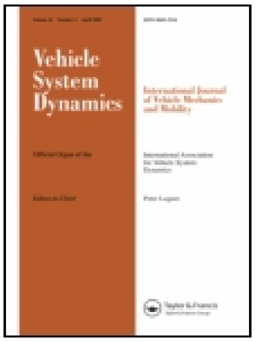


Taylor & Francis
Taylor & Francis Group

Vehicle System Dynamics

ISSN: 0042-3114 (Print) 1744-5159 (Online) Journal homepage: http://www.tandfonline.com/loi/nvsd20

Time-optimal control of the race car: a numerical method to emulate the ideal driver

D. P. Kelly & R. S. Sharp

To cite this article: D. P. Kelly & R. S. Sharp (2010) Time-optimal control of the race car: a numerical method to emulate the ideal driver, Vehicle System Dynamics, 48:12, 1461-1474, DOI: 10.1080/00423110903514236

To link to this article: https://doi.org/10.1080/00423110903514236

Published online: 31 Aug 2010.

Vehicle System Dynamics
Vol. 48, No. 12, December 2010, 1461–1474

Taylor & Francis
Taylor & Francis Group

Time-optimal control of the race car: a numerical method to emulate the ideal driver

D.P. Kelly[a]* and R.S. Sharp[b]

[a]*Control Systems Group, Force India Formula 1 Team, Silverstone, Northamptonshire, NN12 8TJ, UK;*
[b]*Faculty of Engineering and Physical Sciences, University of Surrey, Guildford, Surrey, GU2 7XH, UK*

(*Received 7 August 2009; final version received 15 November 2009; first published 31 August 2010*)

A numerical method for the time-optimal control of the race car is presented. The method is then used to perform the role of the driver in numerical simulations of manoeuvres at the limit of race car performance. The method does not attempt to model the driver but rather replaces the driver with methods normally associated with numerical optimal control. The method simultaneously finds the optimal driven line and the driver control inputs (steer, throttle and brake) to drive this line in minimum time. In principle, the method is capable of operation with arbitrarily complex vehicle models as it requires only limited access to the vehicle model state vector. It also requires solution of the differential equation representing the vehicle model in only the forward time direction and is hence capable of simulating the full vehicle transient response.

Keywords: race car; lap-time simulation; time-optimal control

1. Introduction

The general problem in motor racing is to design and set up a vehicle so as to allow it to complete a given manoeuvre or a set of manoeuvres in the minimum possible time. It is a common practice in the motor racing industry to construct mathematical models of racing vehicles and use numerical optimisation methods to estimate minimum manoeuvre time. This is commonly referred to as lap-time simulation. The principal challenge faced by these methods is to emulate those functions normally performed by the driver. Any such method must synthesise the driver control inputs that will produce a vehicle trajectory that is time-optimal and satisfies the constraint that the vehicle remains on the race track. In the method presented here, it is not intended to model the actual processes carried out by a human driver. The driver is replaced with techniques usually associated with optimal control problems. The use of additional constraints on vehicle state to emulate driver performance limits is considered in a future paper. The method is described in two stages. First, a method is formulated that is suitable for short manoeuvres (<400 m). The method is then extended for manoeuvres of arbitrary length.

*Corresponding author. Email: pat.kelly@forceindiaf1.com

ISSN 0042-3114 print/ISSN 1744-5159 online

DOI: 10.1080/00423110903514236
http://www.informaworld.com

2. Formulation of the method for short manoeuvres

A manoeuvre is defined by a number of ‘Slalom’ poles. Each pair of poles is connected by a line referred to as a ‘wayline’. To complete the manoeuvre, the vehicle must cross each wayline. The manoeuvre time is the time elapsed between crossing the first and the last waylines. The time-optimal control problem is formulated as a finite inequality constrained nonlinear programming (NLP) problem [1] by discretising the driver control inputs and path constraints. The inequality constrained NLP problem can be stated as follows:

$$\text{Minimise}: \ f(\bar{x}), \quad \bar{x} \in R^{n_x} \tag{1}$$

$$\text{Subject to}: \ c_j(\bar{x}) \leq 0, \ldots, j = 1 \ldots J, \tag{2}$$

where $f(\bar{x})$ is the scalar-valued objective function (Equation (1)) and $\bar{x}$ the n_x dimensional vector of independent variables. The scalar-valued functions $c_j(\bar{x})$ compute the J inequality constraints (Equation (2)).

2.1. *Definition of the objective function*

A manoeuvre consists of $N + 1$ waylines. The time at which the vehicle centre of gravity crosses the wayline is given by T_n, for $n = 0, 1, \ldots, N$. The value of the objective function is simply $T_{\mathrm{M}} = T_{\mathrm{N}} - T_0$.

2.2. *Representation of the vehicle model*

A general vehicle model will typically be described by a set of ordinary differential (ODE) or differential algebraic (DAE) equations. An example of ODE is given in Equation (3):

$$\frac{\mathrm{d}(\bar{m}(t))}{\mathrm{d}t} = \bar{V}(\bar{m}(t), \bar{u}(t)), \tag{3}$$

where $\bar{m}(t) \in R^{n_m}$ is the n_m dimensional state vector and $\bar{u}(t) \in R^{n_u}$ the n_u dimensional control vector at time t. The dimension of the state vector is determined by the complexity of the vehicle model. The algorithm requires access to only those elements of the state vector that can generally be considered to be common to all vehicle models. The elements required are given in Table 1. Computation of the objective function and constraints first involves solving an initial value problem in this ODE. The objective function is computed from the terminal time. The constraints are computed from the state trajectory produced during this solution.

2.3. *Representation of the driver controls*

The control inputs available to a driver in a typical vehicle model consist of engine torque demand, front road wheel steering angle, braking torque demand, clutch torque transfer and gear ratio. For the purposes of this work, this is simplified to a two-dimensional control vector consisting of a combined driving/braking torque demand P and steering angle S [2,3]. The independent variables used by the optimiser to represent the driving/braking ‘pedal position’ P are the numbers in the range $[-1, 1]$. This can be thought of as two ranges. The range $[0, 1]$ represents the on-throttle regime. The value 1 represents the maximum available engine driving torque. The value 0 represents the closed throttle scenario producing maximum engine braking. The range $[-1, 0]$ represents the braking regime. The distribution of torques between axles and wheels including gearbox and differential is described in more detail in Appendix 1. The steering control S simply

Table 1. Vehicle state vector elements required by the method.

Symbol	Description
d_x, d_y	Displacement of the vehicle centre of gravity in the plane
v_x	Longitudinal velocity at the vehicle centre of gravity
v_y	Lateral velocity at the vehicle centre of gravity
r	Vehicle yaw rate
A_y	Vehicle lateral acceleration
κ_{LF}	Slip ratio at the left front wheel
κ_{RF}	Slip ratio at the right front wheel
κ_{LR}	Slip ratio at the left rear wheel
κ_{RR}	Slip ratio at the right rear wheel
$F_{z\text{LF}}$	Vertical load at the left front wheel
$F_{z\text{RF}}$	Vertical load at the right front wheel
$F_{z\text{LR}}$	Vertical load at the left rear wheel
$F_{z\text{RR}}$	Vertical load at the right rear wheel

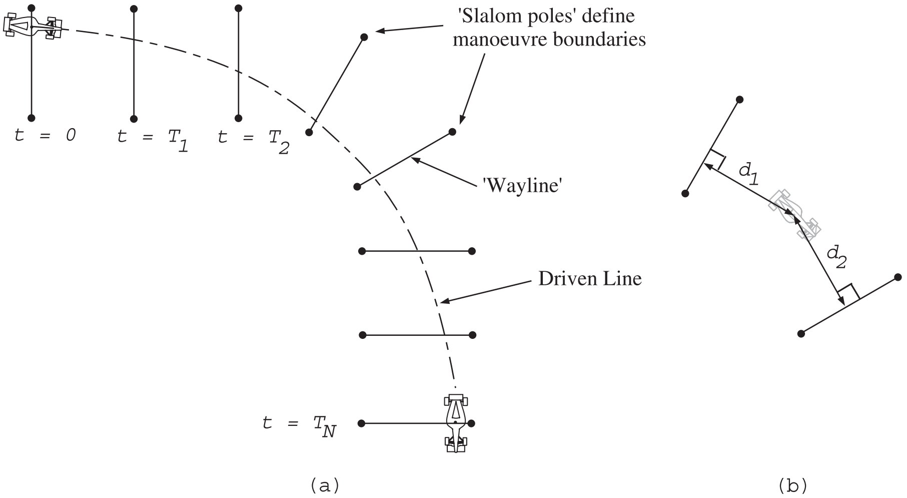


Figure 1. Method definition.

represents the steer angle applied to the front wheels. The vehicle controls are specified as discrete values at each wayline. Interpolation is then used to produce the continuous signals for input to the vehicle model. The vector of independent variables in the NLP problem $\bar{u}$ specifies the control values at the time instants when the vehicle crosses each wayline. At any time instant, the control value is computed based on d_w, the relative distances from the centre of gravity of the vehicle to the nearest points on the two closest waylines as described in Equation (4) and illustrated in Figure 1(b). If P_n and P_{n+1} represent the throttle/brake values at waylines n and $n+1$, then the throttle/brake value is given by

$$d_w(t) = \frac{d_1(t)}{d_1(t) + d_2(t)}, \tag{4}$$

$$P(t) = P_n + (P_{n+1} - P_n) * d_w(t), \tag{5}$$

where d_1 and d_2 are defined in Figure 1(b). In a similar fashion, the steer value is given by

$$S(t) = S_n + (S_{n+1} - S_n) * d_w(t). \tag{6}$$

The distance between the waylines effectively specifies the bandwidth available to the control system. It is shown in Section 3.1 that as the distance between the waylines is reduced, the manoeuvre time decreases until a limiting value is reached. At this point, the controller has sufficient bandwidth for time-optimal control. The distance between the waylines need not be uniform. Their density can be increased as desired around the points of interest, for example, the ‘braking point’ for a corner. In [4], consideration is given to the wayline spacing that produces a controller with a bandwidth representative of a professional driver.

2.4. *Formulation of the constraints*

Constraints are used to place bounds on certain vehicle model state variables at various points along the manoeuvre. The function shown in Figure 2 and Equation (7) is used to allow a bound to be represented as a smooth inequality constraint. The function $B(x)$ is ≤ 0 when $-x_{\text{lim}} \leq x \leq x_{\text{lim}}$.

$$B(x, x_{\text{lim}}) = \left(\frac{x}{x_{\text{lim}}}\right)^2 - 1. \tag{7}$$

The constraints can be considered in two categories. The first is the essential constraint that the vehicle remains within the bounds of the race track. The second is used to prevent the exploration of highly nonlinear (and sub-optimal) areas of vehicle behaviour as the NLP code iterates towards the optimal solution.

2.4.1. *Race track (displacement) constraints*

A race track constraint is computed at the time instant when the vehicle centre of gravity crosses the wayline. If w_n is the distance of the vehicle centre of gravity from the centre of the wayline (Figure 2) and W_n is the length of the wayline (the width of the track), then the displacement constraint is given in Equation (8) as follows:

$$C_d(n) = B\left(w_n, \frac{W_n}{2}\right), \quad n = 0, 1, N. \tag{8}$$

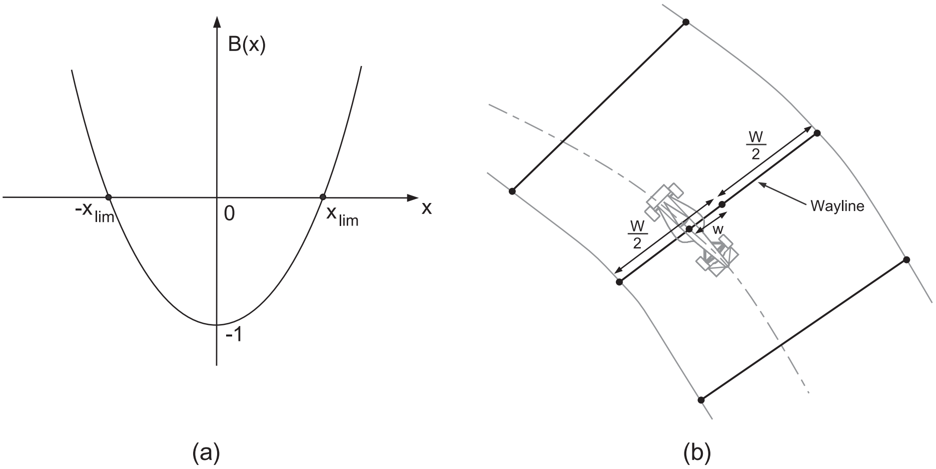


Figure 2. (a) Bounding function and (b) displacement constraint.

This provides a smooth function that is less than or equal to zero when the vehicle centre of gravity is within the track boundary and greater than zero otherwise. Only the vehicle centre of gravity is constrained to the racetrack for simplicity. The method is easily extended to constrain the displacements of each of the four wheels if necessary.

2.4.2. *Stability constraints*

It is inherent in the nature of the problem that small perturbations of the controls associated with time-optimal trajectories will result in large changes in the trajectory corresponding to a 'spin'. This results in highly nonlinear behaviour of the objective function [4]. The functions $\mathrm{Max}(x, T_1, T_2)$ and $\mathrm{Min}(x, T_1, T_2)$ are defined as the maximum and minimum values taken by x in the time interval $T_1 \leq t \leq T_2$.

$$C_\kappa(n) = B(\mathrm{Max}(|\kappa|, T_n, T_{n+1}), \kappa_{\mathrm{lim}}), \tag{9}$$

$$C_{\mathrm{uang}}(n) = B(\mathrm{Max}(|\mathrm{uang}|, T_n, T_{n+1}), \mathrm{uang}_{\mathrm{lim}}), \tag{10}$$

$$C_\beta(n) = B(\mathrm{Max}(|\beta|, T_n, T_{n+1}), \beta_{\mathrm{lim}}), \tag{11}$$

$$C_{Fz}(n) = -\mathrm{Min}(F_z, T_n, T_{n+1}). \tag{12}$$

Four constraints of the form shown in Equation (9) are used to limit the slip ratios at the tyres. Equations (10) and (11) limit the understeer angle and sideslip angle at the vehicle centre of gravity. Four constraints of the form shown in Equation (12) ensure that the minimum vertical load on any tyre is $\geq$0. Computation of the wheel loads in the model must allow 'virtual' wheel loads to become negative so that this constraint is well defined for all vehicle model states. The understeer angle $\theta_{\mathrm{uang}} = \delta - \delta_k$ is defined as the difference between the steer angle δ and the kinematic steer angle $\delta_k = Lr/v_x$, where L is the vehicle wheelbase and r the yaw rate. The sideslip at the vehicle centre of gravity is given by $\beta = \tan^{-1} v_y/v_x$. The bounding function B defined in Equation (7) is used in Equations (9)–(10) to allow the bounds on these parameters to be expressed as an inequality constraint.

2.5. ***Choice of NLP and gradient computation algorithms***

A scheme has been described by which the time-optimal race car driving problem is formulated as an inequality constrained NLP problem. The 'Feasible Sequential Quadratic Programming' code (FSQP) [5] is used to solve this NLP problem. A full description of the operation of FSQP is beyond the scope of this article and only a subset of its capabilities is used in this application. A simplified description of the operation of FSQP to the inequality constrained NLP problem is given below. First, the discretised vehicle control values are mapped onto the NLP-independent variables x_k as shown in Equation (13).

$$x_{2k} = P_k, \quad x_{2k+1} = S_k, \quad k \in [0, 1, \dots, N]. \tag{13}$$

The driving/braking control P is in the range [-1,1] and the road wheel steer angle S is expressed in radians. No scaling factors are used. The vector space occupied by the independent variables can be considered as two regions. The subset of the vector space for which all the constraint functions are satisifed ($\leq$0) is known as the 'feasible region'. The set of all remaining points is termed the 'infeasible region'. The set of all points for which one or more constraint functions are equal to zero with all the remaining constraints less than zero represents the boundary of the feasible region. In all but trivial cases of the time-optimal race car control problem, the solution lies on this boundary. Sequential quadratic programming methods proceed iteratively generating

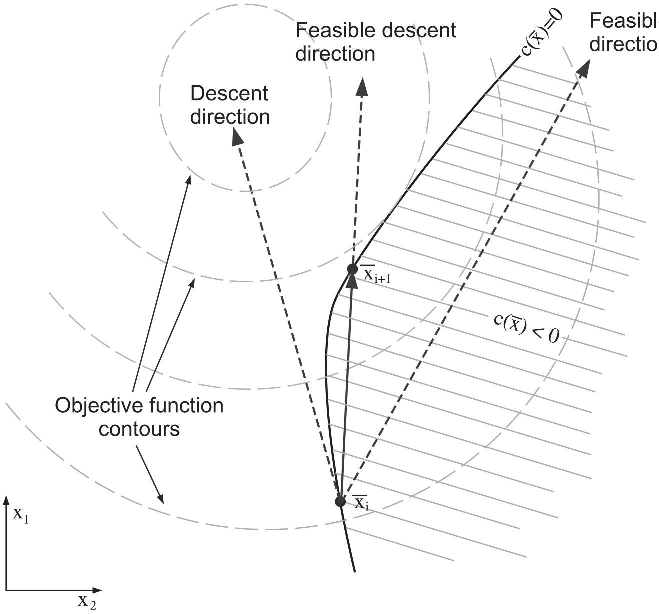


Figure 3. FSQP generation of feasible iterates.

a series of points $\bar{x}_0, \bar{x}_1, \ldots, \bar{x}^*$ approaching an optimal value. All such iterates generated by FSQP are in the feasible region. The operation of FSQP to a simple example problem with two independent variables and one inequality constraint is shown in Figure 3. The figure shows the process by which a new iterate is generated. The boundary of the feasible region lies along the curve marked $c(\bar{x}) = 0$. The feasible region lies to the right of this curve and is shown hatched. At each iterate the following steps take place:

(1) A quadratic model of the objective function about the current iterate is updated. The model is constructed using the first-order gradients of the objective function with respect to the independent variables. FSQP implements a recursive approximation of the second-order derivatives using the change in the first-order gradients between iterates.
(2) Newton's method [1] is used to generate a descent direction for this quadratic model. This descent direction may typically point into the infeasible region (as shown in the example). Following this direction, will reduce manoeuvre time but may cause the car to drive off the circuit.
(3) A further essentially arbitrary feasible direction is generated (pointing into the feasible region) [6].
(4) These two directions are combined to form a 'feasible descent direction' that offers a reduction in the objective function and points into the feasible region. Following this direction will reduce the manoeuvre time, while keeping the car on the track.
(5) The objective and constraint functions are then sampled along this direction in a 'line search' procedure. The line search terminates at a point representing a minimum of the objective function along this line or where the line meets the boundary of the feasible region.

FSQP terminates when it is no longer possible to generate a search direction that offers reduction in the objective function while maintaining feasibility. Only those inequality constraints that are 'active' (equal to zero) at the solution have any influence upon it. In this application, the active constraints at the solution are associated with the waylines where the vehicle reaches the edge of the circuit.

FSQP must be provided with the first-order derivatives of the objective and constraint functions with respect to the independent variables. These are estimated using simple finite differences. The practical benefit to this approach as opposed to the methods that produce gradients analytically [7] is that it allows the vehicle model to be treated as a ‘black box’. Hence, at least in principle, the method is capable of operation with arbitrarily complex vehicle models.

3. Extension to manoeuvres of arbitrary length

The sensitivities of the objective function to the vehicle controls decrease with n. That is, the manoeuvre time is more sensitive to control values early in the manoeuvre. This is also true for the constraints as the distance between the point of control application and the constraint position increases. A wide spread in the magnitude of these sensitivities will ultimately lead to numerical issues in the NLP algorithm [1]. As the manoeuvre length increases, so does the range of sensitivities of the objective function and constraint violations to control variations. In simple terms, small changes in ‘early’ controls are more likely to spin the car. This places a practical limit on the length of the manoeuvre that can be solved by the method so far described.

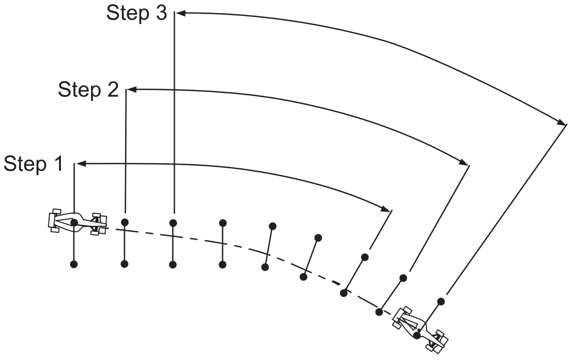


Figure 4. Finite horizon.

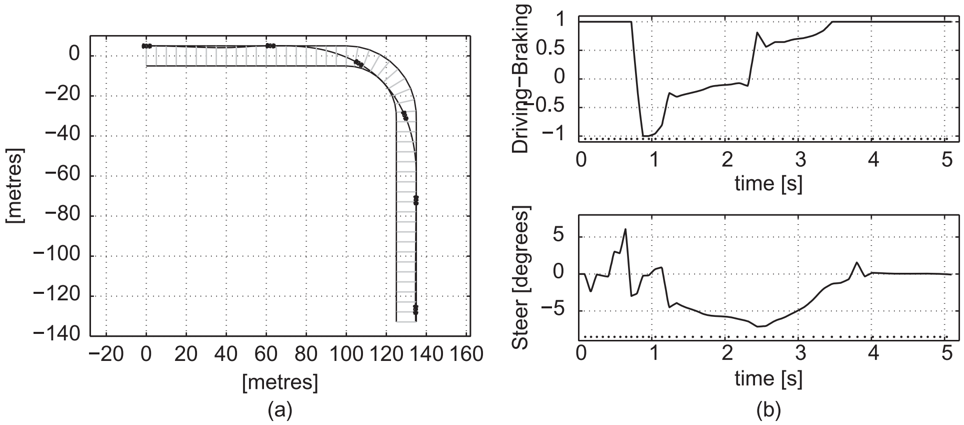
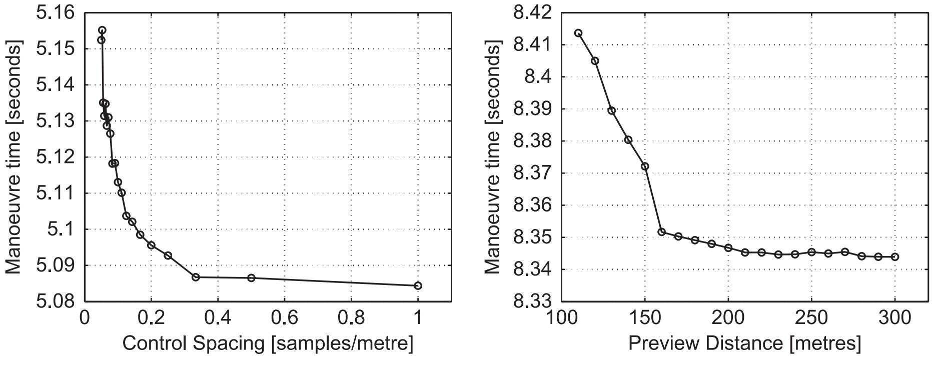


Figure 5. Simple single corner manoeuvre: (a) optimal path and (b) optimal controls.

The method is extended for manoeuvres of arbitrary length using the Finite Horizon approach [8] (Figure 4). A manoeuvre consisting of waypoints $m = 0, 1, \ldots, M(M \gg N)$ is optimised in stages using the short manoeuvre algorithm. At the mth step, the short manoeuvre algorithm is carried out on waypoints m to $m + N$. The control values at point $m + 1$ are then fixed before moving on to the next step. At each step of the algorithm, the controls associated with a single wayline are optimised. The length of the short manoeuvre defines the 'distance to the horizon' or 'preview' used to optimise the control values at each wayline. A sufficiently large value must be chosen for this distance in order to ensure that the time-optimal control is achieved. This is demonstrated in Section 3.1.

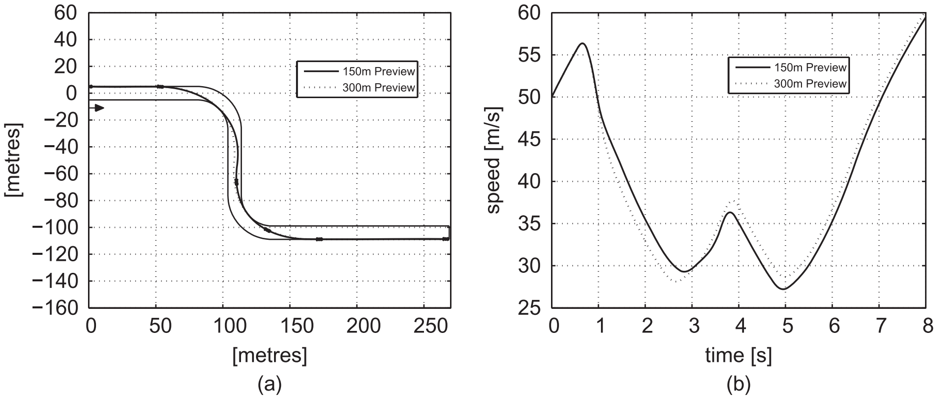
3.1. *Some example results*

The results presented here use the vehicle and tyre model described in Appendix 1. Results using a more complex multibody vehicle model are given in [4]. Figure 5 shows the optimised driven line and control inputs for a single corner using the short manoeuvre algorithm with a uniform control spacing of 10 m. Figure 6(a) shows the influence of control spacing on optimal manoeuvre time. As would be expected, the manoeuvre time decreases as the number of samples/metre increases. There is little gain in manoeuvre time as the sample spacing is reduced from 3 m.



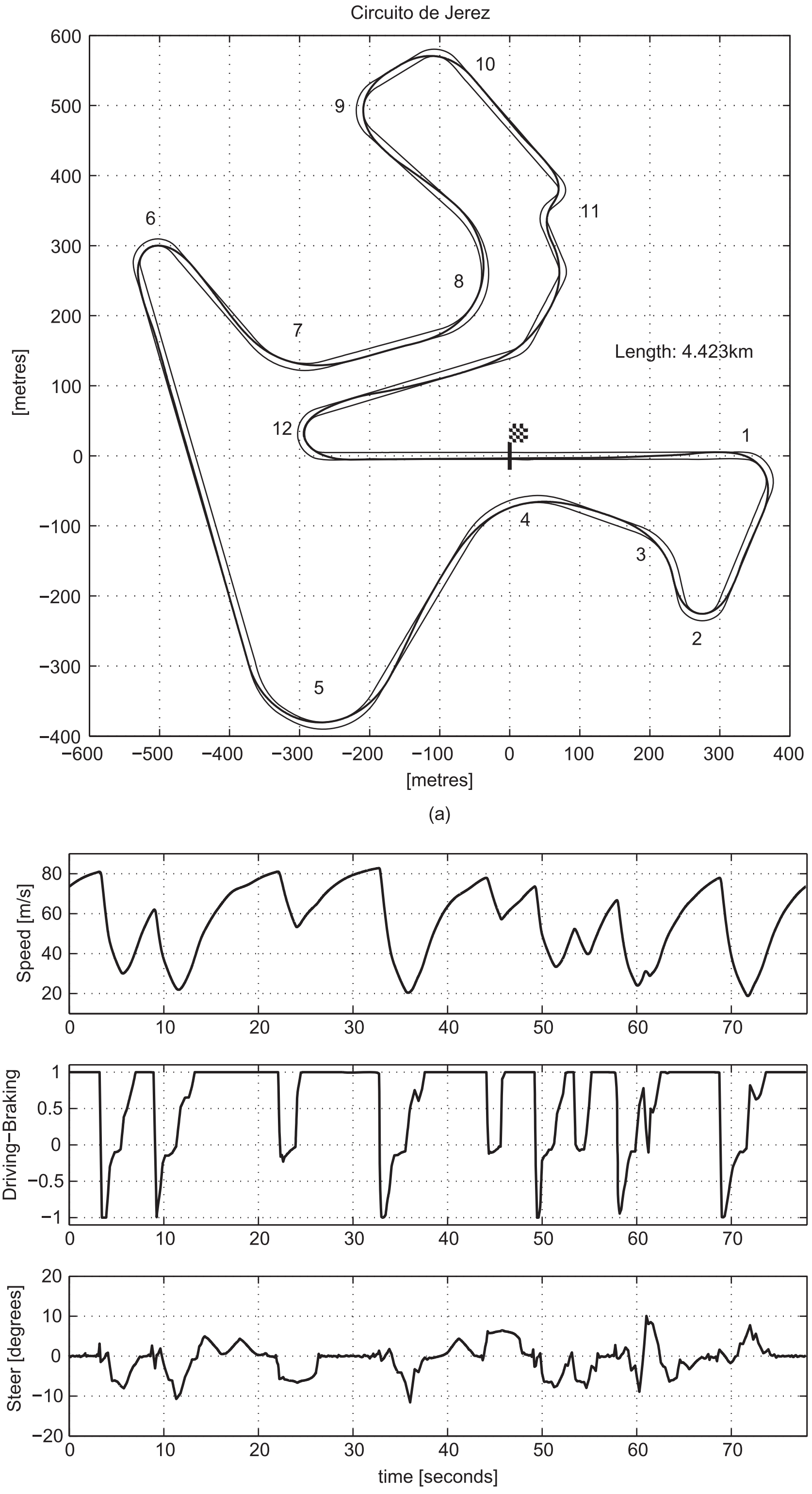
(a) (b)

Figure 6. (a) Influence of the control spacing distance (single corner manoeuvre) and (b) influence of the preview distance (right–left corner combination).



(a) (b)

Figure 7. Examples at 150 and 300 m preview: (a) driven line and (b) speed profile.



(a)

(b)

Figure 8. Jerez circuit: (a) optimised driven line and (b) optimised speed and driver controls.

Figure 7(a) shows two optimised driven lines for a right/left-hand corner combination using the finite horizon method. The solid line corresponds to 150 m preview and the dotted line 300 m preview. Figure 6(b) shows the influence of the preview distance on manoeuvre time. There is little decrease in manoeuvre time as the preview distance is increased from 200 m. The Finite Horizon method allows some insight into the planning of time-optimal trajectories. Figure 7(b) shows the speed profiles for the 150 and 300 m preview cases. The 150 m case retains more speed early in the manoeuvre as it has insufficient view of the corner exit. The 300 m case sacrifices some early speed in order to benefit from increased speed on exit from the final corner. Finally, in Figure 8, the method is applied to the Jerez circuit.

4. Conclusion

It has been demonstrated that time-optimal trajectories can be generated for a representative model of a Formula 1 car using techniques from numerical optimal control. The presented algorithm is robust against variations in the behaviour of the vehicle model. This is achieved by the use of a feasible NLP algorithm and the application of suitable inequality constraints to exclude the most nonlinear (and sub-time-optimal) regions from the vehicle state trajectory. The method requires only limited access to the vehicle model state vector. Hence, to some extent, the vehicle model can be considered as a ‘black box’ allowing the use of arbitrarily complex models. A multibody chassis and thermodynamic tyre model are used in [4]. There is no limit to the manoeuvre length that can be optimised. In [4], the method is used to investigate a wide range of problems in vehicle dynamics such as vehicle stability at the limit, the driven line planning problem, kerbing behaviour and operation with a thermodynamic tyre model. The description of the optimisation method, here, will enable the publication of these further studies.

References

[1] R. Fletcher, *Practical Methods of Optimization*, Wiley-Interscience, New York, 2000.

[2] D. Casanova, *On minimum time Manoeuvring: The theoretical optimal lap*, Ph.D. thesis, Cranfield University, 2000.

[3] R.S. Sharp, D. Casanova, and P. Symonds, *Minimum time manoeuvring: The significance of yaw intertia*, Veh. Syst. Dyn. 34 (2000), pp. 77–115.

[4] D.P. Kelly, *Lap time simulation with transient vehicle and tyre dynamics*, Ph.D. thesis, Cranfield University, 2008.

[5] C. Lawrence, J.L. Zhou, and A.L. Tits, *Users Guide for CFSQP Version 2.5: A C Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality Constraints*, 1997. www.aemdesign.com.

[6] E.R. Panier and A.L. Tits, *On Combining feasibility, descent and superlinear convergence in inequality constrained optimization*, Math. Program. 59 (1993), pp. 261–276.

[7] D. Casanova, R.S. Sharp, M. Final, B. Christianson, and P. Symonds, *Proceedings of Automatic Differentiation 2000: From Simulation to Optimization*, Springer-Verlag, Berlin, 2000.

[8] E. Velenis and P. Tsiotras, *Optimal Velocity Profile for Given Acceleration Limits: Receding Horizon Implementation*, Proceedings of the American Control Conference Vol. 3, IEEE, 2005.

[9] H.B. Pacejka, *Tyre and Vehicle Dynamics*, Butterworth Heinemann, Oxford, 2002.

Appendix 1. Vehicle and tyre model

For brevity of presentation, a simple seven degree-of-freedom planar vehicle model is used with coordinate system shown in Figure A1 and states given in Table A1. Lateral and longitudinal wheel load transfer are computed using steady-state assumptions. A drive-line model including an engine torque map, gear ratio and limited slip differential is used. A simple aerodynamic model is adopted. An empirical Magic-Formula-like tyre model is used [4,9]. Examples of μ against slip curves are given in Figure A2. Computation of the derivatives of the vehicle model state vector is described below.

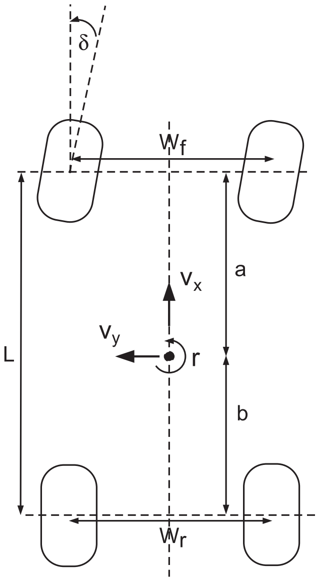


Figure A1. Vehicle model coordinate system.

Table A1. Vehicle model states.

Symbol	Description
v_x	Longitudinal velocity at the vehicle centre of gravity
v_y	Lateral velocity at the vehicle centre of gravity
r	Vehicle yaw rate
ω_{LF}	Left front wheel rotational speed
ω_{RF}	Right front wheel rotational speed
ω_{LR}	Left rear wheel rotational speed
ω_{RR}	Right rear wheel rotational speed

The tyre slip quantities are computed as follows. Small angle assumptions are made. The lateral velocity at the centre of the front and rear axles is given by

$$v_{\mathrm{F}} = v_y + ar, \quad v_{\mathrm{R}} = v_y - br. \tag{A1}$$

Similarly, the longitudinal velocity at the centre of each wheel hub is given by

$$u_{\mathrm{LF}} = v_x - \frac{rW_F}{2}, \quad u_{\mathrm{RF}} = v_x - \frac{rW_{\mathrm{F}}}{2}, \tag{A2}$$

$$u_{\mathrm{LR}} = v_x - \frac{rW_{\mathrm{R}}}{2}, \quad u_{\mathrm{RR}} = v_x + \frac{rW_{\mathrm{R}}}{2}. \tag{A3}$$

Slip angle at each wheel is given by

$$\alpha_{\mathrm{LF}} = \frac{v_{\mathrm{F}}}{u_{\mathrm{LF}}} - \delta, \quad \alpha_{\mathrm{RF}} = \frac{v_{\mathrm{F}}}{u_{\mathrm{RF}}} - \delta, \tag{A4}$$

$$\alpha_{\mathrm{LR}} = \frac{v_{\mathrm{R}}}{u_{\mathrm{LR}}}, \quad \alpha_{\mathrm{RR}} = \frac{v_{\mathrm{R}}}{u_{\mathrm{RR}}}. \tag{A5}$$

Slip ratio at each wheel is given by

$$\kappa_{\mathrm{LF}} = \frac{\omega_{\mathrm{LF}}}{u_{\mathrm{LF}}R_{\mathrm{F}}} - 1, \quad \kappa_{\mathrm{RF}} = \frac{\omega_{\mathrm{RF}}}{u_{\mathrm{RF}}R_{\mathrm{F}}} - 1, \tag{A6}$$

$$\kappa_{\mathrm{LR}} = \frac{\omega_{\mathrm{LR}}}{u_{\mathrm{LR}}R_{\mathrm{R}}} - 1, \quad \kappa_{\mathrm{RR}} = \frac{\omega_{\mathrm{RR}}}{u_{\mathrm{RR}}R_{\mathrm{R}}} - 1. \tag{A7}$$

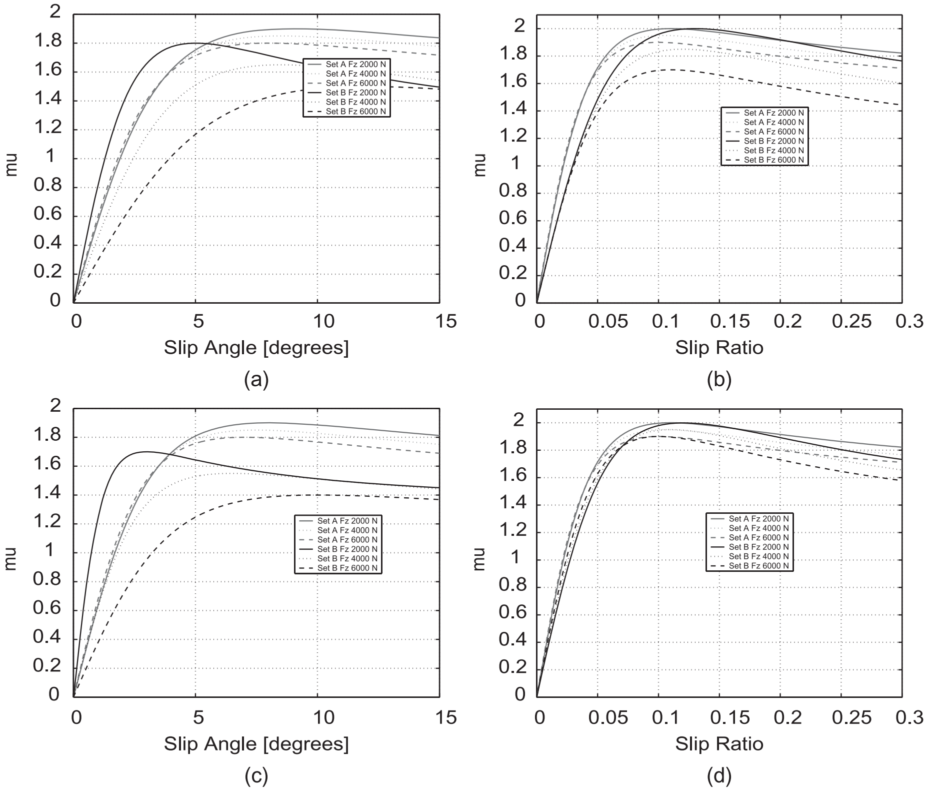


Figure A2. Example tyre μ against slip curves: (a) front lateral (b) front longitudinal, (c) rear lateral and (d) rear longitudinal.

A simple aerodynamic model for drag and downforce based on the square of vehicle velocity is given as follows:

$$F_{\mathrm{df}} = F_{\mathrm{df\,ref}} \left(\frac{v_x}{v_{\mathrm{x\,ref}}} \right)^2, \quad F_{\mathrm{drag}} = F_{\mathrm{drag\,ref}} \left(\frac{v_x}{v_{\mathrm{x\,ref}}} \right)^2, \tag{A8}$$

where $F_{\mathrm{df\,ref}}$ and $F_{\mathrm{drag\,ref}}$ are reference downforce and drag levels, respectively, at the reference speed $v_{\mathrm{x\,ref}}$. The aerodynamic downforce is distributed between the two axles according to the aero balance D_{aero}.

$$F_{\mathrm{df_F}} = D_{\mathrm{aero}} F_{\mathrm{df}}, \quad F_{\mathrm{df_R}} = (1 - D_{\mathrm{aero}}) F_{\mathrm{df}} \tag{A9}$$

The vertical load on each wheel consists of the terms from vehicle mass, aerodynamic downforce, longitudinal and lateral load transfer. Lateral load transfer is distributed between front and rear axles according to the roll balance D_{roll}.

$$F_{\mathrm{z\,LF}} = \frac{F_{\mathrm{df_F}}}{2} + \frac{mgb}{2(a+b)} - \frac{F_{\mathrm{T}} h_{\mathrm{cog}}}{2(a+b)} - D_{\mathrm{roll}} \frac{2m v_x r h_{\mathrm{cog}}}{w_{\mathrm{F}} + w_{\mathrm{R}}}, \tag{A10}$$

$$F_{\mathrm{z\,RF}} = \frac{F_{\mathrm{df_F}}}{2} + \frac{mgb}{2(a+b)} - \frac{F_{\mathrm{T}} h_{\mathrm{cog}}}{2(a+b)} + D_{\mathrm{roll}} \frac{2m v_x r h_{\mathrm{cog}}}{w_{\mathrm{F}} + w_{\mathrm{R}}}, \tag{A11}$$

$$F_{\mathrm{z\,LR}} = \frac{F_{\mathrm{df_R}}}{2} + \frac{mga}{2(a+b)} + \frac{F_{\mathrm{T}} h_{\mathrm{cog}}}{2(a+b)} - (1 - D_{\mathrm{roll}}) \frac{2m v_x r h_{\mathrm{cog}}}{w_{\mathrm{F}} + w_{\mathrm{R}}}, \tag{A12}$$

$$F_{\mathrm{zRR}} = \frac{F_{\mathrm{df_R}}}{2} + \frac{mga}{2(a+b)} + \frac{F_{\mathrm{T}} h_{\mathrm{cog}}}{2(a+b)} + (1 - D_{\mathrm{roll}}) \frac{2m v_x r h_{\mathrm{cog}}}{w_{\mathrm{F}} + w_{\mathrm{R}}}, \tag{A13}$$

where F_{T} is the total longitudinal force acting on the vehicle from the driveline and braking system. Longitudinal ($F_{\mathrm{x\,LF}}, F_{\mathrm{x\,RF}}, F_{\mathrm{x\,LR}}, F_{\mathrm{x\,RR}}$) and lateral ($F_{\mathrm{y\,LF}}, F_{\mathrm{y\,RF}}, F_{\mathrm{y\,LR}}, F_{\mathrm{y\,RR}}$) tyre forces are calculated using the slip quantities and vertical wheel loads.

The torques provided by the engine and the braking system are as follows. Braking torques are applied to the wheels when the driver pedal input is negative. The total maximum available brake torque is given by $T_{\mathrm{brake_{MAX}}}$. When braking

($P < 0$) the braking torque is given by the equation below, otherwise the braking torque is equal to zero:

$$T_{\mathrm{brake}} = -PT_{\mathrm{brake}_{\mathrm{MAX}}} \tag{A14}$$

The brake torque is distributed between front and rear axles according to the brake balance D_{brake}. The torque on each axle is distributed equally between the left and the right.

$$T_{\mathrm{brake}_{\mathrm{F}}} = D_{\mathrm{brake}} T_{\mathrm{brake}}, \quad T_{\mathrm{brake}_{\mathrm{R}}} = (1 - D_{\mathrm{brake}}) T_{\mathrm{brake}}. \tag{A15}$$

The engine model is a simple two-dimensional torque lookup table indexed by engine speed ω_{eng} and throttle position. Linear interpolation is used. The torque map is representative of a Formula 1 engine from the late 1990s. When driving ($P > 0$), the engine torque is given by

$$T_{\mathrm{engine}} = \mathrm{Map}(\omega_{\mathrm{eng}}, P). \tag{A16}$$

When braking ($P \leq 0$), it is given by

$$T_{\mathrm{engine}} = \mathrm{Map}(\omega_{\mathrm{eng}}, 0). \tag{A17}$$

The gearbox is modelled as a simple torque/speed ratio. The gear ratio τ is automatically selected on the basis of vehicle longitudinal speed v_x. The change from one ratio to the next is instantaneous as is typical of the current generation of Formula 1 transmission systems.

$$T_{\mathrm{drive}} = \tau T_{\mathrm{engine}}, \tag{A18}$$

$$\omega_{\mathrm{eng}} = \tau \frac{\omega_{\mathrm{LR}} + \omega_{\mathrm{RR}}}{2}. \tag{A19}$$

The differential is responsible for distributing the gearbox output torque to the rear wheels. A viscous limited slip differential is modelled with separate gains for the drive and overrun cases. If $T_{\mathrm{drive}} > 0$, then

$$\Delta T_d = (K_{d_{\mathrm{PL}}} - K_{d_{\mathrm{D}}} T_{\mathrm{drive}})(\sin(\tan^{-1}(\Delta\omega_{\mathrm{d}}))(1 - \cos(\tan^{-1}(2\Delta\omega_d))), \tag{A20}$$

If $T_{\mathrm{drive}} \leq 0$, then

$$\Delta T_d = (K_{d_{\mathrm{PL}}} + K_{d_{\mathrm{O}}} T_{\mathrm{drive}})(\sin(\tan^{-1}(\Delta\omega_{\mathrm{d}}))(1 - \cos(\tan^{-1}(2\Delta\omega_{\mathrm{d}}))), \tag{A21}$$

$$\Delta\omega_{\mathrm{d}} = \omega_{\mathrm{LR}} - \omega_{\mathrm{RR}}. \tag{A22}$$

Table A2. Vehicle model parameters.

Symbol	Description
m	Total vehicle mass
I_{zz}	Rotational inertia of the vehicle about yaw axis
J_{F}	Rotational inertia of the front wheel/tyre
J_{R}	Rotational inertia of the rear wheel/tyre
J_d	Rotational inertia of the engine and driveline referred to the engine side of the gearbox
R_{F}	Radius of the front wheel/tyre
R_{R}	Radius of the rear wheel/tyre
D_{aero}	Aero balance
D_{brake}	Brake balance
D_{roll}	Roll balance
$F_{\mathrm{df\,ref}}$	Aero downforce at the reference speed
$F_{\mathrm{drag\,ref}}$	Aero drag at the reference speed
$v_{x\,\mathrm{ref}}$	Reference speed
$K_{d_{\mathrm{PL}}}$	Differential ‘preload’ gain
$K_{d_{\mathrm{D}}}$	Differential ‘on drive’ gain
$K_{d_{\mathrm{O}}}$	Differential ‘on overrun’ gain
w_f	Front axle track width
w_r	Rear axle track width
$T_{\mathrm{brake}_{\mathrm{MAX}}}$	Maximum total torque from brakes
g	Gravitational constant

Finally, the vehicle model differential equations are as follows:

$$\dot{v}_x = \frac{F_{x\,\mathrm{LF}} + F_{x\,\mathrm{RF}} + F_{x\,\mathrm{LR}} + F_{x\,\mathrm{RR}} - F_{\mathrm{drag}}}{m}, \tag{A23}$$

$$\dot{v}_y = \frac{F_{y\,\mathrm{LF}} + F_{y\,\mathrm{RF}} + F_{y\,\mathrm{LR}} + F_{y\,\mathrm{RR}}}{m} - v_x r, \tag{A24}$$

$$\dot{r} = \frac{b(F_{y\,\mathrm{LR}} + F_{y\,\mathrm{RR}}) - a(F_{y\,\mathrm{LFs}} + F_{y\,\mathrm{RFs}}) + 0.5W_{\mathrm{R}}(F_{x\,\mathrm{RR}} - F_{x\,\mathrm{LR}}) + 0.5W_{\mathrm{F}}(F_{x\,\mathrm{RF}s} - F_{x\,\mathrm{LF}s})}{I_{zz}}, \tag{A25}$$

$$\dot{\omega}_{\mathrm{LF}} = \frac{T_{\mathrm{brake_{LF}}} - R_{\mathrm{F}} F_{x_{\mathrm{LF}}}}{J_{\mathrm{F}}}, \quad \dot{\omega}_{\mathrm{RF}} = \frac{T_{\mathrm{brake_{RF}}} - R_{\mathrm{F}} F_{x_{\mathrm{RF}}}}{J_{\mathrm{F}}}, \tag{A26}$$

$$\dot{\omega}_{\mathrm{LR}} = \frac{T_{\mathrm{brake_{LR}}} - R_{\mathrm{R}} F_{x_{\mathrm{LR}}} - \Delta T_d + \frac{T_{\mathrm{drive}}}{2}}{J_{\mathrm{R}} + \frac{J_{\mathrm{d}}\tau^2}{2}}, \tag{A27}$$

$$\dot{\omega}_{\mathrm{RR}} = \frac{T_{\mathrm{brake_{RR}}} - R_{\mathrm{R}} F_{x_{\mathrm{RR}}} + \Delta T_d + \frac{T_{\mathrm{drive}}}{2}}{J_{\mathrm{R}} + \frac{J_d\tau^2}{2}}. \tag{A28}$$

The parameters used in these equations are summarised in Table A2.