation of the car at the end of the track.

Path constraints that are a function of only one optimisation variable are applied by setting appropriate bounds to the corresponding decision variables of the NLP. For example, the normal distance of the vehicle to the centre line, which is a state of the model, must be always lower than half of the width of the track. Therefore, with Λ accounting for the vehicle width, the state bounds must be such that:

$$\frac{w}{2} - \Lambda \geq n \geq -\frac{w}{2} + \Lambda \quad (43)$$

3.1. Transcription

The process of transforming the OCP into a discrete NLP is known as transcription (that is why direct collocation is also called direct transcription). As introduced before, the method used here applies direct orthogonal local collocation with barycentric Lagrange interpolation [17]. Good references for this method can be found in [15,16,22].

Firstly, the total length of the track (the independent variable) is divided in N finite elements (hence $N + 1$ grid points, namely s_k) of arbitrary length ds_k . The states are approximated in each interval by a Legendre polynomial of degree d , for which the collocation points $s_{k,j}$ are chosen as the roots of the Legendre polynomials. These are typically defined in the interval $[-1, 1]$, but they can be found in any other interval simply by uniformly scaling the points to the new interval. This choice of collocation points provides the most accurate approximation of the corresponding quadrature formula, along with the other advantages discussed previously.

The states at each collocation point, $X_{k,j}$ - referred to as *helper states*—are used for the formulation of the collocation constraints by evaluating the system dynamics at those points. The derivative of the polynomial evaluated at the collocation points is designated as $\dot{L}_k(s_{k,j})$, where:

$$\begin{aligned} k &= \{1, \dots, N + 1\} \\ j &= \{1, \dots, d\} \end{aligned}$$

Another benefit of using orthogonal polynomials with barycentric Lagrange interpolation is that the differentiation, integration, and interpolation of the polynomials can be computed as a linear combination of the value of the polynomials at the collocation points, that is, a linear combination of the decision variables of the NLP. For instance, let's define $L_k = [l_k(s_{k,1}), l_k(s_{k,2}), \dots, l_k(s_{k,d})]$ being $l_k(s)$ the Lagrange polynomial of order d in the interval $[s_k, s_{k+1}]$, and $s_{k,j}$, the collocation points. Again, note that $l_k(s_{k,j}) = X_{k,j}$. Then, the derivatives at the collocation points can be found as

$$\dot{L}_k = \frac{C}{ds_k} \cdot L_k \quad (44)$$

The same can be done to calculate the contribution of each finite element to the objective function, being $Sf_k = [S_f(s_{k,1}), S_f(s_{k,2}), \dots, S_f(s_{k,d})]$; and to extrapolate the state at the end of the interval. The matrices C , B , and D can be calculated for a 'standardised' interval like $[0, 1]$ and its shape and value would only depend on the degree chosen for the polynomial.

$$J_k = B \cdot Sf_k \cdot ds_k \quad (45)$$

$$X_{end} = D \cdot [X_k, X_{k,1}, \dots, X_{k,d}] \quad (46)$$

The collocation constraints are constructed by matching the derivatives of the polynomials with the actual dynamics of the system at the collocation points, and by matching the extrapolated final state of each finite element with the first state of the consecutive one.

The inputs are usually selected as piece-wise constant functions, however, in this case, linear functions are used. This can be achieved by defining the value of the inputs at the grid points instead of the intervals (therefore the number of decision variables is only increased by the number of inputs). This implementation often results in faster solution times; the path constraints can be constructed in a more straightforward way since they can now be defined for every grid point; and it provides an easy way of calculating the derivatives in each interval, to apply the constraints over the rate of inputs. The same idea is applied to discretise the auxiliary variables y .

Finally, the NLP formulation results as follows:

minimise

$$J = \sum_{k=1}^{k=N} \mathbf{B} \cdot \mathbf{S} \mathbf{f}_k \cdot ds_k \quad (47)$$

subject to

$$\frac{d}{ds} l_k(s_{k,j}) - f(X_{k,j}, U_k, Y_k, p, \kappa_k) = 0 \quad \forall k \in [0, N] \quad \text{and} \quad j \in [1, d] \quad (48)$$

$$X_{end} - X_k = 0 \quad \forall k \in [1, N+1] \quad (49)$$

$$\mathbf{H}_{lb} \leq \mathbf{h}(X_k, U_k, Y_k, p, \kappa) \leq \mathbf{H}_{ub} \quad \forall k \in [0, N] \quad (50)$$

$$\mathbf{B}_{lb,k} \leq \mathbf{h}(X_k, U_k, Y_k, p, \kappa) \leq \mathbf{B}_{ub,k} \quad k = 0, k = N+1 \quad (51)$$

where the objective function (39) is transformed into (47), the dynamic constraints of the system (40) are transformed into the collocation constraints (48) and (49), the path constraints (41) become (50), and the boundary constraints (42) appear as (51). Path (and bound) constraints can be equality constraints simply by making $H_{lb} = H_{ub}$, however, equality constraints are eased in the implementation by allowing a small slack. This has proven to result in shorter simulation times. The decision variables become:

states	\mathbf{X}_k
helper states	\mathbf{X}_{kj}
inputs	\mathbf{U}_k
auxiliary variables	\mathbf{Y}_k
parameters	p

All of the decision variables are constrained by constant bounds. This is the procedure used to limit the minimum and maximum value of the inputs as well as that of some states like the maximum speed according to the maximum power and drive torque, or the maximum distance to the centre line to ensure that the vehicle stays on the track, as explained before. An additional set of constraints is included in the NLP to limit the maximum rate of change of the inputs. The derivative of the inputs is calculated as

$$\Delta U_k = \frac{U_{k+1} - U_k}{ds_k} \quad (52)$$

3.2. Scaling

The scaling of the decision variables can have a major impact on the solver's performance. This is a well-known limitation of this type of solution algorithm [3]. As suggested in [14], there is no clear answer to what is the best scaling procedure. Some studies scale the variables by non-dimensionalising using characteristic magnitudes of the problem such as the acceleration of gravity, the mass, or the length of the vehicle. However, in this work,

all the decision variables are scaled so that their value remains in the interval $[-1, 1]$, which has been observed to provide the fastest solution times.

For the variables for which the maximum value is known, the scaling is achieved by dividing over that value, for example, the decision variables corresponding to the drive torque would be divided by the maximum drive torque, which is a parameter of the car model. The objective function has not been scaled since the performance of the solver was reduced, although it has been found beneficial in other works. The path constraints are also scaled so that the value of h remains within the same order of magnitude as the decision variables.

3.3. Regularisation

One problem that may arise when solving an OCP is the existence of a family of solutions, that is, multiple sets of values of the decision variables that produce the same (or sufficiently close) value of the objective function. Regularisation is a method to give preference to certain solutions by introducing penalisation terms, which are usually scaled by a regularisation factor. If these factors are chosen sufficiently small, the value of the objective function would remain very close to the original. The most simple regularisation term is the square of the inputs. This aims to reduce the control effort, which is a common problem in disciplines like robotics. A slightly more sophisticated approach that penalises non-smooth solutions is achieved by including the derivative of the inputs. Furthermore, targeting solutions with oscillations, the second derivative is also included. Regularisation is applied to the inputs and the auxiliary variables Y_k .

A similar approach can be used to balance the weight of different targets in the objective function, like time and energy usage. The term to reflect the deployed electric energy in the objective function consists of the sum of the electric power in each grid point. It is worth noting that this term is signed, so it also favours the usage of regenerative braking to charge the battery.

The objective function J results as follows:

$$J = \sum_{k=1}^N \underbrace{[\mathbf{B} \cdot \mathbf{Sf}_k \cdot ds_k]}_{\text{time term}} + \underbrace{r_u \cdot U_k^2 + r_{du} \cdot \Delta U_k^2 + r_{du2} \cdot \Delta^2 U_k^2 + r_{dy} \cdot \Delta Y_k^2 + r_{dy2} \cdot \Delta^2 Y_k^2}_{\text{inputs and auxiliary variables smoothening term}} + \underbrace{r_e \cdot \Omega_{motor} \cdot T_{motor}}_{\text{electric energy term}} \quad (53)$$

3.4. Initial Guess

An initial guess is needed to solve the NLP. This initial guess is one of the most important aspects when setting up a trajectory optimisation problem. As Kelly [29] states “initialisation is more of an art than a science”. While some problems can be initialised with constant inputs and states, others may require a bit more sophisticated initialisation strategies. Solving a simpler model is one such strategy, as it provides results that can be close enough to the final optimal solution.

That is the approach used here. A single track model (Figure 6) with less control variables has been used for this purpose, and its solutions are used to create the initial guess for the double track model previously described. This is a rigid body model in 2D space with 3 degrees of freedom. It includes non-linear tyre forces, wheel dynamics, longitudinal load transfer and aerodynamic forces (lift and drag with constant coefficients). The powertrain is not included, so the control variable is the wheel torque with a constant maximum limit.

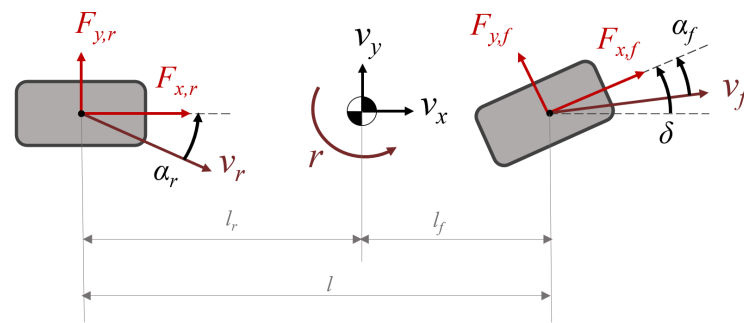


Figure 6. Single track model used for initialisation.

The four figures below compare the inputs of the initialisation solution with those of the final simulation. In Figure 7c the driving torque represented as the solution is the sum of the ICE and the motor torque.

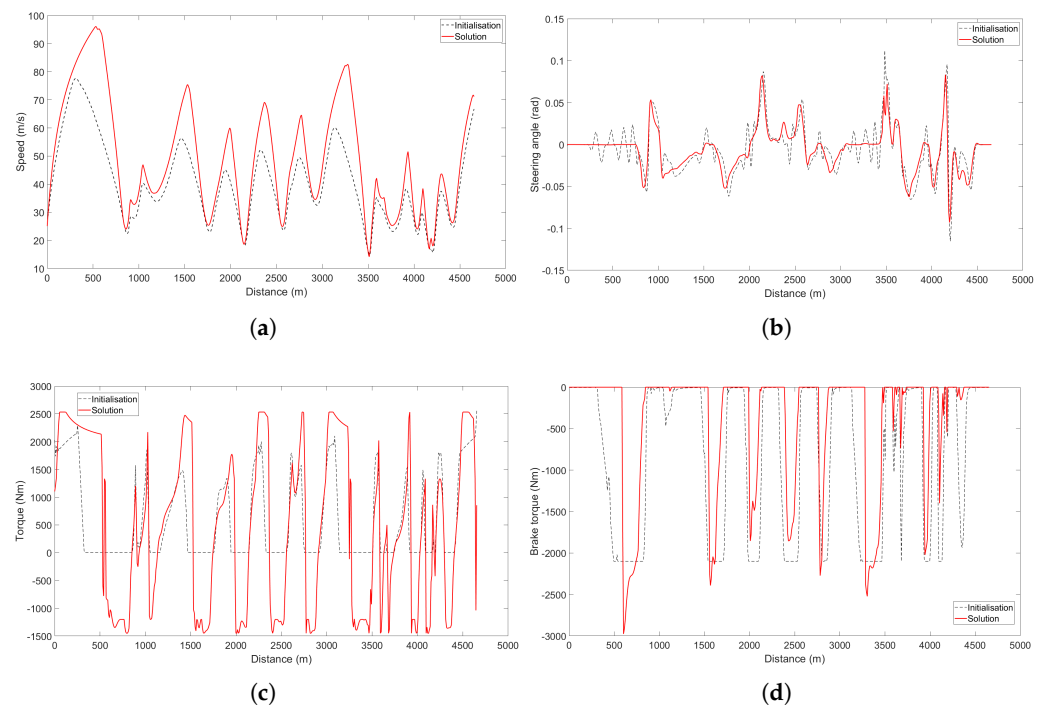


Figure 7. Input initial guesses and final simulation comparisons. (a) Vehicle speed; (b) Steering angle; (c) Driving torque; (d) Brake torque.

4. Results

This section presents the results of the optimization study conducted using the developed software on a DELL Precision 5680 laptop with a 13th Gen Intel(R) Core(TM) i7-13800H processor and 32 GB of RAM utilizing CasADi 3.6.5 version. The study shows lap times across various tracks by optimizing parameters such as energy recovery, active torque distribution, and active aerodynamics. Finally it shows the value that performing parameter optimisation within the laptime simulator brings to the vehicle designers.

4.1. Active Torque Distribution (ATD)

The ability to vary the torque at each wheel as desired is referred to as active torque distribution. This is achieved with the four clutches, allowing the vehicle to maximise the grip at each wheel. Without it, the wheel with less grip is limiting the torque that can be delivered without spinning the tyres. It is worth noting that the torque can not be varied in such way that some wheels produce positive torque while the others are producing

negative torque. This limitation is due to the powetrain scheme selected and the reason why the term torque vectoring is not being used.

In this section the same model with and without active torque distribution will be compared to see the performance improvement under different conditions. To perform this comparison, one of the parameters that will be used is the available μ . This is the difference between the coefficient of adherence used in each wheel, and the coefficient of adherence that is available based on the vertical load, evaluated by the Equation (14).

4.1.1. Straight Line

Under this situation, the ATD only changes the distribution from front to rear. The available adherence is near zero until the vehicle powertrain reaches the maximum torque, so the available μ starts increasing. This occurs at 47 m/s (170 kph), meaning that it has the ability to spin the wheels up until this speed.

With a 50/50 front to rear driving torque distribution, the ATD provides an improvement in performance, as the front tyres are maxing their adherence when the front tyres still have a μ of around 0.5 remaining. But with a better fixed distribution (in this case 36/64. Just obtained by iterating with different simulations), the improvement in performance is not that significant. The comparison is shown in Figure 8 and Table 1.

This is why ATD is not fully taken advantage of in a straight line. It is significantly more beneficial in turning conditions.

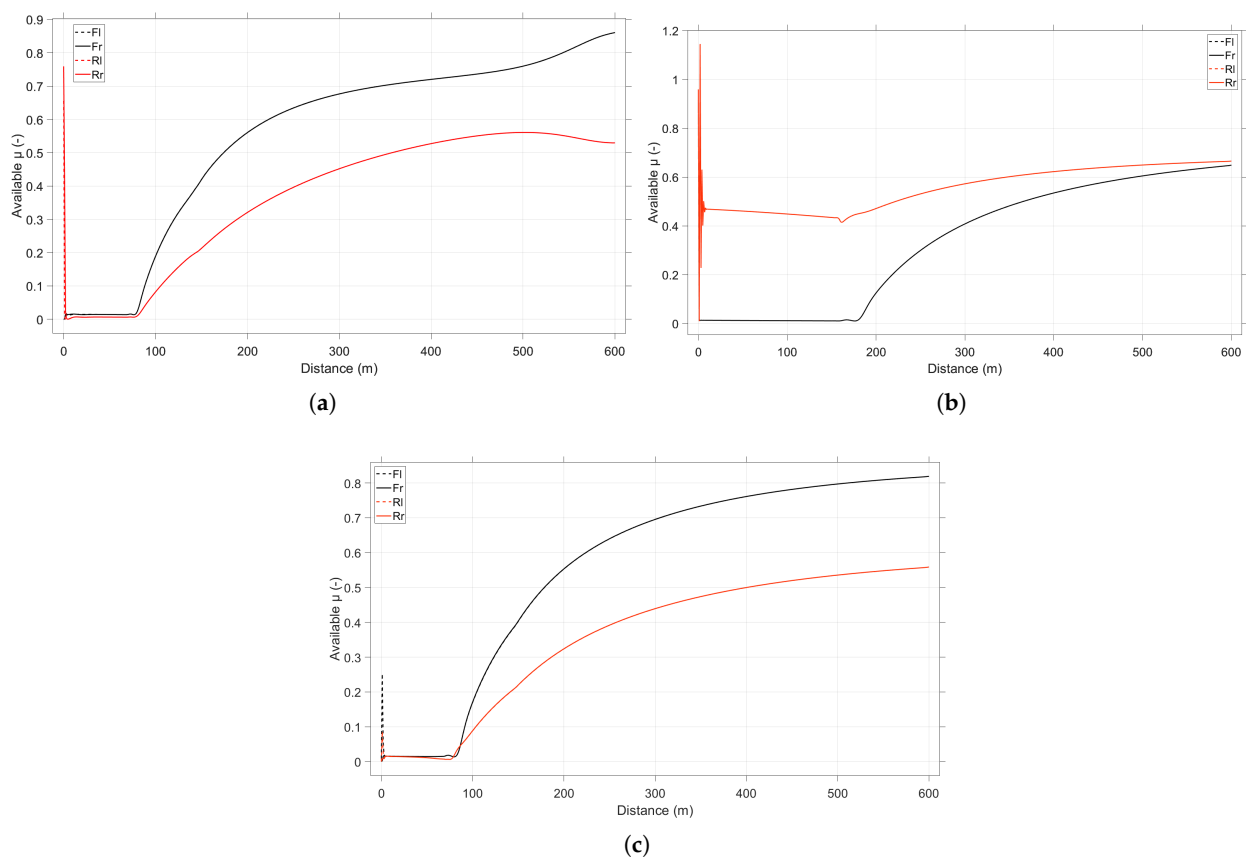


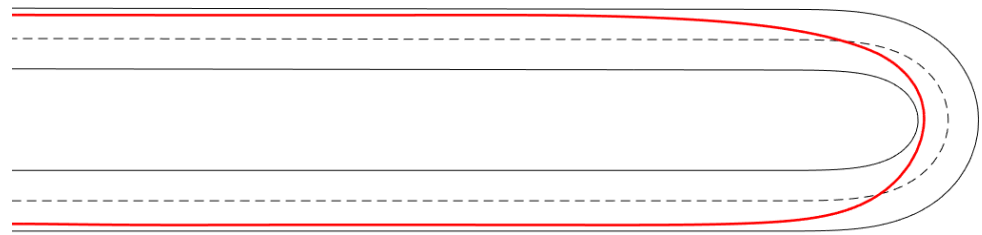
Figure 8. Straight line available μ . (a) With ATD; (b) Without ATD and 50/50 fixed torque distribution; (c) Without ATD and 36/64 fixed torque distribution.

Table 1. Straight line times.

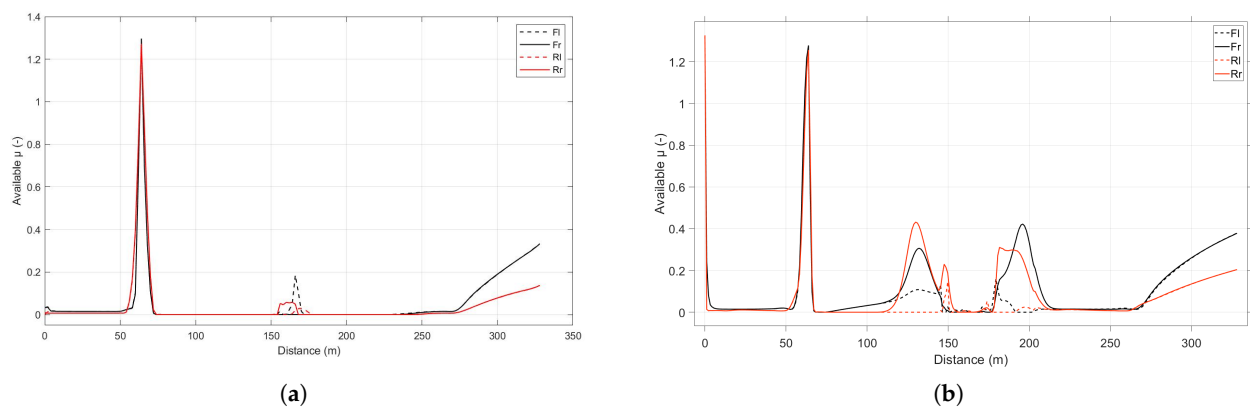
	Time Comparison (s)		
	ATD	50/50	36/64
Time (s)	9.578	10.218	9.607

4.1.2. Hairpin

Here, a big difference in μ usage can be appreciated from the entry to the exit of the turn depicted in Figure 9, which directly translates in a time improvement, as shown in Table 2.

**Figure 9.** Hairpin trajectory.

To understand the plots in Figure 10 it is important to know that the braking starts around 60 m from the start. This instant matches the large spike in the available μ plots. This spike is a non optimality derived from the rate limit imposed to the inputs. The transition from braking to accelerating occurs at 150 m, where another small increase in available μ can be observed for the same reason. Around 260 m the torque reaches the powertrain limit (seen in both figures). In Figure 10a all four wheels are taking advantage of the full available adherence in the rest of situations. From 60 m to 110 m it is straight line braking, and that is why in Figure 10b the rear wheels are both reaching the adherence limit while the front ones are not. From 110 m to 150 m and from 175 m to 210 m the vehicle is braking while turning and accelerating while turning respectively. Thus only one wheel is reaching the limit and it is where the ATD model takes more advantage.

**Figure 10.** Hairpin available μ . (a) With ATD; (b) Without ATD and 36/64 torque distribution.**Table 2.** Hairpin times.

	Time Comparison (s)
ATD	No ATD
10.525	11.207

4.1.3. Barcelona Circuit

The Circuit de Barcelona–Catalunya (Figure 11, is a renowned racing track located in Montmeló, near Barcelona, Spain. It spans 4.675 km and features a mix of high-speed straights and technical corners. The last top qualifying times were 1:18:750 in F1, while the fastest GT car clocked 1:41:060. This circuit hosts a variety of motorsport events, including the Formula (1) Spanish Grand Prix and the MotoGP Catalan Grand Prix, and is a pivotal venue for the development and performance evaluation of racing vehicles.

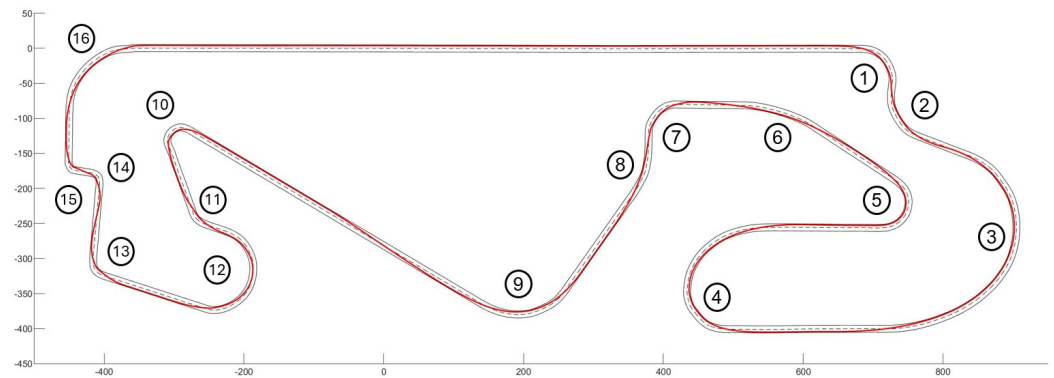


Figure 11. Circuit de Barcelona–Catalunya.

In Figure 12 it can be observed that, as previously stated, the corners are where the ATD obtains more advantage. The two horizontal sections in the graph correspond to the long straights of the circuit.

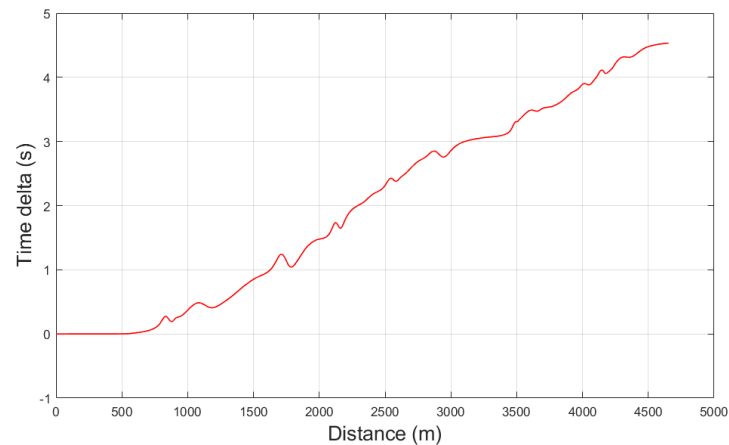


Figure 12. Time delta in Barcelona between the vehicle with and without ATD.

The comparison of the available μ is shown in Figure 13. It is very clear that the vehicle with ATD is using the tyres' full adherence during almost the whole lap. The same situations described earlier can be observed: The spikes in the transitions from accelerating to braking and vice-versa, and the curved segments in the plots corresponding to the powertrain limit (50 m to 550 m, 3000 m to 3300 m and 4500 to the end).

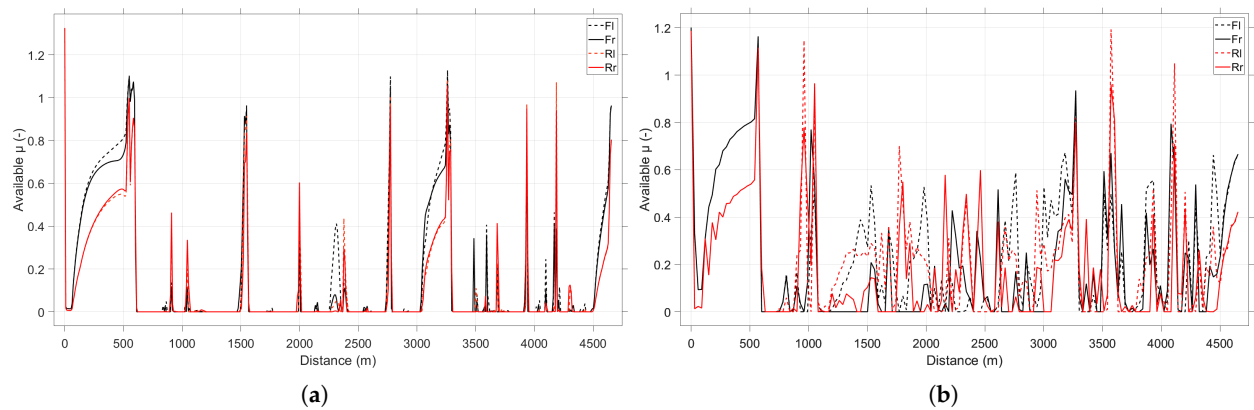


Figure 13. Available μ in Barcelona circuit. (a) With ATD; (b) Without ATD.

4.2. Regen Braking and Energy Saving

The regenerative braking allows the vehicle to obtain back some of the kinetic energy using the electric motor as a generator. This has a lot of benefits, such as improved range and battery downsizing. It also enables the ability to vary the brake torque at each wheel, allowing the vehicle to maximise the grip usage under braking as well.

This section will describe the effects of regenerative braking on range and performance, as well as the influence of the regularisation term that controls the electric energy usage described in Section 3.3.

4.2.1. Regenerative Braking

The aim of the first comparison (Figure 14) is to show the performance improvement under braking that the regenerative braking provides. Using the clutches and the braking torque from the electric motor, the vehicle can effectively use all the available grip at each wheel. This is clearly visible in the following figure, where a side by side comparison of the available μ in the hairpin turn described earlier is shown.

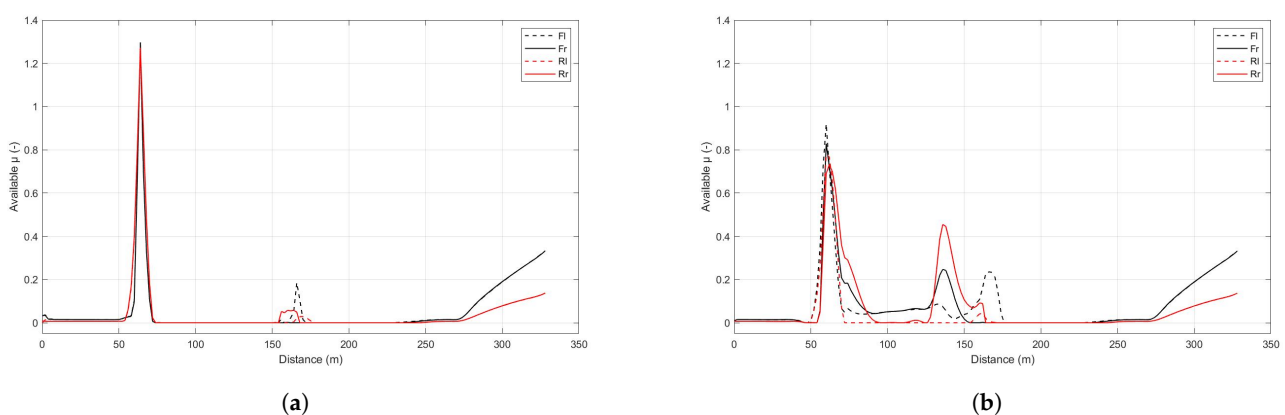


Figure 14. Available μ in a hairpin turn (regen vs. no regen). (a) With regen; (b) Without regen.

This translates in a time improvement, where the vehicle with regenerative braking runs through this section 0.351 s faster, but still ending with more energy in the battery.

To show how much energy can be saved within a lap just by using regenerative braking, a comparison of the net electric energy used during a full lap in Barcelona is shown in Figure 15. Both runs are simulated with exactly the same configuration, except for the regenerative braking capabilities. It is visible how the net energy used reduces when braking, adding to a total difference of 3 kWh in one lap. This means that with a 13 kWh battery, the vehicle could do 18 more laps (4 without regen and 22 with) while performing faster lap times (107.538 s vs. 108.962 s).

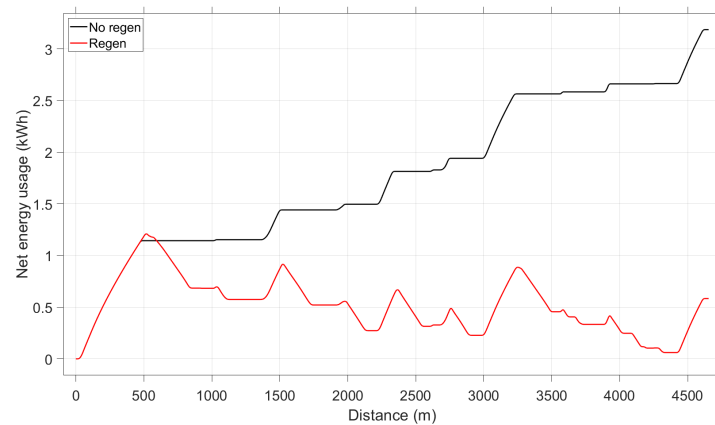


Figure 15. Electric energy usage comparison in a lap at Barcelona.

4.2.2. Energy Saving

To show the effect of the energy regularisation factor r_e , a series of simulations around Barcelona circuit have been performed to plot the value of this factor against lap time and against energy usage. This can be observed in Figure 16.

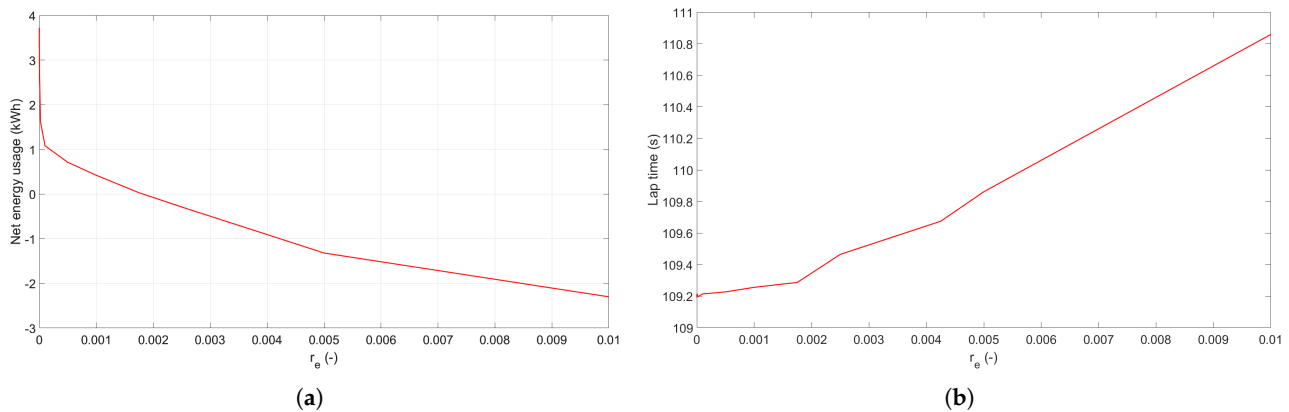


Figure 16. Regularisation factor r_e effect on lap time and energy usage. (a) Net energy usage; (b) Lap time.

It is worth noting that after a certain value of r_e , the energy becomes negative. This means that it regenerates more electric energy than it uses, so the battery has more energy after a full lap.

To achieve this energy usage reduction the solution obtained uses a similar strategy to the “lift and coast”, which is a well known driving style used to save fuel in racing. It starts braking earlier only with regenerative braking, where the motor is using the maximum power, sacrificing a bit of top speed for energy saving. And only when the regenerative braking is reaching its limit, the hydraulic brakes are used. Also, when possible the motor torque is replaced by the ICE torque.

In Figure 17, three different values of r_e are compared. For the greatest value, the vehicle uses the electric machine almost only for regenerative braking, and it starts braking much earlier. This is clearly visible in Figure 18, where the speed profile along the track is compared for these three r_e values. It is also easy to see where the vehicle is using regenerative braking and when it starts to use the hydraulic brakes, as the deceleration is greater when this second system is used.

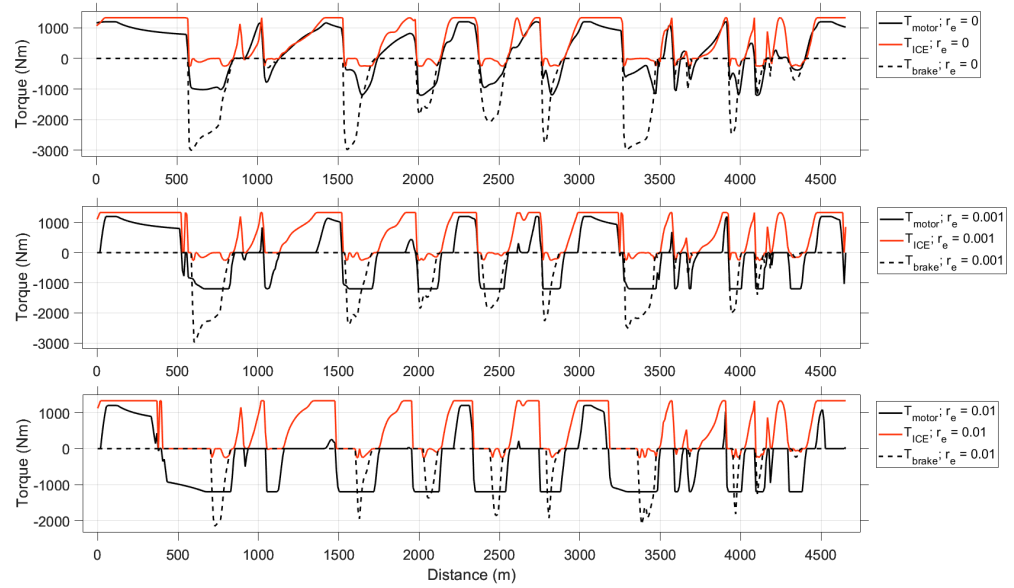


Figure 17. Torque curves comparison with different r_e values.

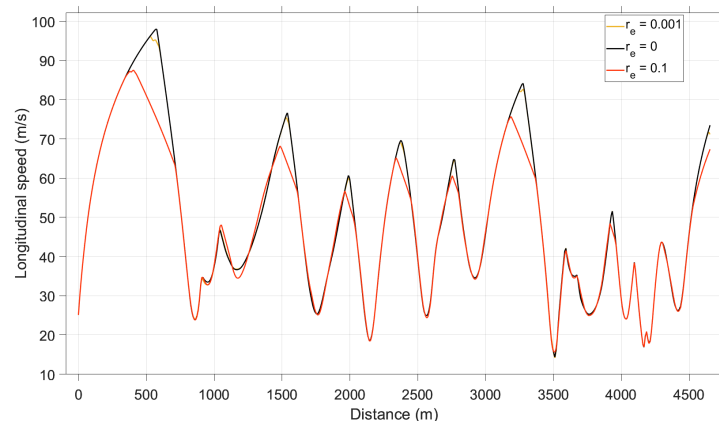


Figure 18. Longitudinal speed comparison with different r_e values.

This image illustrates the sacrifice in top speed made for regenerating more energy, as the speed along the whole track is almost the same except for the hardest braking points.

It can be observed how the speed profile along the whole circuit is almost identical independent of the r_e value, while the greater the value of r_e the sooner the vehicle starts to brake using regen, to start braking with the hydraulic brakes after. Regen braking can be distinguished from hydraulic braking by observing the rate at which speed decreases (greater with hydraulic braking).

In Figure 19 the influence of the last term in the objective function is shown in terms of electric energy usage. Depending on the value of r_e (in this case $r_e = 0.001$), this term will have more or less impact on the lap time. Here, the time difference is just 0.046 s in a full lap in Barcelona (Around 108.2 s).

The energy usage is calculated after the simulation is finished using trapezoidal integration. Knowing the elapsed time between each pair of grid points (t_k) and the motor's torque and speed at each grid point ($T_{motor,k}$ and Ω_k), the energy used is:

$$E = \sum_{k=1}^N \frac{T_{motor,k} \cdot \Omega_k + T_{motor,k+1} \cdot \Omega_{k+1}}{2} \cdot (t_{k+1} - t_k) \quad (54)$$

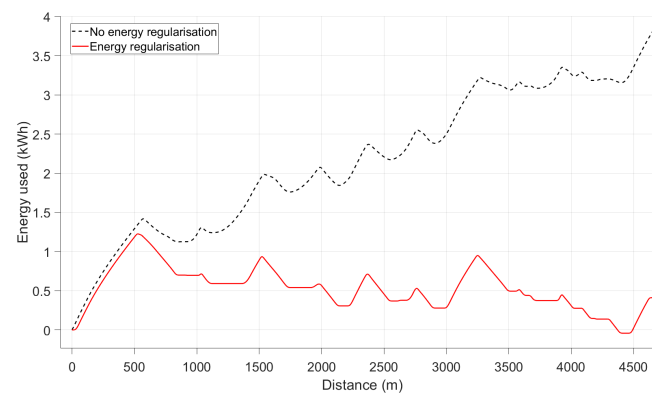


Figure 19. Energy usage comparison.

4.3. Active Aerodynamics

Active aerodynamics can be used to improve a car's performance around a track—a clear example of this is DRS, used in many motorsport categories. Without the restrictions of competition rules, more sophisticated systems are possible, for example, using variable angle flaps in both the front and the rear of the car it would be possible to control the handling balance. However, this study focuses not on performance gains but on how optimization techniques can aid the development of these systems.

The control of a simple active aerodynamics system like the one presented in this work (Equation (24)) may seem at first glance a straightforward problem. The model varies between a high downforce-high drag configuration and a low downforce-low drag configuration. It would be expected that the highest downforce configuration is preferred during braking and cornering, whereas the lowest drag configuration is used in the straights. This has been observed to be the case, as shown in Figure 20, for the single-track model used for the initialisation of the decision variables.

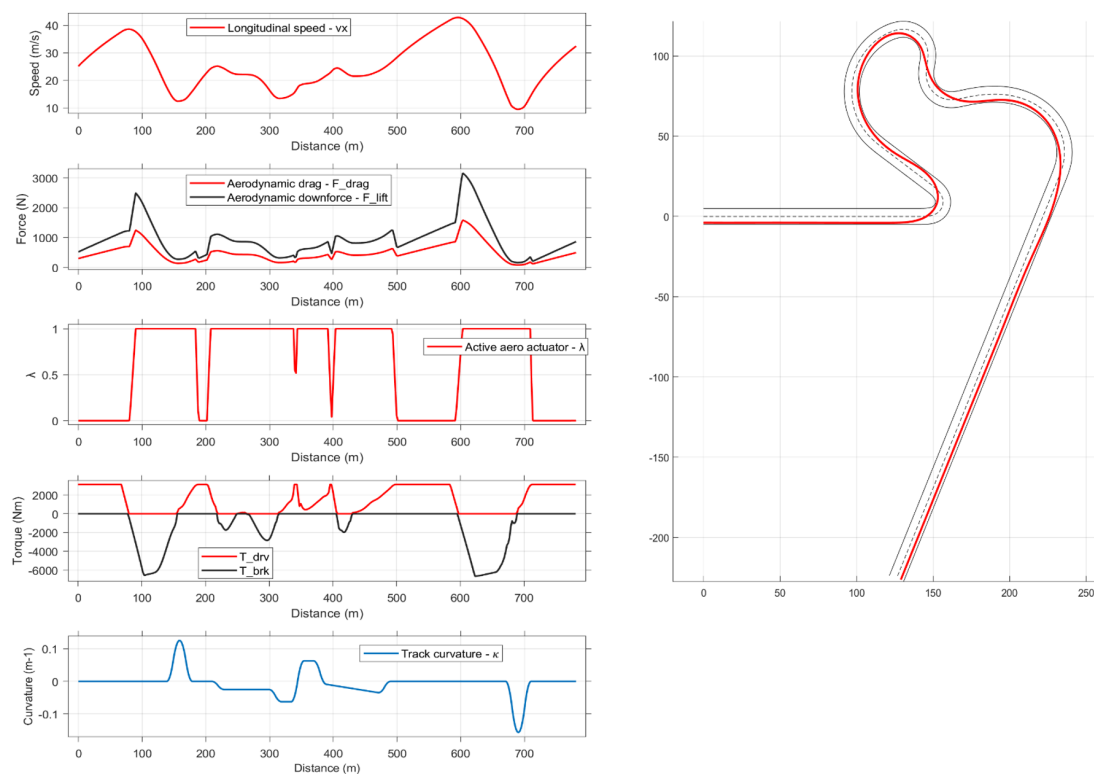


Figure 20. Control of active aerodynamics for a single-track model with no energy recovery (left) and track (right).

Active aerodynamics constitute a particularly interesting case study combined with a hybrid powertrain and active torque distribution because of the complex interactions between these subsystems. For example, during a hard braking event, a higher downforce would increase the available grip of the tyres, allowing harder braking and potentially more energy recovery—provided the maximum power of the electrical system is not a limitation. In turn, a higher downforce configuration generally implies a higher drag, which would slow down the car, reducing the amount of energy available for recovery. This example illustrates why finding the optimal actuation of active aerodynamics is not such a trivial problem in this case.

Figure 21 shows the solution for different values of re . As expected, the low drag configuration is always preferred in the straights. However, the extreme maximum downforce and maximum drag configuration is never used. Instead, the active aerodynamics are regulated to achieve the optimal balance. It can also be observed how this balance is different for each value of re as the simulation finds different strategies to improve energy usage whilst maximising performance. For instance, it can be observed that the actuation of active aerodynamics is delayed during braking until the hydraulic brakes are applied, which maximises energy recovery when using the higher value of re .

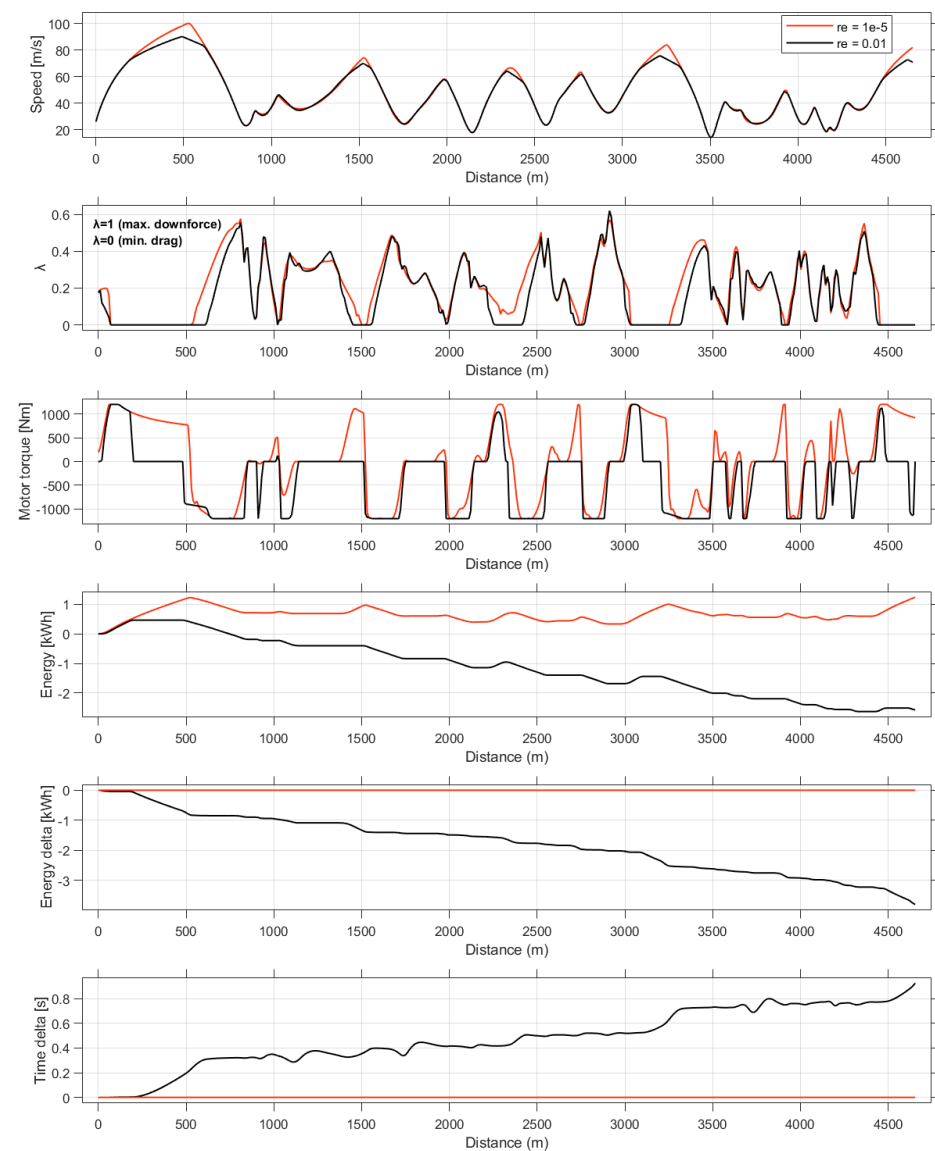


Figure 21. Control of active aerodynamics for different values of re in a lap at Barcelona.

4.4. Parameter Optimisation

Up to this point, we have demonstrated that the simulator works effectively and provides results that align well with real-life expectations. However, our current focus extends beyond validation to illustrate the potential real-world applications of this work. Integrating parameter optimization within the lap time simulator adds significant value by allowing for the simultaneous optimization of key parameters, such as weight distribution, alongside active systems like active torque distribution and active aerodynamics. This holistic optimization approach is crucial because optimizing parameters in isolation can lead to sub-optimal solutions. By considering all these elements together, we ensure that the overall vehicle is tuned for maximum performance. This enables engineers to adjust key parameters, such as weight distribution or brake balance, during the vehicle design phase.

In this study, parameter optimization was carried out both with and without active systems to compare the results. The optimal weight balance was found to vary from 49/51 front-to-rear in the absence of active systems to 52/48 when active systems were utilized, demonstrating a substantial difference in optimal weight distribution. Figure 22 illustrates the improvement in lap time around the Barcelona–Catalunya circuit, highlighting the time delta for the two weight balances in a vehicle equipped with active systems.

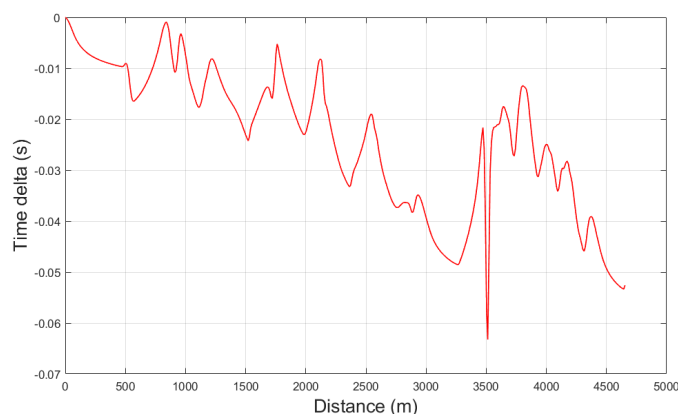


Figure 22. Time delta between 49/51 and 52/48 weight balance.

The results indicate that in this particular case, the performance improvement is minimal (only 0.05 s). This is likely due to the use of a very flexible and optimal ATD strategy. In a vehicle with a non-optimal control strategy, changing the weight distribution might have a greater impact. This is evident in Figure 23, which compares the average absolute forces between the left and right tyres on each axle throughout the lap. The vehicle with a 52/48 weight balance produces more force on the front axle (dashed line) and less on the rear axle (solid line). As a result, both vehicles utilize almost the same adherence (μ), with the difference in performance mainly attributed to load sensitivity. The 52/48 weight balance demonstrates a more evenly distributed weight throughout the lap.

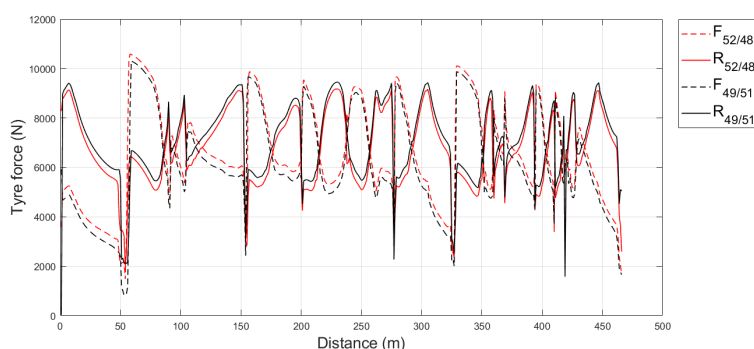


Figure 23. Tyre force comparison between 49/51 and 52/48 weight balance.

Nevertheless, the findings demonstrate that performing parameter optimization while considering all active systems not only improves the vehicle's speed but also offers valuable insights into the impact of design choices on performance. This approach, therefore, facilitates more informed decision-making. The practical implications of these findings highlight the simulator's value as a critical tool in the design and engineering process.

4.5. Computational Times

The computational time is a very important factor to consider when simulating. There are numerous factors that affect this like the initial guess or the way of expressing some equations are the ones that have been found to have the greatest effect, although they also affect the ease with which the problem converges to a feasible solution. To better understand the current level of computational effort, a few examples are provided (see Table 3). These examples involve fixing the value of r_e (0.001) and using ATD under various conditions: with and without active aerodynamics, as well as with and without parameter optimization. The table below shows the simulation times for each scenario.

Table 3. Computational times for different simulations around Barcelona circuit.

Computational Times	
Model	Elapsed time (s)
No active aero without param. opt	1714.7
Active aero without param. opt.	2038.8
Active aero with param. opt.	2062.8

Based on the data presented in the previous table, the introduction of parameter optimization does not significantly elevate computational costs. This is because it does not modify the model itself but merely adds one more variable to the optimization process. This is another reason why it emerges as an exceptionally valuable tool. Its implementation provides substantial benefits without imposing a heavy computational burden.

5. Conclusions

In this piece of work, we have demonstrated that a simulation approach to optimizing both vehicle parameters and control strategies effectively optimises the overall vehicle performance. This holistic approach allows for simultaneous optimization of key parameters such as weight distribution, ATD and active aerodynamics, providing a great tool for vehicle design that can help to produce faster and more efficient vehicles. For instance, the simulations showed a reduction in energy usage of 80% when introducing regenerative braking. Additionally, optimizing the torque distribution in various driving scenarios resulted in up to a 6% increase slow cornering times and a 4% reduction lap times around Barcelona Circuit.

For future work, we are currently focused on incorporating suspension kinematics and dynamics. This includes accounting for ride height, roll and pitch variations, and load transfer dynamics. The goal is to enhance the accuracy of tyre performance and aerodynamics. Another potential area of future work is exploring different applications. For example, using the parameter optimization feature, we could apply this workflow to optimize the battery pack size or the e-motor's power by establishing a relationship between capacity/power and weight.

Author Contributions: Conceptualization, A.J.E., A.Z.C. and E.S.; Methodology, A.J.E. and A.Z.C.; Software, A.J.E. and A.Z.C.; Validation, A.J.E. and A.Z.C.; Formal analysis, A.J.E. and A.Z.C.; Investigation, A.J.E. and A.Z.C.; Resources, E.S.; Writing—original draft, A.J.E. and A.Z.C.; Writing—review and editing, A.J.E., A.Z.C. and E.S.; Visualization, A.J.E. and A.Z.C.; Supervision, E.S.; Project administration, E.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: The authors would like to thank Fritiof Hegardt and Dragos-Mihai Postariu from Koenigsegg Automotive AB for their assistance and valuable input during the development of this work. Thank you for your dedication and unwavering support.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MLTP	Minimum lap time problem
OC	Optimal control problem
NLP	Non-linear program
ICE	Internal combustion engine
ATD	Active torque distribution

References

1. Scherenberg, H. Mercedes-Benz Racing Design and Cars Experience (SAE Technical Paper; Report No.: 1958-01-01). 1958. Available online: <https://www.sae.org/publications/technical-papers/content/580042/> (accessed on 1 May 2022)
2. Casanova, D. On Minimum Time Vehicle Manoeuvring: The Theoretical Optimal Lap. Master's Thesis, Cranfield University, Bedford, UK, 2000.
3. Perantoni, G.; Limebeer, D.J. Optimal control for a Formula One car with variable parameters. *Veh. Syst. Dyn.* **2014**, *52*, 653–678. [CrossRef]
4. Liang, J.; Feng, J.; Fang, Z.; Lu, Y.; Yin, G.; Mao, X.; Wu, J.; Wang, F. An Energy-Oriented Torque-Vector Control Framework for Distributed Drive Electric Vehicles. *IEEE Trans. Transp. Electr.* **2023**, *9*, 4014–4031. [CrossRef]
5. Adeleke, O.P.; Li, Y.; Chen, Q.; Zhou, W.; Xu, X.; Cui, X. Torque Distribution Based on Dynamic Programming Algorithm for Four In-Wheel Motor Drive Electric Vehicle Considering Energy Efficiency Optimization. *World Electr. Veh. J.* **2022**, *13*, 181. [CrossRef]
6. Liang, J.; Wang, F.; Feng, J.; Zhao, M.; Fang, R.; Pi, D.; Yin, G. A Hierarchical Control of Independently Driven Electric Vehicles Considering Handling Stability and Energy Conservation. *IEEE Trans. Intell. Veh.* **2024**, *9*, 738–751. [CrossRef]
7. Mangia, A.; Lenzo, B.; Sabbioni, E. An integrated torque-vectoring control framework for electric vehicles featuring multiple handling and energy-efficiency modes selectable by the driver. *Meccanica* **2021**, *56*, 991–1010. [CrossRef]
8. Velenis, E.; Tsiotras, P. Minimum-Time Travel for a Vehicle with Acceleration Limits: Theoretical Analysis and Receding-Horizon Implementation. *J. Optim. Theory Appl.* **2008**, *138*, 275–296. [CrossRef]
9. Ruslan, N.A.I.; Amer, N.H.; Hudha, K.; Kadir, Z.A.; Ishak, S.A.F.M.; Dardin, S.M.F.S. Modelling and control strategies in path tracking control for autonomous tracked vehicles: A review of state of the art and challenges. *J. Terramech.* **2023**, *105*, 67–79. [CrossRef]
10. Bianco, N.D.; Bertolazzi, E.; Biral, F.; Massaro, M. Comparison of direct and indirect methods for minimum lap time optimal control problems. *Veh. Syst. Dyn.* **2019**, *57*, 665–696. [CrossRef]
11. Brayshaw, D.L.; Harrison, M.F. A quasi steady state approach to race car lap simulation in order to understand the effects of racing line and centre of gravity location. *Proc. Inst. Mech. Eng. Part J. Automob. Eng.* **2005**, *219*, 725–739. [CrossRef]
12. Betts, J.T. Survey of numerical methods for trajectory optimization. *J. Guid. Control. Dyn.* **1998**, *21*, 193–207. [CrossRef]
13. Best, M.; Gordon, T. Simultaneous Optimisation of Vehicle Parameter and Control Action to Examine the Validity of Handling Control Assumptions. Master's Thesis, Loughborough University, Loughborough, UK, 2002.
14. Betts, J.T. *Practical Methods for Optimal Control Using Nonlinear Programming*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2001.
15. Kelly, M. Trajectory Optimisation. Overview and Tutorial. Cornell Scientific Computation and Numerics Seminar. Available online: https://www.matthewpeterkelly.com/research/MatthewKelly_IntroTrajectoryOptimization_SIAM_Review_2017.pdf (accessed on 15 May 2022).
16. Biegler, L.T. Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes. 2010. Available online: <http://cepac.cheme.cmu.edu/pasilectures/biegler/BieglerLecture.pdf> (accessed on 20 May 2022).
17. Berrut, J.P.; Trefethen, L.N. Barycentric Lagrange interpolation. *SIAM Rev.* **2004**, *46*, 501–517. [CrossRef]
18. Trefethen, L.N. *Approximation Theory and Approximation Practice*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2013.
19. Kang, W. Rate of convergence for the Legendre pseudospectral optimal control of feedback linearizable systems. *J. Control. Theory Appl.* **2010**, *8*, 391–405. [CrossRef]
20. Ipopt: Documentation. Available online: <https://coin-or.github.io/Ipopt/> (accessed on 1 June 2022).

21. Andersson, J.A.; Gillis, J.; Horn, G.; Rawlings, J.B.; Diehl, M. CasADi: A software framework for nonlinear optimization and optimal control. *Math. Program. Comput.* **2019**, *11*, 1–36. [[CrossRef](#)]
22. Diehl, M.; Gros, S. Numerical Optimal Control 2017. Available online: <https://www.syscop.de/teaching/ss2017/numerical-optimal-control> (accessed on 1 June 2022).
23. Pacejka, H.B. *Tire and Vehicle Dynamics*; Butterworth Heinemann: Oxford, UK, 2012.
24. Cervone, D.; Sessa, B.; Arsie, I.; Pianese, C.; Polverino, P. A Comprehensive Hybrid Vehicle Model for Energetic Analyses on Different Powertrain Architectures. In Proceedings of the 14th International Conference on Engines & Vehicles, Napoli, Italy, 15–19 September 2019; SAE Technical Papers; SAE: Warrendale, PA, USA, 2019. [[CrossRef](#)]
25. Geng, S.; Schulte, T. Real-time powertrain models of hybrid electric vehicles. *SAE Int. J. Altern. Powertrains* **2015**, *4*, 34–45. [[CrossRef](#)]
26. Geng, S.; Meier, A.; Schulte, T. Model-based optimization of a plug-in hybrid electric powertrain with multimode transmission. *World Electr. Veh. J.* **2018**, *9*, 12. [[CrossRef](#)]
27. Regera—Technical Specifications. Available online: <https://www.koenigsegg.com/technical-specifications-regera> (accessed on 1 June 2024).
28. Rao, A. A Survey of Numerical Methods for Optimal Control. *Adv. Astronaut. Sci.* **2010**, *135*, 497–528.
29. Kelly, M. An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation. *SIAM Rev.* **2017**, *59*, 849–904. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.