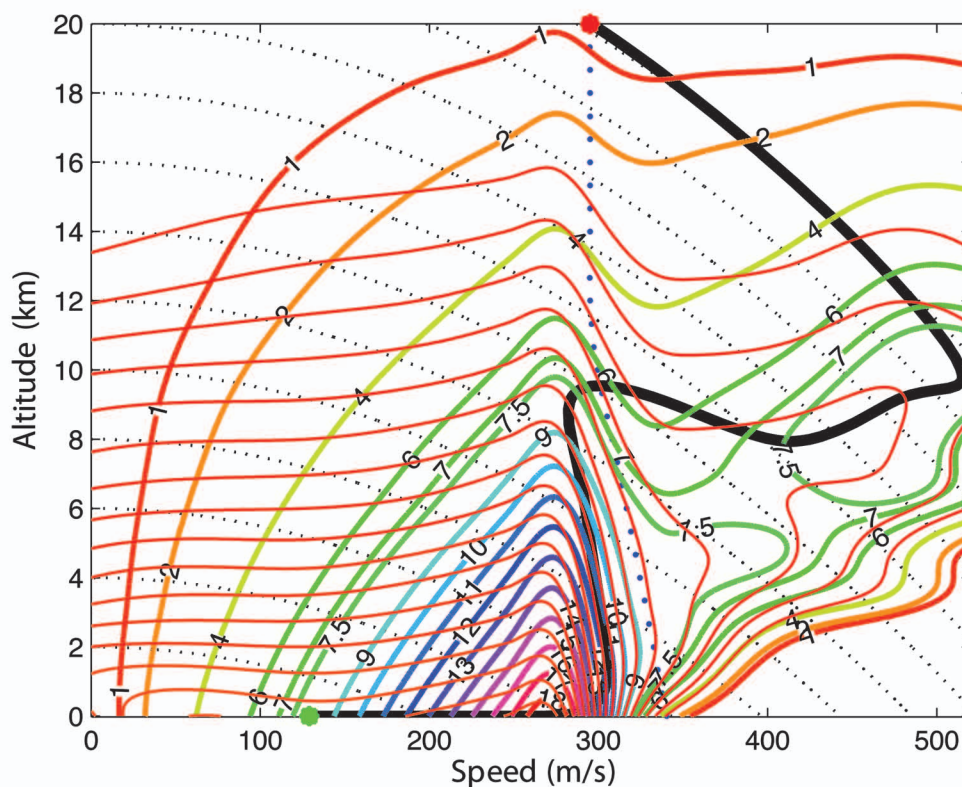


Faster, Higher, and Greener

VEHICULAR OPTIMAL CONTROL

DAVID J.N. LIMEBEER
and ANIL V. RAO



Vehicular optimal control problems have been studied extensively since the early part of the 20th century. Progress in solving these problems has been driven primarily by applications in space and atmospheric flight, including launch vehicles, Earth-based and interplanetary space orbital transfer, and high-performance supersonic aircraft. In all of these applications, the ability to solve increasingly com-

plex optimal control problems has been made possible by advances in high-speed computing. The mathematical and computing techniques being developed are now so diverse, and the range of applications of optimal control so broad, that a comprehensive review of the entire scope of vehicular optimal control is an impossible task. Instead, we will attempt only to provide a flavor of the scope and variety of these problems.

The albatross uses a variant of dynamic soaring, which is a flying strategy that can be optimized by exploiting wind gradients in the atmospheric boundary layer [1]. Dynamic

Optimal control calculations for closed-circuit racing must ensure that the car remains within the track boundaries while its wheels remain in contact with the road.

soaring is exploited in unmanned aerial vehicles (UAVs) and gives them an energy-free loitering and traveling capability that prolongs their endurance [2]. Energy-optimal trajectories for UAVs equipped with solar cells are considered in [3]. These vehicles may be used to travel between given positions within an allowed time or for loitering indefinitely in a target neighborhood. Another problem relates to the optimal level turn of a solar-powered UAV flying in the atmosphere [4]. Readers who do not have optimal control as one of their main interests may wish to study one of the iconic fuel-optimal problems in vehicular optimal control. The famous *Goddard Problem*, posed by Robert H. Goddard (1882–1945) [5], is reviewed in “The Goddard Rocket.”

The minimal fuel thrust programming for the terminal phase of a lunar soft-landing mission is studied in [6]. The optimal thrust program consists of a period of zero thrust (free-fall) followed by full thrust until touchdown, which is a bang-bang strategy. Controlled thrusters and moment gyroscopes are used to maintain and correct the orbit and attitude of the International Space Station. The moment gyroscopes, which are zero-propellant actuators, can be used to re-align the structure. Optimal guidance trajectories can be designed using optimal control [7], which also plays a key role in the design of low-energy, low-thrust transfers to the moon [8]. The result is trajectories that use less propellant than required for standard low-energy transfers. Optimal low-thrust transfers to geosynchronous and Molniya orbits are studied in [9]. Another iconic optimal control problem, which moved to center stage with the introduction of the jet engine, is the *minimum-time to climb problem*. As with many of these problems, it comes in a variety of flavors. The optimal usage of a particular propulsion system to arrive at a predetermined altitude and speed in minimum time is studied in “Minimum Time to Climb.” The solution is not obvious!

Hybrid vehicles are equipped with at least two energy sources and powertrains that combine at least two modes of propulsion. While hybrid powertrains increase the complexity of the drivetrain control problem, they also increase its flexibility and in some sense can outperform anything achievable with a single-source drivetrain [10], [11]. Optimal control is used in [12] to optimize the powertrain energy flow and racing line of a hybrid electric vehicle (HEV) around a closed race track. The utility of optimal control in the solution of minimum-fuel problems in HEVs is demonstrated in [13]. Fuel-optimal control strategies for HEVs are also studied in [14]. The optimal control of hybrid vehicles using direct trans-

cription is studied in [15], where the problem is discretized and the resulting finite-dimensional optimization problem is solved using a nonlinear programming code.

New technologies, as well as political, environmental, and marketing pressures, are driving the rapid development of hybrid vehicles. As a result, pollution thresholds are becoming increasingly more stringent, and hydrocarbon-based fuels are becoming more expensive and less socially acceptable. At the same time, the road-vehicle customer base is demanding the maintenance of customary levels of vehicle performance. More recently, turbo-compounded internal combustion (IC) engines and thermal energy recovery systems (ERSs) have received attention. These systems facilitate the capture of thermal energy from the notoriously inefficient IC engine’s exhaust gases. The introduction of these technologies inevitably brings new technological complications, with the deployment of kinetic and thermal ERSs increasing the complexity of the associated control problem. The optimal control of multiple energy-related resources in real driving conditions, when the future trajectory is unknown, is a difficult problem.

The computation of numerical solutions to optimal control problems in motor racing originated in the 1950s, when simple heuristic arguments were employed to estimate lap times and optimize setup parameters [16]. In a development of the motorcycle work in [17], an indirect method is used to examine minimum-time maneuvers for a single-track car model through a U-turn [18]. An early example of a direct numerical method for time-optimal control being applied to road vehicles is given in [19], which considers a lane-change maneuver for a simple nonlinear single-mass car model, and the optimal control problem was solved using a gradient projection algorithm. Direct multiple shooting is used in a minimum lap-time setting in [20]. This approach was deemed a sensible compromise between the sensitivity problems associated with shooting methods and the much larger nonlinear programs (NLPs) associated with direct collocation techniques [21]. Full-lap optimal control calculations for the Barcelona track using a simple four-wheeled car model required between 28 and 60 h to converge on a Sun Sparc workstation [20]. Some related work is described in [22]. The method reported simultaneously finds the optimal racing line and the controls required to follow this line in minimum time.

The Goddard Rocket

The *Goddard rocket problem* and its variants determine the thrust program that maximizes the altitude achievable by a rocket that expends a fixed amount of propellant [5]. The atmospheric drag plays a significant role in the switching structure of the optimal thrust for this problem. It was shown in 1951 [S1] that, for the low-drag case, a simple bang-bang structure results, where the first subarc is flown at maximum thrust while the second subarc consists of a coast. When atmospheric drag is introduced, the optimal thrust program may, under certain conditions, contain a singular interval that consists of a variable thrust subarc. If a realistic transonic drag model is used (in which the coefficient of drag transitions quickly from a relatively low value in the subsonic region to a higher value in the supersonic region), the optimal thrust program may be bang-singular-bang-singular-bang [S2].

To convey some of the technical detail of this problem, we follow the treatment given in [34]. Specifically, suppose it is desired to determine the thrust program that maximizes the terminal altitude of a one-dimensional sounding rocket subject to the expenditure of a fixed amount of propellant. The equations of motion for this problem are

$$\dot{v} = (T(t) - D(v, h))/m - g, \quad (S1)$$

$$\dot{h} = v, \quad (S2)$$

$$\dot{m} = -T(t)/c, \quad (S3)$$

where h is the altitude, v is the velocity, m is the mass, T is the thrust (the control), $D = D(v, h)$ is the drag force, c is the exhaust velocity of the propellant, and g is the acceleration due to gravity. The thrust is assumed to lie in the interval

$$0 \leq T(t) \leq T_{\max}. \quad (S4)$$

The objective is to maximize $h(t_f)$ such that the final mass is constrained to be $m(t_f) = m_f$.

A singular interval exists if

$$0 = D(v, h) + mg - \frac{v}{c} D(v, h) - v \frac{\partial D(v, h)}{\partial v}. \quad (S5)$$

If $\rho = \rho_0 \exp(-\beta h)$, the drag force has the form $D(v, h) = (\rho_0/2)v^2 C_D \exp(-\beta h)$, in which all the new quantities are constants. In this case, (S5) becomes the nonlinear speed-dependent feedback law

$$T = D + mg + \frac{mg}{1 + 4c/v + 2c^2/v^2} \left[\frac{\beta c^2}{g} \left(1 + \frac{v}{c} \right) - 1 - \frac{2c}{v} \right]. \quad (S6)$$

In the case that (S5) is never satisfied, a bang-bang thrust program results and the vehicle expends all of its propellant at the maximum allowed rate and then coasts to its maximum altitude. Otherwise, upon completion of the maximum thrust subarc, the singular arc characterized by (S5) is encountered and the feedback law (S6) is used to propel the vehicle until its propellant is depleted. On the third subarc the vehicle coasts to its maximum altitude.

The performance of a supersonic ballistic missile, based loosely on the German V-2 illustrated in Figure S1, is shown in Figure S2. The maximum thrust phase propels the vehicle



FIGURE S1 The Peenemünde rocket test site. The V-2 rocket at launch (at left is the mobile maintenance platform) in 1943. The V-2 was an unmanned, guided, ballistic missile. It was guided by a gyroscope-based system that controlled aerodynamic steering tabs on the tail fins. The missile's launch mass was approximately 13,000 kg, with a thrust at launch of approximately 250 kN. The motor typically burned for about 60 s, propelling the missile to approximately four times the speed of sound. (Attribution: Bundesarchiv, Bild 146-1978-Anh.026-01/CC-BY-SA.)

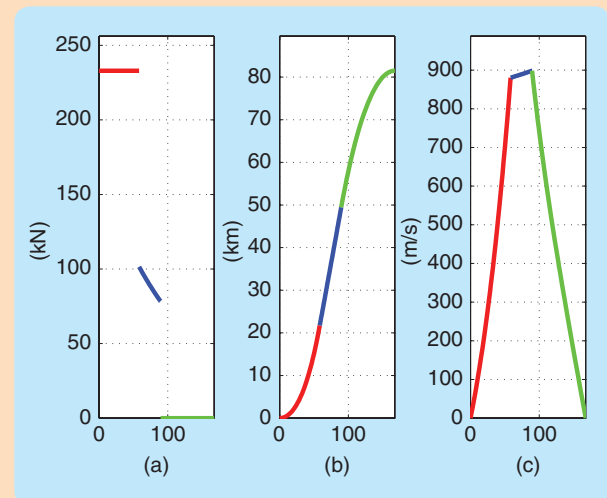


FIGURE S2 An optimized thrust profile for the V-2 ballistic missile. Part (a) shows the classical bang-singular-bang thrust program, (b) shows the missile rising to an apogee of approximately 80 km, while (c) shows the missile's speed reaching a peak during the singular subarc. The full-thrust subarc is in red, the singular subarc is in blue, and the coasting subarc is in green.

to an altitude of almost 20 km at a supersonic speed. The altitude of the vehicle increased to almost 35 km along a reduced thrust singular subarc. The vehicle then coasts to its maximum altitude of over 80 km.

REFERENCES

- [S1] H. S. Tsien and R. C. Evans, "Optimal thrust programming for a sounding rocket," *J. Amer. Rocket Soc.*, vol. 21, no. 3, pp. 99–107, 1951.
- [S2] P. Tsiotras and H. J. Kelley, "Drag-law effects in the Goddard problem," *Automatica*, vol. 27, no. 3, pp. 481–490, 1991.

The apparently unusable computation times associated with high-fidelity optimal control calculations motivated a search for faster optimization methods. One hypothesis was that the computation time could be reduced by supplying the racing line rather than computing it as part of the lap-time optimization problem. Another was the use of quasi-steady-state models, which are now widespread within the Formula One racing community. Both of these ideas were investigated in [23], and, not surprisingly, much shorter computation times were achieved. The reported difference between the predicted lap times using time-optimal and quasi-steady-state analysis for the Barcelona circuit was 2.19 s, which is a significant discrepancy.

Another approach to minimizing the burden associated with minimum-lap-time studies is the use of a linear-quadratic preview to follow a prescribed driving line at a fixed speed [24]. The proposed method makes use of multiple linear models and a gain scheduling scheme, and it is demonstrated on three track segments. Ideas based on linear approximations have been further developed in [25], where (linear time-varying) model predictive control rather than linear preview control is exploited. This approximation allows the problem of finding a racing line and speed profile to be defined as a sequence of convex optimization problems. Prior work by the authors on minimum-lap-time optimal control is reported in [26], where the optimization of the car setup is also considered. The optimal control of hybrid powertrains is considered in [27], with extensions to three-dimensional tracks given in [28] and [29].

To improve the “green” credentials of motor sport, and accelerate road-car-relevant powertrain developments, the Fédération Internationale de l’Automobile (FIA) set out, in 2006, a “green agenda” for Formula One racing, announcing its intention to turn a sport in which cars consume over 150 kg of fuel per race into a catalyst for the development of “green technologies” for road cars. Thermal-electric hybrid powertrains were introduced into Formula One in 2014 [30]. In the lead up to the 2008–2017 Formula One engine development freeze, it was estimated that an average of 4 ms per lap were gained for every million dollars spent on engine development [31]. The hope is that the introduction of hybrid thermal-kinetic ERSs will produce gains that are far more significant and cost-effective than further engine development alone. The 2014 Formula One technical regulations mandate a fuel mass flow limit on all competing engines as well as a total fuel consumption limit of 100 kg per race. The results of recent research show that contemporary lap times can be maintained using less powerful turbo-charged engines, which require only two-thirds of the fuel required by contemporary cars, when thermal-electric ERSs are simultaneously employed [27].

The focus of this article is on optimal control applied to high-performance road vehicles, together with recently developed orthogonal collocation methods for the numerical solution of continuous optimal control problems. These

topics were chosen as focal points for two reasons. First, the solution of optimal control problems in road vehicles has lagged behind solutions to air and space vehicles, and only recently has this gap begun to close in exciting and innovative ways. Second, the recently developed orthogonal collocation methods, together with methods for solving large, sparse NLPs, are only now being widely accepted as a viable solution method for complex, constrained optimal control problems. High-performance road vehicle optimal control problems that were previously challenging are now becoming computationally tractable.

HYBRID POWERTRAIN

From the beginning of the 2014 Formula One season, the race cars will be equipped with hybrid thermal-kinetic ERSs, which will be governed by a complex set of rules. These include fuel consumption limits, fuel mass-flow rate restrictions, energy storage limits, and ERS power constraints. These changes represent the biggest technological upgrade to have occurred over the entire history of the sport.

The *kinetic ERS* (KERS), which is part of the ERS, comprises an electric machine that is coupled directly to the crank shaft of an IC engine. This machine can be used to recover kinetic energy under braking conditions or augment the engine’s drive power when the car is accelerating or traveling at high speed. The KERS motor is called the *motor-generator-unit-kinetic* (MGU-K). Any energy recovered, either thermally or under braking, is stored in the energy store (ES), which is typically a Li-ion battery pack. The 2014 technical regulations also mandate the use of turbo-compounded engines [32]. The pressure-boosting system includes a turbine, a compressor, and a motor-generator on a common drive shaft. This motor-generator is referred to as the *motor-generator-unit-heat* (MGU-H); see Figure 1. The turbine is bypassed by a controllable waste gate. Closing the waste gate increases the power generated by the MGU-H but decreases the power generated by the engine. The key features of the thermal ERS are summarized in Figure 2, which shows a plot of power from the IC engine P_{IC} (under maximum fueling) versus power supplied by the MGU-H P_h .

Figure 2 shows that when the waste gate is fully open, the MGU-H requires 60 kW to operate the compressor and the IC engine output power is boosted by 20 kW. When the waste gate is closed, 40 kW is recovered from the MGU-H and the IC engine output power drops to approximately 440 kW. It is clear that operating the engine with an open waste gate is very inefficient from an energy-usage point of view. In summary, the powertrain controls are the fuel mass-flow rate, the waste gate opening, and the MGU-K demand. It is assumed here that the engine full-speed rotational losses are a constant 40 kW.

The 2014 Formula One energy-recovery rules are more complex than anything seen previously, with the optimal control problem correspondingly intricate. The key control-related

Minimum Time to Climb

Energy-state modeling was one of the most important analysis concepts in aircraft flight performance optimization to come out of World War II [S3]. Traditionally, climb performance was based on an analysis of the aircraft's rate of change of potential energy. Following the introduction of jet engines, the acceleration of jet-powered aircraft at subsonic speeds was so great that the rate of change of the vehicle's kinetic energy could no longer be ignored. When flying at supersonic speeds, the kinetic energy of an aircraft is comparable to its potential energy (relative to the ground). For instance, an aircraft flying at Mach 2, at an altitude of 20 km, has approximately equal kinetic and potential energies. In this case, it becomes necessary to consider the tradeoff between kinetic and potential energy when optimizing supersonic aircraft maneuvering. In a broad sense, the analysis of the tradeoff between kinetic and potential energy is not new because the classical brachistochrone problem is a minimum-time *zoom-dive* problem in which the tradeoff between kinetic and potential energy is optimized [40], [S4].

The study of energy-trading ideas in an aircraft-related context goes back to [S5] that followed the development of the Messerschmitt Me 262 (shown in Figure S3). Kaiser's key observation was that the slow dynamics of the aircraft could be captured in a single state variable, which was called the "gesamthöhe" (the *resultant height*). Denoting h and v as the altitude and speed of the aircraft, respectively, the resultant height is $h_r = h + v^2/2g$ (where g is the acceleration due to gravity). The resultant height was subsequently renamed the *energy height* [S6], then the *specific energy* $e_s = gh + v^2/2$ [S7], and then the *energy state* [S8]. Standard equations of motion for an aircraft modeled as a point mass flying in the vertical plane are

$$\dot{h} = v \sin \gamma, \quad (S7)$$

$$\dot{v} = \hat{T}(h, v) - \hat{D}(h, v) - g \sin \gamma, \quad (S8)$$



FIGURE S3 A Messerschmitt Me 262 in Dübendorf, Switzerland; a jet fighter on an airfield, circa late April/early May 1945. (Attribution: Bundesarchiv, Bild 141-2497/CC-BY-SA.)

$$\dot{\gamma} = \frac{1}{v} (\hat{L}(h, v) - g \cos \gamma), \quad (S9)$$

in which γ is the flight-path angle and $\hat{T}(h, v)$, $\hat{D}(h, v)$, and $\hat{L}(h, v)$ are, respectively, the thrust, drag, and lift specific forces. It then follows by direct calculation that

$$\begin{aligned} \dot{e}_s &= v \dot{v} + g \dot{h} \\ &= v (\hat{T} - \hat{D}), \end{aligned} \quad (S10)$$

in which \dot{e}_s is called the *excess specific power* (that is, the power available to accelerate the aircraft), and $\hat{T} - \hat{D}$ is the *excess specific thrust*. Because the flight-path angle has been eliminated from the excess specific power expression (S10), it is possible to use the specific energy as a single-state model with v as the control. The height is then recoverable from the resultant height

$$h = \frac{(e_s - \frac{v^2}{2})}{g}. \quad (S11)$$

Finally, it is assumed that γ is small and that $\dot{\gamma} \approx 0 \Rightarrow \hat{L}(h, v) = g$. Implicit in the energy-state model is the assumption that kinetic and potential energy can be exchanged instantaneously,

features of the technical regulations [30], which must be translated into control constraints, are as follows:

- 1) The car is restricted to a 100 kg of fuel per race.
- 2) The fuel mass flow rate must not exceed 100 kg/h.
- 3) The car's ES may be a maximum of 4 MJ.
- 4) The ES can accept up to 2 MJ per lap from the MGU-K.
- 5) The MGU-K can draw up to 4 MJ per lap from the ES.
- 6) Power flows to and from the MGU-K are restricted to ± 120 kW.
- 7) The power and energy flow to and from the MGU-H is unrestricted.

The asymmetry in rules 4) and 5) encourages the use and development of thermal energy recovery, which is a relatively untried technology.

We will now explain how these various constraints can be set up in an optimal control setting. The power available at the rear wheels P_m is constrained so that

$$(P_{\max} + P_{W_g} W_g) \dot{E} - P_{\text{loss}} + P_k - P_m \geq 0. \quad (1)$$

The P_{\max} term in (1) represents the power generated by the IC engine under full fueling when the waste gate is closed; we will use $P_{\max} = 480$ kW. The normalized fuel mass flow rate is $\dot{E} \in [0, 1]$. The second term in (1) represents the power boost resulting from opening the waste gate; $W_g \in [0, 1]$ is the waste gate control with $P_{W_g} = 20$ kW. The third term represents the engine's rotational losses and is set at a constant 40 kW, P_k is the power delivered to the MGU-K (P_k is positive when the MGU-K is motoring), and P_m represents the mechanical power delivered to the rear wheels (positive values accelerate the car). The power generated/absorbed by the MGU-H is

$$P_h = ((P_h^{\max} - P_h^{\min}) W_g + P_h^{\min}) \dot{E}, \quad (2)$$

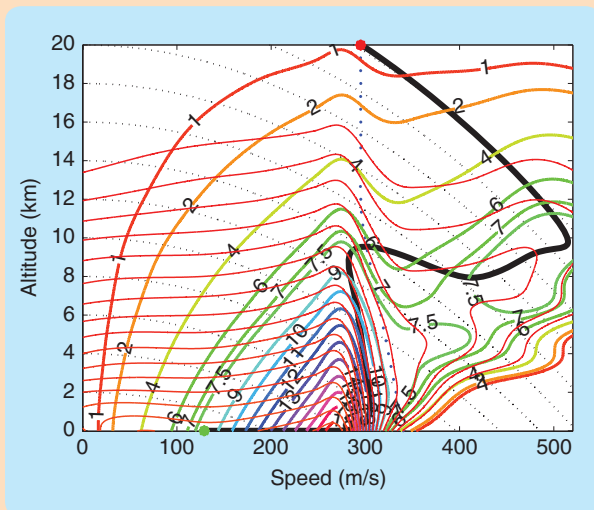


FIGURE S4 Minimum time to a given altitude and speed for Airplane 1 in [S8]. The black curve shows the aircraft's climb profile; the start and end points of the trajectory are green and red stars, respectively. Shown colored and numbered are excess specific power contours. Constant specific energy contours are represented as dotted concentric arcs. The Mach 1 line is shown as (blue) dots. Excess specific thrust contours are shown in red.

with e_s representing the slow dynamics, while changes in h and γ occur quickly (at constant e_s). In reality, kinetic-potential energy trades can be made relatively quickly using zoom-climb and zoom-dive maneuvering.

Figure S4 shows the Kaiser diagram for Airplane 1, as given in [S8], using a four-state, flight-mechanics model. The initial zero-altitude speed of the aircraft is 130 m/s, and it is required to climb to 20 km and Mach 1 in minimum time. In accordance with the principles described, the aircraft gathers speed (and specific energy) through an increase in kinetic energy at zero altitude until it reaches 285 m/s. The reason for this initial ma-

neuver is explained by the excess specific thrust contours (that is, constant $\hat{T}(h, v) - \hat{D}(h, v)$). For this particular engine, the excess thrust is large and almost constant at low altitude up to a speed of approximately Mach 0.8, and so it is best to build up kinetic energy on this subarc. The aircraft then climbs to 9.5 km at an almost constant Mach number. Referring again to the excess specific thrust contours, it is clear that the excess specific thrust maintains at an approximately constant specific thrust until the aircraft reaches approximately 10 km; this subarc is used to further increase the aircraft's specific energy but this time through an increase in potential energy. This climb is followed by a rapid dive to 8 km and Mach 1.32 with a relatively small associated increase in specific energy. This subarc is followed by a speed and specific energy increasing climb back to 10 km and Mach 1.67. At approximately 10 km, the excess thrust contours "flatten out," suggesting that increases in specific energy are now best achieved through kinetic energy increases. The final maneuver is an almost-constant specific energy zoom climb to 20 km and Mach 1, as required.

REFERENCES

- [S3] S. R. Merritt, E. M. Cliff, and H. J. Kelley, "Energy-modelled climb and climb-dash the Kaiser technique," *Automatica*, vol. 21, no. 3, pp. 319–321, 1985.
- [S4] G. Perantoni and D. J. N. Limebeer, "Time-optimal control of rolling bodies," *Int. J. Control*, vol. 86, no. 11, pp. 2006–2021, 2013.
- [S5] F. Kaiser, "The climb of jet-propelled aircraft, part I. speed along the path of optimum climb," Ministry of Supply (Gt. Brit.), RTP/TIB Translation No. Tech. Rep. GDC/15/14ST, Apr. 1944.
- [S6] K. J. Lush, "A review of the problem of choosing a climb technique with proposals for a new climb technique for high performance aircraft," Ministry of Supply (Gt. Brit.), Aeronau. Res. Council Reports and Memoranda, Tech. Rep. R&M 2557, 1951.
- [S7] E. S. Rutowski, "Energy approach to the general aircraft performance problem," *J. Aeronaut. Sci.*, vol. 21, no. 3, pp. 187–195, 1954.
- [S8] A. E. Bryson, N. Desai, and W. C. Hoffman, "Energy-state approximation in performance optimization of supersonic aircraft," *J. Aircraft*, vol. 6, no. 6, pp. 481–488, 1969.

and is a function of the waste gate opening and the fuel mass flow rate; $P_h^{\min} = -40$ kW and $P_h^{\max} = 60$ kW.

To monitor the resource constraints associated with the engine fueling, and the ES and the MGU-K usage, four auxiliary state variables are introduced. The first of these states is used to monitor the fuel consumption and is found by integrating the fuel mass flow rate (as a function of elapsed distance) around the circuit

$$F = \int_{\text{lap}} \dot{F}(s) ds. \quad (3)$$

It is often convenient to use the elapsed distance rather than the elapsed time as the independent variable. Elapsed distance s and time t are related by $ds = v dt$, where v is the vehicle's speed, which is assumed positive. The consumed fuel mass is constrained by $0 \leq F \leq 100/n_l$ kg, in which n_l is the number of laps in the race. As pointed out in regulation 2) above, the

fuel mass flow rate \dot{F} , when normalized, is constrained to the interval $[0, 1]$. The second auxiliary state is described by

$$\dot{E}_s = -(P_h + P_k). \quad (4)$$

This state is used to monitor the energy stored in the ES and is constrained by $4MJ \geq E_s \geq 0$. The third auxiliary state is described by

$$\dot{E}_{\text{ES2K}} = \begin{cases} P_k & P_k > 0, P_h > 0, \\ P_k + P_h & P_k > -P_h > 0, \\ 0 & \text{Otherwise.} \end{cases} \quad (5)$$

This state is used to monitor the energy supplied to the MGU-K from the ES and is constrained by $4MJ \geq E_{\text{ES2K}} \geq 0$. The first alternative corresponds to the case when both electrical machines are operating as motors. In this case, it is only the MGU-K energy usage that is monitored for compliance purposes. The second alternative corresponds to the

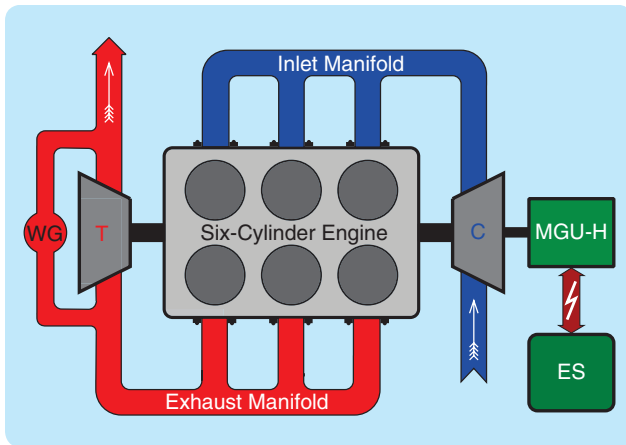


FIGURE 1 A six-cylinder, 1.6-L turbo-compounded Formula One engine with the motor-generator-unit-heat (MGU-H) and energy store (ES). The compressor (C), which boosts the inlet manifold pressure, is driven by the turbine (T) and/or the MGU-H. The compressor, turbine, and MGU-H share a common shaft that is independent of the engine's crank shaft. The waste gate (WG) is used to bypass the turbine when maximum engine power is required. The motor-generator-unit-kinetic is directly coupled to the engine's crank shaft, which is not shown.

case in which the MGU-K is motoring but the MGU-H is generating. In this case, the power generated by the MGU-H is offset against the MGU-K power requirements, with only the difference monitored for compliance purposes.

The fourth auxiliary state is used to monitor energy flows from the MGU-K into the ES and is described by

$$\dot{E}_{K2ES} = \begin{cases} P_k & P_k < 0, P_h < 0, \\ P_k + P_h & 0 > -P_h > P_k, \\ 0 & \text{Otherwise.} \end{cases} \quad (6)$$

This state is constrained by $0 \geq E_{ES2K} \geq -2\text{MJ}$. The first alternative corresponds to the case when both electrical machines are operating as generators. In this case, it is only the MGU-K energy generation that is monitored. The second alternative corresponds to the case in which the MGU-K is generating but the MGU-H is motoring. In this case, the power required

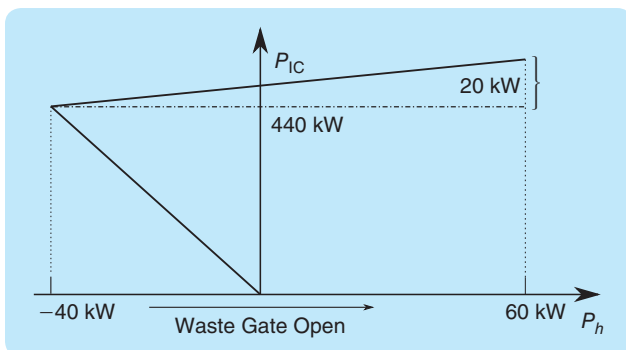


FIGURE 2 The operating regime of the 2014 Formula One engine and thermal energy recovery system at full power. Power absorbed by the motor-generator-unit-heat is deemed positive. P_h is the power supplied by the MGU-H and P_{IC} is the power from the IC engine.

by the MGU-H is offset against the power generated by the MGU-K, with the difference monitored. Differentiable approximations to the Heaviside step function are used to enforce constraints (5) and (6). The inequality constraint (1) and the four auxiliary states (3)–(6) and their associated box constraints are used to model the power and energy constraints required by the 2014 Formula One rules [30].

VEHICLE MODEL

Optimal control calculations for closed-circuit racing must ensure that the car remains within the track boundaries while its wheels remain in contact with the road. To enforce the associated path constraints, it is necessary to model the circuit and monitor the position and orientation of the vehicle on this circuit. As explained in [26], track and car kinematics are modeled using elementary ideas from classical differential geometry. The orientation of the car is described in terms of the angle between its center line and the center line of the track. The lateral position of the vehicle on the track is described by the orthogonal distance between the geometric center of the vehicle and the spine (or center line) of the track. The dynamic model of the car follows from Lagrange's equations and will be based on the geometric quantities shown in Figure 3. The external forces acting on the vehicle come from the tires and the aerodynamic loads; both of these influences will be described briefly. Several more subtle influences must also be modeled relating to tire load transfer effects, the braking system, the differential system, and the powertrains. Each of these modeling components will be described in the remainder of this section. The car parameters are available in [26] and [27]. In the interest of clarity, the track will be assumed to be flat (the interested reader can find a detailed treatment of three-dimensional track modeling in [28], with the associated three-dimensional vehicle dynamics and optimal control dealt with in [29]).

Track Model

The track is modeled using a curvilinear coordinate system that follows the vehicle using the track center-line position as the curvilinear abscissa [17]. Referring to Figure 4, the location

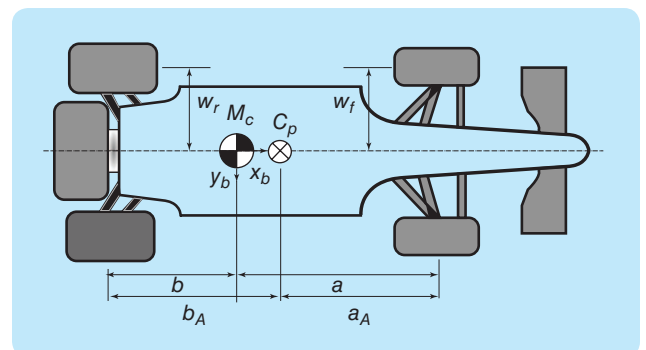


FIGURE 3 A plan view of a Formula One car with its basic geometric parameters. The car mass center is at M_c , while the center of pressure is located at C_p . The body-fixed axes x_b and y_b are in the ground plane.

of the vehicle's mass center is described in terms of the curvilinear abscissa $s(t)$ and the vector $\mathbf{n}(s(t))$. The former quantity defines the distance traveled along the track spine, while the latter gives the position of the vehicle's center of mass in a direction perpendicular to the track spine tangent vector $\mathbf{t}(s(t))$. It is assumed that the traveled distance $s(t)$ is an increasing function of time, and that "time" and "distance" are alternate independent variables. The standard dot notation will be used to signify derivatives with respect to time. At any point s , the track's curvature is given by C (where the s -dependence is implied), and its radius of curvature is given by \mathcal{R} . The track spine tangent vector \mathbf{t} will be described in terms of the track-orientation angle θ , with the track's left- and right-hand half widths given by N_l and N_r , respectively. The yaw angle of the vehicle is given by ψ and the angle between the vehicle and the track by ξ ; $\psi = \theta + \xi$. In this coordinate system, constraints on the track width are easily expressed in terms of constraints on the magnitude of \mathbf{n} . The right- and left-hand half widths need not be equal.

It follows by routine calculation [26] that

$$\dot{s} = \frac{u \cos \xi - v \sin \xi}{1 - nC}, \quad (7)$$

in which u and v are the longitudinal and lateral components of the velocity of the car. This equation describes the way in which the point s on the track spine is "dragged along" by the moving car. The rate of change of n is given by

$$\dot{n} = u \sin \xi + v \cos \xi. \quad (8)$$

Differentiating $\psi = \xi + \theta$ with respect to time results in

$$\dot{\xi} = \dot{\psi} - C\dot{s}. \quad (9)$$

Independent Variable

The distance traveled will be used as the independent variable. This (rather than time) has the advantage of maintaining an explicit connection with the track position, as well as reducing (by one) the number of problem state variables. Suppose

$$dt = \frac{dt}{ds} ds = S_f(s) ds,$$

where S_f comes from (7) as follows:

$$S_f = \left(\frac{ds}{dt} \right)^{-1} = \frac{1 - nC}{u \cos \xi - v \sin \xi}. \quad (10)$$

The quantity S_f is the reciprocal of the component of the vehicle velocity in the track-tangent direction (on the spine at s). It follows from (8) that

$$\frac{dn}{ds} = S_f(u \sin \xi + v \cos \xi) \quad (11)$$

and from (9) that

$$\frac{d\xi}{ds} = S_f \omega - C, \quad (12)$$

where $\omega = \dot{\psi}$ is the vehicle yaw rate.

Car Model

Each tire produces longitudinal and lateral forces that are responsive to the tire's normal loads and slip; the tire forces together with the steer and yaw angle definitions are given in Figure 5.

The tire's longitudinal slip is described by a longitudinal slip coefficient κ , while the lateral slip is described by a slip angle α [33]. Following standard conventions we use

$$\kappa = -\left(1 + \frac{R\omega_w}{u_w}\right), \quad (13)$$

$$\tan \alpha = -\frac{v_w}{u_w}, \quad (14)$$

where R is the wheel radius and ω_w the wheel's spin velocity. The quantities u_w and v_w are the absolute speed components of the wheel center in a wheel-fixed coordinate system. The four tire-slip angles are given by

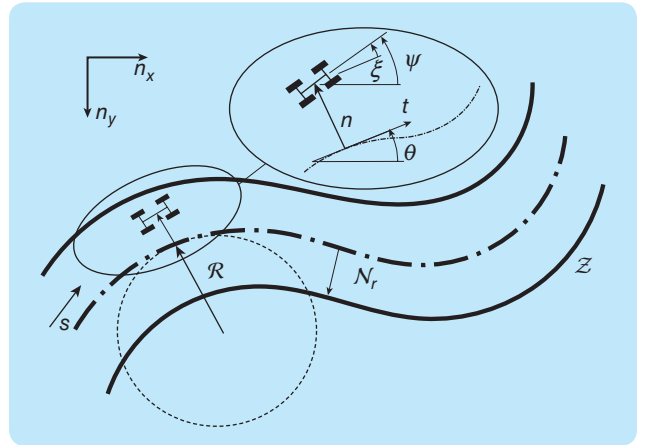


FIGURE 4 A curvilinear-coordinate-based description of a track segment \mathcal{Z} . The independent variable s represents the elapsed center-line distance traveled, with $\mathcal{R}(s)$ as the radius of curvature and $N_r(s)$ as the track right-hand half-width; n_x and n_y are coordinates fixed in an inertial reference frame.

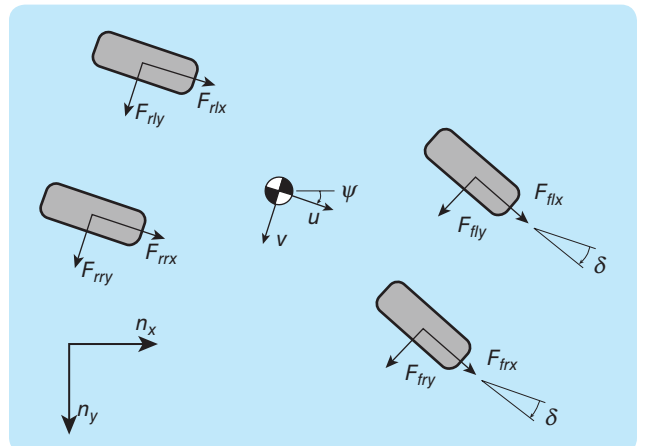


FIGURE 5 The tire force system. The inertial reference frame is shown as n_x and n_y .

$$\begin{aligned}
\alpha_{rr} &= \arctan\left(\frac{v - \dot{\psi}b}{u - \dot{\psi}w_r}\right), \\
\alpha_{rl} &= \arctan\left(\frac{v - \dot{\psi}b}{u + \dot{\psi}w_r}\right), \\
\alpha_{fr} &= \arctan\left(\frac{\sin\delta(\dot{\psi}w_f - u) + \cos\delta(\dot{\psi}a + v)}{\cos\delta(u - \dot{\psi}w_f) + \sin\delta(\dot{\psi}a + v)}\right), \\
\alpha_{fl} &= \arctan\left(\frac{\cos\delta(\dot{\psi}a + v) - \sin\delta(\dot{\psi}w_f + u)}{\cos\delta(\dot{\psi}w_f + u) + \sin\delta(\dot{\psi}a + v)}\right). \quad (15)
\end{aligned}$$

Figure 6(a) shows the load dependence of the longitudinal tire force under pure longitudinal slip conditions; larger longitudinal forces are available when the normal load is increased. The main reason for introducing aerodynamic down-force generating systems on Formula One cars is to exploit this effect. Figure 6(b) shows how the longitudinal force is compromised when the tire is also side slipping. As the side-slip angle increases, the longitudinal peak force reduces and moves toward higher slip values. The normal-load-dependent inverted witch's hat in Figure 7 gives a three-dimensional representation of the combined slip characteristics of the tire model at a fixed normal load of 2000 N. The tire's pure slip characteristics are obtained by taking vertical cuts through the lines $\alpha = 0$ and $\kappa = 0$. A more detailed account of tire modeling can be found in [26].

Balancing forces in the longitudinal and lateral directions, while also balancing the yaw moments, gives

$$\begin{aligned}
M \frac{d}{dt} u(t) &= M\omega v + F_x, \\
M \frac{d}{dt} v(t) &= -M\omega u + F_y, \\
I_z \frac{d}{dt} \omega(t) &= a(\cos\delta(F_{rly} + F_{fly}) + \sin\delta(F_{frx} + F_{flx})) \\
&\quad + w_f(\sin\delta F_{fry} - \cos\delta F_{frx}) - w_r F_{rrx} \\
&\quad + w_f(\cos\delta F_{flx} - \sin\delta F_{fly}) + w_r F_{rlx} - b(F_{rry} + F_{rly}), \quad (16)
\end{aligned}$$

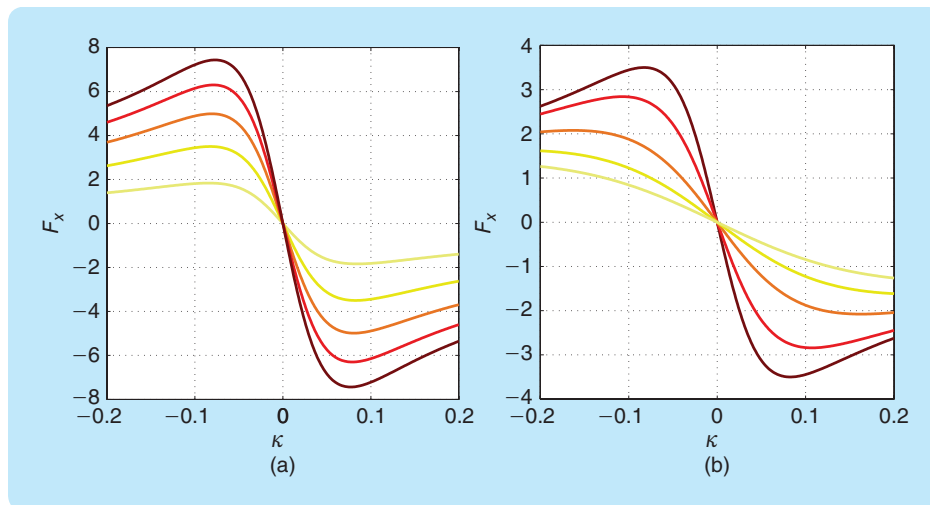


FIGURE 6 Longitudinal tire force versus longitudinal slip; F_x is given in kN. Part (a) considers five normal loads (at $\alpha = 0^\circ$); the lightest curve corresponds to 1000 N with the normal load and then increased in steps of 1000 N. Part (b) considers a combined slip for five values of side-slip angle at a normal load of 2000 N; the darkest curve corresponds to a slip angle of 0° with the slip angle increased in steps of 5° .

in which F_x and F_y are the longitudinal and lateral forces, respectively, acting on the car. These forces are given by

$$F_x = \cos\delta(F_{frx} + F_{flx}) - \sin\delta(F_{fry} + F_{fly}) + (F_{rrx} + F_{rlx}) + F_{ax}, \quad (17)$$

$$F_y = \cos\delta(F_{fry} + F_{fly}) + \sin\delta(F_{frx} + F_{flx}) + (F_{rry} + F_{rly}), \quad (18)$$

in which F_{ax} is the aerodynamic drag force. These equations can be expressed in terms of the independent variables

$$\frac{du}{ds} = S_f(s) \dot{u}, \quad (19)$$

$$\frac{dv}{ds} = S_f(s) \dot{v}, \quad (20)$$

$$\frac{d\omega}{ds} = S_f(s) \dot{\omega}. \quad (21)$$

Aerodynamic Loads

The external forces acting on the car come from the tires and from aerodynamic influences. The aerodynamic force is applied at the center of pressure, which is located in the plane of symmetry of the vehicle. The drag and lift forces are given by

$$F_{ax} = -0.5 C_D \rho A u^2, \quad (22)$$

and

$$F_{az} = 0.5 C_L \rho A u^2, \quad (23)$$

respectively. The negative sign in (22) indicates that the drag acts in the negative body-fixed x -direction. The speed-dependent drag and down-force coefficients and the speed-dependent location of the aerodynamic center of pressure are shown in Figure 8. The interpolating functions for C_D , C_L , and the aerodynamic center of pressure are all quadratic functions of the speed of the vehicle. In more elaborate aerodynamic models, the drag and down force coefficients are functions of the car's ride height, pitch, and yaw and steering angles.

Load Transfer

To compute the time-varying tire loads normal to the ground plane, the forces acting on the car are balanced in the n_z direction and balance moments around the body-fixed x_b - and y_b -axes (see Figure 3). Next, balancing the vertical forces gives

$$\begin{aligned}
0 &= F_{rrz} + F_{rlz} + F_{fz} \\
&\quad + F_{flz} + Mg + F_{az}, \quad (24)
\end{aligned}$$

in which the F_{-z} 's are the vertical tire forces for each of its four

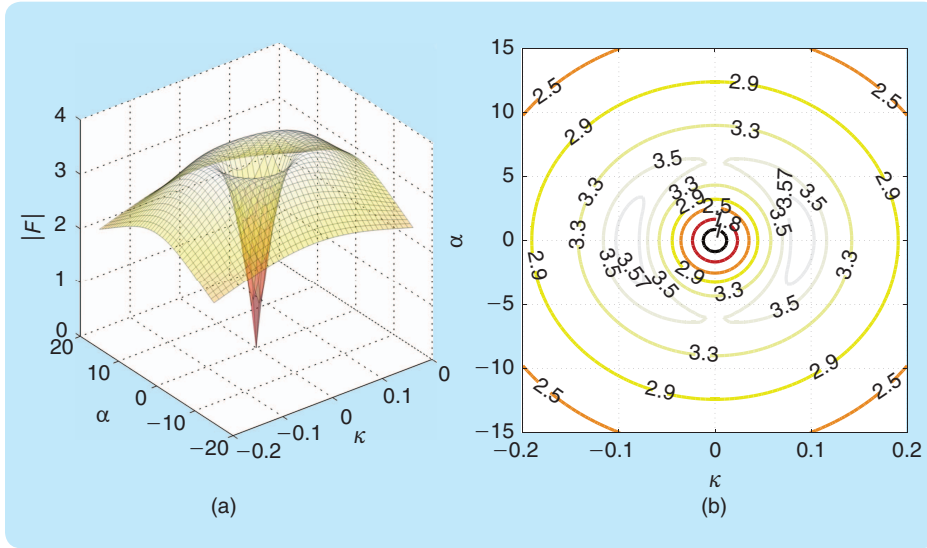


FIGURE 7 A modulus of the tire force as a function of the longitudinal slip κ and the slip angle α (in degrees) for a normal load of 2000 N. Part (b) shows contours of equal tire force.

wheels (the first subscript, r or f , refers to the rear or front of the vehicle, and the second subscript, r or l , refers to the right or left of the vehicle), g is the acceleration due to gravity, and F_{az} is the aerodynamic down force acting on the car. Balancing moments around the car's body-fixed x_b -axis gives

$$0 = w_r(F_{rlz} - F_{rrz}) + w_f(F_{flz} - F_{frz}) + hF_y, \quad (25)$$

in which F_y is the lateral inertial force acting on the car's mass center; see (18). Balancing moments around the car's body-fixed y_b -axis gives

$$0 = b(F_{rrz} + F_{rlz}) - a(F_{frz} + F_{flz}) + hF_x + (a_A - a)F_{az}, \quad (26)$$

where F_x is the longitudinal inertial force acting on the car's mass center [see (17)], while F_{az} is the aerodynamic down force.

Equations (24)–(26) are a set of linear equations in four unknowns. A unique solution for the tire loads can be obtained by adding a suspension-related roll balance relationship, in which the lateral load difference across the front axle is some fraction of the whole

$$F_{frz} - F_{flz} = D_r(F_{frz} + F_{rrz} - F_{flz} - F_{rlz}), \quad (27)$$

where $D_r \in [0, 1]$. Equations (24)–(27) can be solved for the four normal loads

$$\begin{bmatrix} F_{rrz} \\ F_{rlz} \\ F_{frz} \\ F_{flz} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \frac{aM_g + a_A F_{az}}{a+b} & -\frac{(D_r - 1)hF_y}{(1-D_r)w_r + D_r w_f} & -\frac{hF_x}{a+b} \\ \frac{aM_g + a_A F_{az}}{a+b} & \frac{(D_r - 1)hF_y}{(1-D_r)w_r + D_r w_f} & -\frac{hF_x}{a+b} \\ -\frac{bM_g + a_B F_{az}}{a+b} & \frac{D_r hF_y}{(1-D_r)w_r + D_r w_f} & \frac{hF_x}{a+b} \\ -\frac{bM_g + a_B F_{az}}{a+b} & -\frac{D_r hF_y}{(1-D_r)w_r + D_r w_f} & \frac{hF_x}{a+b} \end{bmatrix}. \quad (28)$$

The first column gives the steady-state tire loading due to the vehicle's mass and aerodynamic down force; the second column derives from the lateral load transfer; while the third column is due to the vehicle's acceleration, braking, and drag force and produces the longitudinal load transfer effect.

Brakes

We approximate equal brake caliper pressures with equal braking torques when neither wheel on an axle is locked. If a wheel locks up, the braking torque applied to the locked wheel may be lower than that applied to the rolling wheel.

For the front wheels, this constraint is modeled as

$$0 = \max(\omega_{fr}, 0) \max(\omega_{fl}, 0) (F_{frx} - F_{flx}), \quad (29)$$

in which ω_{fr} and ω_{fl} are the angular velocities of the front-right and front-left wheel, respectively. If either road wheel "locks up," the corresponding angular velocity will be nonpositive and the braking torque constraint (29) becomes inactive.

Differential

The drive torque is delivered to the rear wheels through a limited-slip differential, which is modeled by

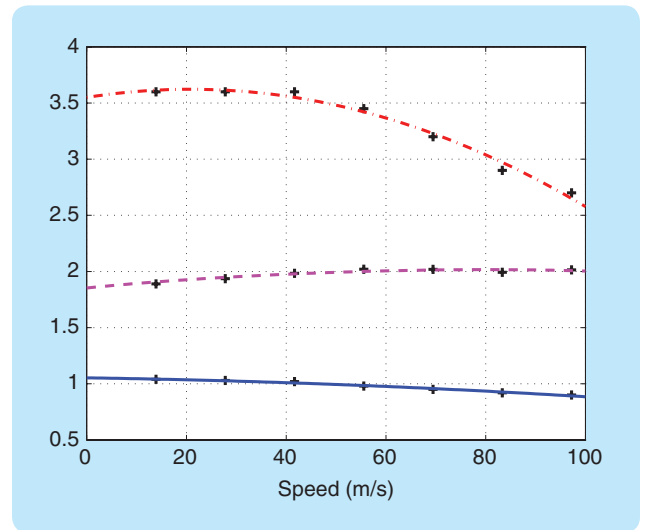


FIGURE 8 Car aerodynamic maps. The drag coefficient C_D is the solid blue curve, which is a function of speed. The down-force coefficient C_L , also a function of speed, is given by the dot-dash red curve. The speed-dependent aerodynamic center of pressure location is given by the dashed magenta curve in meters from the front axle. The + symbols represent measured data points.

$$R(F_{lr} - F_{rr}) = -k_d(\omega_{lr} - \omega_{rr}), \quad (30)$$

in which ω_{lr} and ω_{rr} are the rear-wheel angular velocities, R is the wheel radius, and k_d is a torsional damping coefficient. The special cases of an open- and a locked-differential correspond to $k_d = 0$ and k_d arbitrarily large, respectively. Limited slipping occurs between these extremes.

OPTIMAL CONTROL

Optimal control has been described extensively in a wide array of excellent books including [34]–[39], while *IEEE Control Systems Magazine* articles on optimal control are found in [40] and [41]. In addition, excellent survey articles on numerical methods for optimal control are found in [42] and [43].

Most often, optimal control problems are posed in a general Bolza form [44]. The objective is to determine the state $\mathbf{x}(t) \in \mathbb{R}^n$, the control $\mathbf{u}(t) \in \mathbb{R}^m$, the initial time $t_0 \in \mathbb{R}$, and the terminal time $t_f \in \mathbb{R}$ that minimize the objective functional

$$J = \mathcal{M}(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} \mathcal{L}(\mathbf{x}(t), \mathbf{u}(t), t) dt, \quad (31)$$

subject to the dynamic constraint

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \quad (32)$$

the path constraint

$$\mathbf{c}(\mathbf{x}(t), \mathbf{u}(t), t) \leq \mathbf{0}, \quad (33)$$

and the boundary condition

$$\mathbf{b}(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) = \mathbf{0}, \quad (34)$$

where

$$\begin{aligned} \mathcal{M}: \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} &\rightarrow \mathbb{R}, \\ \mathcal{L}: \mathbb{R}^n \times \mathbb{R}^m &\rightarrow \mathbb{R}, \\ \mathbf{f}: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} &\rightarrow \mathbb{R}^n, \\ \mathbf{c}: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} &\rightarrow \mathbb{R}^p, \\ \mathbf{b}: \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} &\rightarrow \mathbb{R}^q. \end{aligned}$$

The Bolza optimal control problem given in (31)–(34) gives rise to first-order calculus of variations [34], [36]–[38], [44]–[50] conditions

$$\dot{\mathbf{x}} = \frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}}, \quad (35)$$

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial \mathcal{H}}{\partial \mathbf{x}}, \quad (36)$$

$$\mathbf{u}^* = \arg \min_{\mathbf{u} \in \mathcal{U}} \mathcal{H}. \quad (37)$$

The transversality conditions are

$$\boldsymbol{\lambda}(t_0) = -\frac{\partial \mathcal{M}}{\partial \mathbf{x}(t_0)} + \boldsymbol{\nu}^\top \frac{\partial \mathbf{b}}{\partial \mathbf{x}(t_0)},$$

$$\boldsymbol{\lambda}(t_f) = \frac{\partial \mathcal{M}}{\partial \mathbf{x}(t_f)} - \boldsymbol{\nu}^\top \frac{\partial \mathbf{b}}{\partial \mathbf{x}(t_f)}, \quad (38)$$

$$\mathcal{H}(t_0) = \frac{\partial \mathcal{M}}{\partial t_0} - \boldsymbol{\nu}^\top \frac{\partial \mathbf{b}}{\partial t_0}, \mathcal{H}(t_f) = -\frac{\partial \mathcal{M}}{\partial t_f} + \boldsymbol{\nu}^\top \frac{\partial \mathbf{b}}{\partial t_f}, \quad (39)$$

$$\mu_j(t) = 0, \text{ when } c_j(\mathbf{x}, \mathbf{u}, t) < 0, \quad j = 1, \dots, p,$$

$$\mu_j(t) \leq 0, \text{ when } c_j(\mathbf{x}, \mathbf{u}, t) = 0, \quad j = 1, \dots, p, \quad (40)$$

where $\boldsymbol{\lambda}(t) \in \mathbb{R}^n$ is the adjoint or costate, $\boldsymbol{\mu}(t)$ is the path-constraint multiplier, $\mathcal{H}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{u}, t) = \mathcal{L} + \boldsymbol{\lambda}^\top \mathbf{f} - \boldsymbol{\mu}^\top \mathbf{c}$ is the control Hamiltonian, $\boldsymbol{\nu}$ is the Lagrange multiplier associated with the boundary conditions, and \mathcal{U} is the admissible control set. Equations (35) and (36) are called a *Hamiltonian system* because they come from differentiating the (control) Hamiltonian [35], [38], [47]. Equation (37) is known as *Pontryagin's minimum principle* [51] and is a classical result to determine the optimal control. The conditions on the initial and final costate given in (38) are called *transversality conditions* [34]–[38], [48]–[50], [52], while the conditions on the Lagrange multipliers of the path constraints given in (40) are called *complementary slackness conditions* [53]–[55]. The Hamiltonian system, together with its boundary conditions, transversality conditions, and complementary slackness conditions, is called a *Hamiltonian boundary-value problem* (HBVP) [35], [38], [56]. Any solution $(\mathbf{x}(t), \boldsymbol{\lambda}(t), \mathbf{u}(t), \boldsymbol{\mu}(t), \boldsymbol{\nu})$ is called an *extremal solution* and consists of the state, the costate, the controls, any Lagrange multipliers that satisfy the boundary conditions, and any interior-point constraints on the state and costate.

NUMERICAL METHODS USED IN OPTIMAL CONTROL

Most practical optimal control problems must be solved numerically, and these solution techniques are separated along two different lines. The first distinction relates to the actual problem being solved, while the second difference pertains to the type of numerical method used to solve the problem. The problem being solved is either the one that arises from the first-order necessary conditions or the optimal control problem itself. A method that attempts to find a solution to the first-order necessary conditions is called an *indirect method*, while a method that attempts to solve the optimal control problem itself is called a *direct method*. In an indirect approach, a multiple-point BVP is solved with the boundary conditions at the endpoints and/or interior points only partially known. In a direct method, an optimization problem is solved numerically by approximating the original optimal control problem with a finite-dimensional optimization problem. Once the solution approach has been chosen, the problem is solved using either explicit or implicit simulation. In explicit simulation, the dynamics are integrated using a time-marching approach, where the solution at a given time step is obtained from the solution of the dynamics at one or more previous time steps. This process of time marching is then repeated by making progressively better approximations to the unknown boundary conditions until the known boundary conditions are satisfied. In implicit simulation, the solution of the dynamics at each time step along with the boundary conditions is obtained simultaneously without any notion of time marching. Both methods were developed in the context of a general BVP

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), t), \quad \mathbf{b}(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) = \mathbf{0}, \quad (41)$$

where neither the initial conditions $\mathbf{x}(t_0)$ nor the terminal conditions $\mathbf{x}(t_f)$ are, in general, completely known. As a result, (41) must be solved, while seeking simultaneously to determine any missing components of $\mathbf{x}(t_0)$ and/or $\mathbf{x}(t_f)$.

Explicit Simulation (Time-Marching)

In explicit simulation, typically performed by time-marching, the solution of the differential equation at a future time step is obtained using the solution at current and/or previous time steps. The most basic explicit simulation method is the *shooting method* [57]. In shooting, an initial guess is made of the unknown boundary conditions at one end of the interval. Using this guess, together with the known conditions at that endpoint, the differential equation in (41) is integrated across the time interval of interest using a time-marching method (for example, Euler or Runge–Kutta). Upon reaching the opposite end of the time interval $[t_0, t_f]$, the boundary conditions in (41) are evaluated and an assessment is made as to how to update the unknown boundary conditions. The process of integrating the dynamics and updating the unknown conditions is repeated until the boundary conditions on (41) are satisfied to within a specified accuracy tolerance.

The standard shooting method can present numerical difficulties due to instabilities in the dynamics. To overcome this difficulty, the *multiple-shooting method* [58] was developed. In a multiple-shooting method, the time interval $[t_0, t_f]$ is divided into K subintervals $S_k = [t_{k-1}, t_k]$, where $t_K = t_f$ and $\bigcup_{k=1}^K S_k = [t_0, t_f]$. The shooting method is then applied over each subinterval $[t_{k-1}, t_k]$ with the initial value of $\mathbf{x}(t_{k-1})$, $k = 2, \dots, K$ unknown. To enforce continuity, the conditions $\mathbf{x}(t_{\bar{k}}) = \mathbf{x}(t_{\bar{k}}^+)$, $k = 1, \dots, K-1$ must be recognized. These continuity conditions result in a vector-valued root-finding problem, where it is desired to satisfy the boundary conditions in (41) in combination with

$$\mathbf{x}(t_{\bar{k}}) - \mathbf{x}(t_{\bar{k}}^+) = \mathbf{0}, \quad k = 1, \dots, K-1.$$

Despite being of higher dimension, the multiple-shooting method represents an improvement over the standard shooting method because the shorter integration intervals reduce the sensitivity to errors of the unknown terminal conditions.

Implicit Simulation (Collocation)

In implicit simulation, the BVP (41) is again solved by dividing the solution interval $[t_0, t_f]$ into mesh intervals $S_k = [t_{k-1}, t_k]$; $k = 1, \dots, K$, where $t_K = t_f$ and $\bigcup_{k=1}^K S_k = [t_0, t_f]$. In each mesh interval S_k the function $\mathbf{x}(t)$ is approximated in terms of a set of basis functions $\psi_j^{(k)}(t)$, that is,

$$\mathbf{x}^{(k)}(t) \approx \mathbf{X}^{(k)}(t) = \sum_{j=1}^I c_j^{(k)} \psi_j^{(k)}(t). \quad (42)$$

This parameterization is used to integrate (41) from t_{k-1} to one or more stage times $t_{ik} \in S_k$; $i = 1, \dots, I$, where $t_{k-1} < t_{1k} < t_{2k} < \dots < t_{(i-1)k} < t_{ik} = t_k$ for all $i \in [1, \dots, I]$ as follows

$$\mathbf{X}(t_{ik}) = \mathbf{X}(t_{k-1}) + \int_{t_{k-1}}^{t_{ik}} \mathbf{f}(\mathbf{X}(\tau), \tau) d\tau, \quad (i = 1, \dots, I). \quad (43)$$

The integrals given in (43) are then replaced by quadrature approximations of the form

$$\mathbf{X}_{ik} = \mathbf{X}_{k-1} + \sum_{j=1}^I A_{ij}^{(k)} \mathbf{f}(\mathbf{X}_{jk}, t_{jk}), \quad (i = 1, \dots, I), \quad (44)$$

where \mathbf{X}_{jk} ; $j = 1, \dots, I$ are the values of the function approximation $\mathbf{X}(t)$ at the stage points t_{jk} ; $j = 1, \dots, I$ and $A_{ij}^{(k)}$; $i, j = 1, \dots, I$ is the *integration matrix* associated with the particular integration (quadrature) rule that is used to integrate the dynamics in mesh interval S_k . Equation (44) can then be rearranged in the form

$$\mathbf{X}_{ik} - \mathbf{X}_{k-1} - \sum_{j=1}^I A_{ij}^{(k)} \mathbf{f}(\mathbf{X}_{jk}, t_{jk}) = \mathbf{0}, \quad (i = 1, \dots, I), \quad (45)$$

which are called *defect conditions* that must be satisfied at all stage times in each mesh interval. In addition, assuming that the solution is continuous, continuity conditions $\mathbf{x}(t_{\bar{k}}) = \mathbf{x}(t_{\bar{k}}^+)$ must be satisfied at every mesh interval interface. The goal then is to solve for the coefficients $c_j^{(k)}$ in each mesh interval. As opposed to the seemingly similar explicit simulation method and multiple shooting, in implicit simulation all of the defect conditions are found *simultaneously* by solving a large-scale algebraic system of the form $\mathbf{F}(\mathbf{z}) = \mathbf{0}$. Implicit simulation is often referred to as *collocation* because the defect conditions in (45) make the value of the function at each stage point equal to the quadrature approximation of the dynamics at those points.

INDIRECT METHODS

The first-order necessary conditions given in (35)–(40) constitute a two-point BVP. This BVP can be solved using either explicit or implicit simulation as described above. A typical explicit simulation method would be *indirect shooting* or *indirect multiple shooting*, while a typical simultaneous indirect method would be *indirect collocation*. In an indirect-shooting method [57], an initial guess is made for the unknown boundary conditions at one boundary. Using this guess, together with the known initial conditions, the Hamiltonian system in (35) and (36) is integrated to the other boundary (that is, either forward from t_0 to t_f or backward from t_f to t_0) using a time-marching method. Upon reaching t_f , the terminal conditions obtained from the numerical integration are compared to the known terminal conditions given in (34) and (38). If the integrated terminal conditions differ from the known terminal conditions by more than a specified tolerance, the unknown initial conditions are adjusted, and the process is repeated until the difference between the integrated terminal conditions and the required terminal

Uncertainty Principle

A $2n$ -dimensional phase space can be associated with the HBVP given in (35) and (36). If these equations are well behaved, each point in the phase space evolves along a unique path, and we can think of these paths as *flows* in phase space. This time evolution can be described in terms of the *phase flow operator* $\Phi(x, \lambda): (x(t_0), \lambda(t_0)) \mapsto (x(t), \lambda(t))$. Following [S9], we study an important property of these flows.

Consider an infinitesimal volume dV in phase space that evolves over an infinitesimal time dt . We begin in the neighborhood of the point $(x(t_0), \lambda(t_0))$, with volume element

$$dV = \prod_{i=1}^n dx_i \prod_{j=1}^n d\lambda_j. \quad (\text{S12})$$

After an elapsed time dt

$$\hat{x}_i = x_i + \dot{x}_i dt = x_i + \frac{\partial \mathcal{H}}{\partial \lambda_i} dt \quad (\text{S13})$$

and

$$\hat{\lambda}_i = \lambda_i + \dot{\lambda}_i dt = \lambda_i - \frac{\partial \mathcal{H}}{\partial x_i} dt. \quad (\text{S14})$$

The new elemental volume in phase space is

$$d\hat{V} = \prod_{i=1}^n d\hat{x}_i \prod_{j=1}^n d\hat{\lambda}_j = (\det \mathcal{J}) dV, \quad (\text{S15})$$

where \mathcal{J} is the $2n \times 2n$ Jacobian matrix

$$\mathcal{J} = \begin{pmatrix} \frac{\partial \hat{x}}{\partial x} & \frac{\partial \hat{x}}{\partial \lambda} \\ \frac{\partial \hat{\lambda}}{\partial x} & \frac{\partial \hat{\lambda}}{\partial \lambda} \end{pmatrix}. \quad (\text{S16})$$

Substituting (S13) and (S14) into (S16) gives

$$\mathcal{J} = \begin{pmatrix} I_n + \frac{\partial^2 \mathcal{H}}{\partial x \partial \lambda} dt & \frac{\partial^2 \mathcal{H}}{\partial \lambda^2} dt \\ \frac{\partial^2 \mathcal{H}}{\partial x^2} dt & I_n + \frac{\partial^2 \mathcal{H}}{\partial x \partial \lambda} dt \end{pmatrix}. \quad (\text{S17})$$

To compute the determinant of the \mathcal{J} matrix, we use the fact that $\det(I + \epsilon A) = 1 + \epsilon \text{tr}(A) + \mathcal{O}(\epsilon^2)$ for matrix A with ϵ small; $\text{tr}(\cdot)$ denotes the trace. This gives

$$\det(\mathcal{J}) = 1 + \sum_{i=1}^n \left(\frac{\partial^2 \mathcal{H}}{\partial x_i \partial \lambda_i} - \frac{\partial^2 \mathcal{H}}{\partial x_i \partial \lambda_i} \right) dt + \mathcal{O}(\epsilon^2) = 1 + \mathcal{O}(\epsilon^2). \quad (\text{S18})$$

Since $\det(\mathcal{J})/dt = 0$, $\det(\mathcal{J}) = 1$, and so $\Phi(x, \lambda)$ is indeed volume preserving. This property of Hamiltonian maps is known as Liouville's theorem [S9].

As shown in Figure S5, the Hamiltonian phase flow $\Phi(x, \lambda)$ is volume preserving but not shape preserving. Suppose a region of the phase space has its boundaries determined by particles

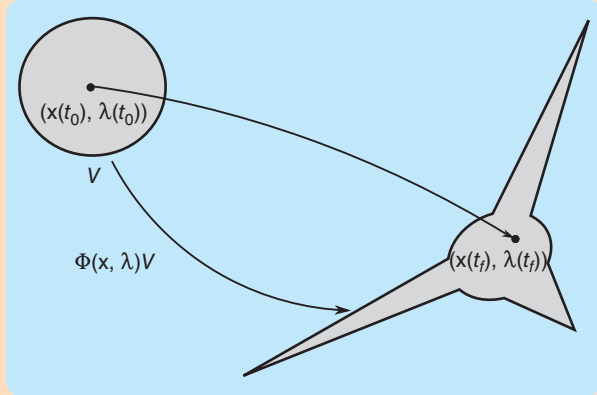


FIGURE S5 The conservation of volume in phase space.

representing initial conditions for the HBVP given in (35) and (36). As the positions of these particles evolve in time, so will the boundaries of the associated control volume. In accordance with Liouville's theorem, any evolution of the control volume in the phase space, with its moving boundaries as specified, will maintain a constant volume. Its shape, however, will, in general, distort, becoming dendritic so as to disperse over different parts of phase space. *It is the formation of these dendrites that causes numerical stability problems in the solution of the HBVP.* If we think of $\Delta p = \prod_{i=1}^n dx_i$ as uncertainty in the system's "position" and $\Delta q = \prod_{j=1}^n d\lambda_j$ as uncertainty in its "momentum," then $V = \Delta p \Delta q$ is reminiscent of Heisenberg's uncertainty principle. Unlike the quantum mechanical setting, the set of starting points for the HBVP is a collection of discrete predetermined points rather than a probability distribution. It is this critical difference between quantum mechanical systems and classical shooting-type schemes that allow us to avoid the consequences of Liouville's theorem if we are careful about our choice of solution techniques. Recognizing that volume must be preserved, in a classical system, we can "push" the starting points (classical particles) closer together, while forcing the spaces between the points to the periphery of the control volume. The obvious way to do this is to use an implicit integration scheme, rather than time matching, to solve the HBVP because this essentially converts the HBVP problem into a large rootsolving problem on an integration mesh that can be solved to high accuracy.

REFERENCE

[S9] V. I. Arnold, *Mathematical Methods of Classical Mechanics*. Berlin Heidelberg, Germany: Springer-Verlag, 1978.

conditions is less than some specified threshold. While the simplicity of simple shooting is appealing, this technique suffers from significant numerical difficulties due to ill-conditioning, which is examined in more detail in "Uncertainty Principle." The shooting method poses particularly poor characteristics when the optimal control problem is *hyper-sensitive* [59]–[63], that is, when time interval of interest is

long in comparison with the time scales of the Hamiltonian system in a neighborhood of the optimal solution. In an indirect collocation method, the state and costate can be parameterized using piecewise polynomials, as shown in (42). The collocation procedure leads to a large-scale root-finding problem $\mathbf{F}(\mathbf{z}) = \mathbf{0}$, where the vector of unknown coefficients \mathbf{z} consists of the coefficients of the piecewise polynomial.

Orthogonal collocation methods have been used to solve optimal control problems since the early 1980s.

This system of nonlinear equations is then solved using an appropriate root-finding technique.

DIRECT METHODS

The most basic direct explicit simulation method for solving optimal control problems is the *direct-shooting method*. The control is parameterized as

$$\mathbf{u}(t) = \mathbf{U}(t) \approx \sum_{i=1}^m a_i \psi_i(t), \quad (46)$$

in which $\psi_i(t)$, ($i = 1, \dots, m$) are preassigned basis functions and a_i , ($i = 1, \dots, m$) are the parameters to be determined by optimization. The dynamics are then satisfied by integrating the differential equations using a time-marching integration method. Similarly, the cost function (31) is determined using a quadrature approximation that is consistent with the numerical integrator used to solve the differential equations. The NLP that arises from direct shooting then minimizes the cost subject to any path and interior-point constraints.

Direct shooting suffers from issues similar to those associated with indirect shooting in that instabilities arise due to the long time interval over which the time-marching method is applied. Thus, in a manner similar to that for indirect methods, an alternate direct explicit simulation method is the *direct multiple-shooting method*. In this case, the time interval $[t_0, t_f]$ is divided into K subintervals. The aforementioned direct-shooting method is then used over each subinterval $[t_{k-1}, t_k]$ with the values of the state at the beginning of each subinterval and the unknown coefficients in the control parameterization being unknowns in the optimization. To enforce continuity, the following conditions are enforced at the interface of each subinterval

$$\mathbf{x}(t_k^-) - \mathbf{x}(t_k^+) = 0, \quad (k = 1, \dots, K - 1). \quad (47)$$

The continuity conditions (47) result in a vector root-finding problem, where it is desired to drive the values of the difference between $\mathbf{x}(t_k^-) - \mathbf{x}(t_k^+)$ to zero. It is seen that the direct multiple-shooting method increases the size of the optimization problem because the values of the state at the beginning of each subinterval are parameters in the optimization. Despite the increased size of the problem due to these extra variables, the direct multiple-shooting method is an improvement over the standard direct-shooting method because the sensitivity to errors in the unknown initial conditions is reduced due to integration performed over significantly smaller time intervals.

DIRECT COLLOCATION METHODS

Over the past two decades, implicit simulation *direct collocation methods* have become the de facto standard for the numerical solution of optimal control problems. A direct collocation method is an implicit simulation approach where both the state and control are parameterized in terms of basis functions (generally piecewise polynomials) for implicit simulation. In the context of optimal control, fixed-order integration methods such as Runge–Kutta and Hermite–Simpson methods [21], [64]–[73] have been employed. In direct collocation methods, the following NLP arises from the approximation of the optimal control problem:

$$\min F(\mathbf{z}) \text{ subject to } \begin{cases} \mathbf{z}_{\min} \leq \mathbf{z} \leq \mathbf{z}_{\max}, \\ \mathbf{g}(\mathbf{z}) \leq \mathbf{0}. \end{cases}$$

As a rule, this NLP is large, containing tens or even hundreds of thousands of variables and constraints. The key feature of these problems, which makes them tractable, is the fact that the NLP is *sparse*. A variety of NLP solvers, such as SNOPT [74], IPOPT, and [75], have been developed specifically for solving large, sparse NLPs. In a typical fixed-order collocation method (such as a Runge–Kutta method), accuracy is improved by increasing the number of mesh intervals. A limitation of a fixed-order method is that, for complex problems, the only way to achieve convergence to the true optimal solution is to increase significantly the size of the mesh.

ORTHOGONAL COLLOCATION METHODS

In recent years, a great deal of research has been done on *orthogonal collocation methods* for solving optimal control problems. In these methods, the optimization horizon is subdivided into several mesh intervals, with the state in each interval approximated by a linear combination of Lagrange polynomials. The collocation is performed at Gaussian quadrature collocation points. A Gaussian quadrature method converges exponentially when used to approximate the integral of a smooth function. In fixed-order algorithms, such as Runge–Kutta and Hermite–Simpson methods, accuracy can only be improved by adjusting the number and width of the mesh intervals. In an orthogonal collocation method, accuracy can also be improved by adjusting the degree of the polynomial approximations. In an orthogonal collocation method, by increasing the approximating polynomial degrees in segments where the solution is smooth, it is possible to achieve convergence to the optimal solution by exploiting the exponential convergence of the Gaussian quadrature. Mesh

The results presented show that the racing performance achieved in 2013 with 150 kg of fuel can now be achieved with 100 kg of fuel when an optimized hybrid powertrain is used.

boundaries can also be added at strategic locations where smoothness in the solution is lost.

To see how a Gaussian quadrature method works in practice, we consider the most-studied orthogonal collocation method that employs collocation at Legendre–Gauss–Radau (LGR) points. In the LGR discretization, the time interval $t \in [t_0, t_f]$ is transformed to the fixed interval $\tau \in [-1, +1]$ with the affine transformation $t = (t_f - t_0)\tau/2 + (t_f + t_0)/2$. The interval $[-1, +1]$ is then partitioned into a *mesh* consisting of K *mesh intervals* $S_k = [T_{k-1}, T_k]$, $k = 1, \dots, K$, where $-1 = T_0 < T_1 < \dots < T_K = +1$. The mesh intervals have the property that $\bigcup_{k=1}^K S_k = [-1, +1]$. Let $\mathbf{x}^{(k)}(\tau)$ and $\mathbf{u}^{(k)}(\tau)$ be the state and control in S_k . The Bolza optimal control problem (31)–(34) can then be rewritten as

minimize the cost functional

$$J = \mathcal{M}(\mathbf{x}^{(1)}(-1), t_0, \mathbf{x}^{(K)}(+1), t_f) + \frac{t_f - t_0}{2} \sum_{k=1}^K \int_{T_{k-1}}^{T_k} \mathcal{L}(\mathbf{x}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), t(\tau, t_0, t_f)) d\tau, \quad (48)$$

subject to the dynamic constraints

$$\frac{d\mathbf{x}^{(k)}(\tau)}{d\tau} = \frac{t_f - t_0}{2} \mathbf{f}(\mathbf{x}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), t(\tau, t_0, t_f)), \quad (k = 1, \dots, K), \quad (49)$$

the path constraints

$$\mathbf{c}_{\min} \leq \mathbf{c}(\mathbf{x}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), t(\tau, t_0, t_f)) \leq \mathbf{c}_{\max}, \quad (k = 1, \dots, K), \quad (50)$$

and the boundary conditions

$$\mathbf{b}(\mathbf{x}^{(1)}(-1), t_0, \mathbf{x}^{(K)}(+1), t_f) = \mathbf{0}. \quad (51)$$

As the state must be continuous at each interior mesh point, it is required that the condition $\mathbf{x}(T_k^-) = \mathbf{x}(T_k^+)$, $(k = 1, \dots, K - 1)$ be satisfied at the interior mesh points (T_1, \dots, T_{K-1}) .

Finally, to pose the LGR collocation method as an integration scheme, (49) is replaced by the equivalent integral

$$\mathbf{x}^{(k)}(\tau) = \mathbf{x}^{(k)}(T_{k-1}) + \frac{t_f - t_0}{2} \int_{T_{k-1}}^{\tau} \mathbf{f}(\mathbf{x}^{(k)}(s), \mathbf{u}^{(k)}(s), t(s, t_0, t_f)) ds. \quad (52)$$

The multiple-interval form of the LGR collocation method [76]–[85] is then stated as follows. First, the state of the continuous-time Bolza optimal control problem is approximated in S_k , $k \in [1, \dots, K]$, as

$$\mathbf{x}^{(k)}(\tau) \approx \mathbf{X}^{(k)}(\tau) = \sum_{j=1}^{N_k+1} X_j^{(k)} \ell_j^{(k)}(\tau), \quad (53)$$

where $\tau \in [-1, +1]$, with

$$\ell_j^{(k)}(\tau) = \prod_{\substack{l=1 \\ l \neq j}}^{N_k+1} \frac{\tau - \tau_l^{(k)}}{\tau_j^{(k)} - \tau_l^{(k)}},$$

a basis of Lagrange polynomials, $(\tau_1^{(k)}, \dots, \tau_{N_k}^{(k)})$ are the LGR [86] collocation points in $S_k = [T_{k-1}, T_k]$, and $\tau_{N_k+1}^{(k)} = T_k$ is a noncollocated point. Differentiating $\mathbf{X}^{(k)}(\tau)$ in (53) with respect to τ gives

$$\frac{d\mathbf{X}^{(k)}(\tau)}{d\tau} = \sum_{j=1}^{N_k+1} \mathbf{X}_j^{(k)} \frac{d\ell_j^{(k)}(\tau)}{d\tau}. \quad (54)$$

Then, using the integral form of the dynamics given in (52) and discretizing the result with an LGR quadrature on the interval $[T_{k-1}, T_k]$ gives

$$\mathbf{X}_{i+1}^{(k)} = \mathbf{X}_1^{(k)} + \frac{t_f - t_0}{2} \sum_{j=1}^{N_k} I_{ij}^{(k)} \mathbf{f}(\mathbf{X}_j^{(k)}, \mathbf{U}_j^{(k)}, t(\tau_j^{(k)}, t_0, t_f)) = \mathbf{0} \quad (i = 1, \dots, N_k). \quad (55)$$

The following NLP then arises from the LGR discretization. Minimize the cost functional

$$J \approx \mathcal{M}(\mathbf{X}_1^{(1)}, t_0, \mathbf{X}_{N_k+1}^{(K)}, t_f) + \sum_{k=1}^K \sum_{j=1}^{N_k} \frac{t_f - t_0}{2} w_j^{(k)} \mathcal{L}(\mathbf{X}_j^{(k)}, \mathbf{U}_j^{(k)}, t(\tau_j^{(k)}, t_0, t_f)), \quad (56)$$

subject to defect constraints

$$\mathbf{X}_{i+1}^{(k)} - \mathbf{X}_1^{(k)} - \frac{t_f - t_0}{2} \sum_{j=1}^{N_k} I_{ij}^{(k)} \mathbf{f}(\mathbf{X}_j^{(k)}, \mathbf{U}_j^{(k)}, t(\tau_j^{(k)}, t_0, t_f)) = \mathbf{0}, \quad (i = 1, \dots, N_k), \quad (57)$$

where $I_{ij}^{(k)}$, $(i = 1, \dots, N_k, j = 1, \dots, N_k, k = 1, \dots, K)$ is the $N_k \times N_k$ LGR *integration matrix* in mesh interval $k \in [1, \dots, K]$. The discretized path constraints

$$\mathbf{c}_{\min} \leq \mathbf{c}(\mathbf{X}_i^{(k)}, \mathbf{U}_i^{(k)}, t(\tau_i^{(k)}, t_0, t_f)) \leq \mathbf{c}_{\max}, \quad (i = 1, \dots, N_k), \quad (58)$$

and the discretized boundary conditions,

$$\mathbf{b}(\mathbf{X}_1^{(1)}, t_0, \mathbf{X}_{N_k+1}^{(K)}, t_f) = \mathbf{0}, \quad (59)$$

must also be recognized. The continuity in the state at the interior mesh points $k \in [1, \dots, K - 1]$ is enforced by the condition

$$\mathbf{X}_{N_k+1}^{(k)} = \mathbf{X}_1^{(k+1)}, \quad (k = 1, \dots, K - 1). \quad (60)$$

Computationally, the constraint (60) is eliminated from the problem by using the *same* variable for both $\mathbf{X}_{N_k+1}^{(k)}$ and $\mathbf{X}_1^{(k+1)}$. The NLP that arises from the LGR collocation method is then to minimize the cost function (56) subject to the algebraic constraints (57)–(59).

MESH REFINEMENT

A key aspect of efficiently generating accurate solutions to optimal control problems using collocation methods is the proper placement of the mesh segment boundaries and the collocation points within each mesh segment. The algorithmic determination of the mesh boundaries and collocation points is called *mesh refinement*. In a traditional fixed-order collocation method (trapezoid, Hermite–Simpson, or Runge–Kutta method), the degree of the polynomial in each interval is fixed and the mesh refinement determines the location of the mesh boundaries. Unlike a traditional fixed-order method, an orthogonal collocation method has the flexibility to change the degree of the approximating polynomial (that is, the order of the method). As seen from the prior discussion of the LGR method, the order of the method is changed by modifying the number of LGR points in any mesh interval. The freedom to change the degree of the approximating polynomial in a mesh interval, combined with the ability to modify the width and location of a mesh interval, are referred to, respectively, as the p and h parts of the mesh refinement.

Fixed-order (h) collocation methods have been used extensively because they are typically implicit simulation forms of fixed-order, time-marching methods, which were computationally efficient in the early days of computing. In the last decade, methods that employ only p collocation (that is, a method that employs a single mesh interval and only varies the degree of the approximating polynomial) showed promise for problems whose solutions are smooth and can be accurately approximated using relatively low-degree (that is, lower than a degree 10 to 15) polynomials. The use of a pure h , or a pure p method has limitations. Achieving a desired accuracy tolerance may require an extremely fine mesh (in the case of an h method), or it may require the use of an unreasonably large degree polynomial approximation (in the case of a p method).

To reduce significantly the size of the finite-dimensional approximation, and thus improve computational efficiency of solving the NLP arising from orthogonal collocation methods, a new class of hp orthogonal collocation mesh refinement methods has been recently developed. As described above, in an hp mesh refinement scheme, the number of mesh intervals, the width of each mesh interval, and the degree of the approximating polynomial within each mesh interval are allowed to vary simultaneously. The hp approach has led to several new algorithms for mesh refinement [80]–[83]. This recent research has shown that an accurate solution to the continuous-time optimal control problem can be obtained in a much more efficient manner than by using a

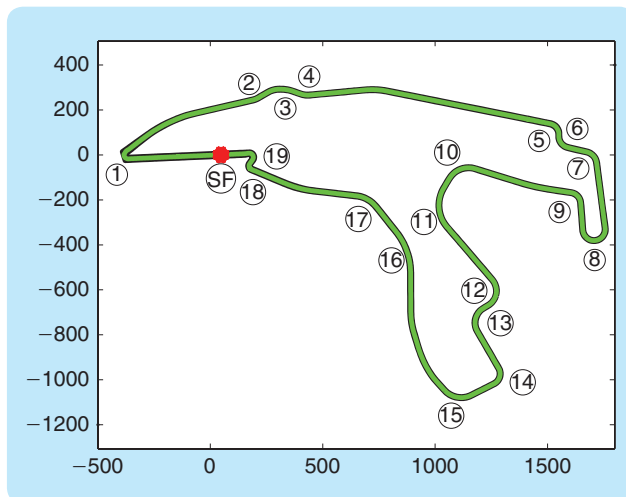


FIGURE 9 The Circuit de Spa-Francorchamps, showing the start-finish line (SF) and its numbered corners.

fixed-order method, while concurrently taking advantage of the key features of a p method in segments where the solution is smooth. Various hp mesh refinement methods have been implemented in the Matlab orthogonal collocation software GPOPS-II [87] that is used here.

OPTIMAL PERFORMANCE

Circuit de Spa-Francorchamps

An optimal control problem can be formulated to identify the track curvature (as a function of the elapsed distance) from noisy GPS measurement data [28]. In effect, this optimal control problem is an optimal parameter identifier for function-valued parameters (the track curvatures). Boundary constraints, which are part of the problem formulation, ensure the closure of the track. Figure 9 illustrates the corner positions on the Circuit de Spa-Francorchamps (Spa) in Belgium with the distances to each corner given in Table 1.

TABLE 1 Distances to mid corner from the start-finish line on the Circuit de Spa-Francorchamps (in meters, ± 25 m).

Corner	Distance (m)	Corner	Distance
1	400	11	4075
2	1100	12	4525
3	1175	13	4650
4	1275	14	4925
5	2400	15	5200
6	2525	16	5875
7	2650	17	6200
8	3075	18	6725
9	3300	19	6800
10	3825	SF	7004

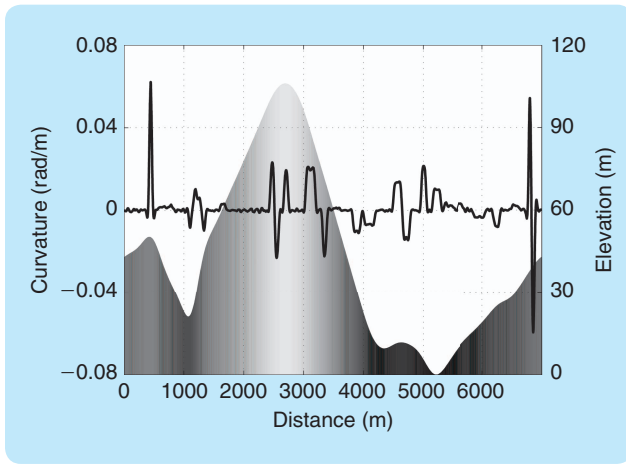


FIGURE 10 The elevation [shaded and right-hand axis (m)] and curvature [solid line and left-hand axis (rad/m)] of Circuit de Spa-Francorchamps.

Figure 10 shows the geodesic curvature and elevation changes for Spa that were estimated using the algorithm described in [28]. The normal curvature and relative torsion are not shown but are “small” in comparison with the geodesic curvature.

Mesh Refinement

The elapsed distance along the track spine (rather than time) is used as the independent variable. This choice of independent variable has the advantage of maintaining a direct link with the track boundary constraints and the position of the car. Figure 11 illustrates the mesh refinement process used when solving the optimal control problem. Figure 11(a) shows the initial mesh that results from dividing the track into 25 equal intervals of approximately 280 m. The boundaries of these mesh intervals are shown as red circles. The four Radau points (per mesh interval) are

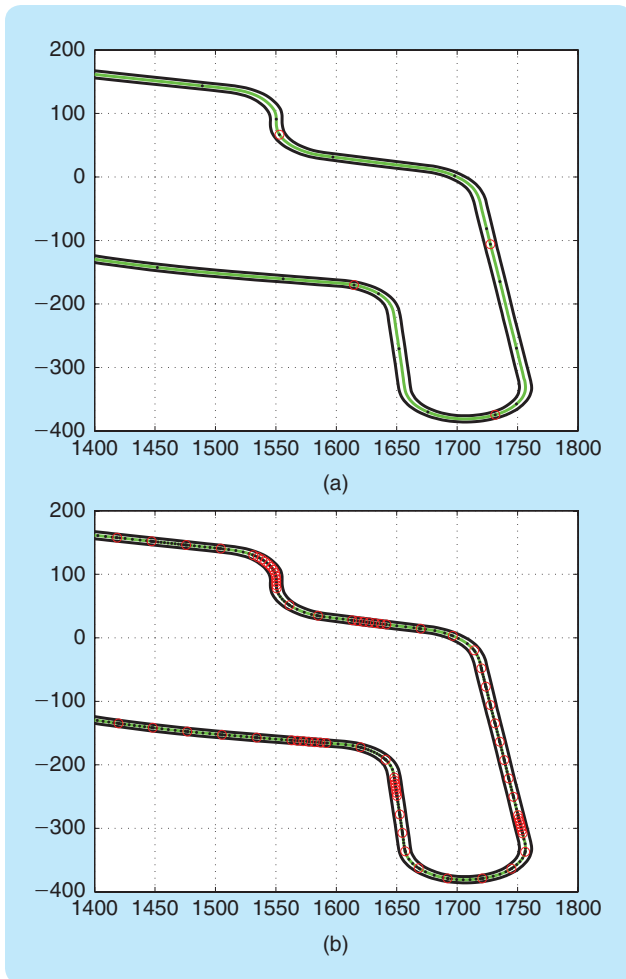


FIGURE 11 Mesh refinement between corners ⑤ and ⑨. Part (a) shows the initial meshing of the track, which is divided into 25 mesh intervals of approximately 280 m using fourth-order interpolating polynomials in each mesh interval. The red circles show the mesh interval boundaries, with the black dots illustrating the collocation points along the track’s spine. Part (b) shows the optimized mesh after 16 mesh refinement iterations.

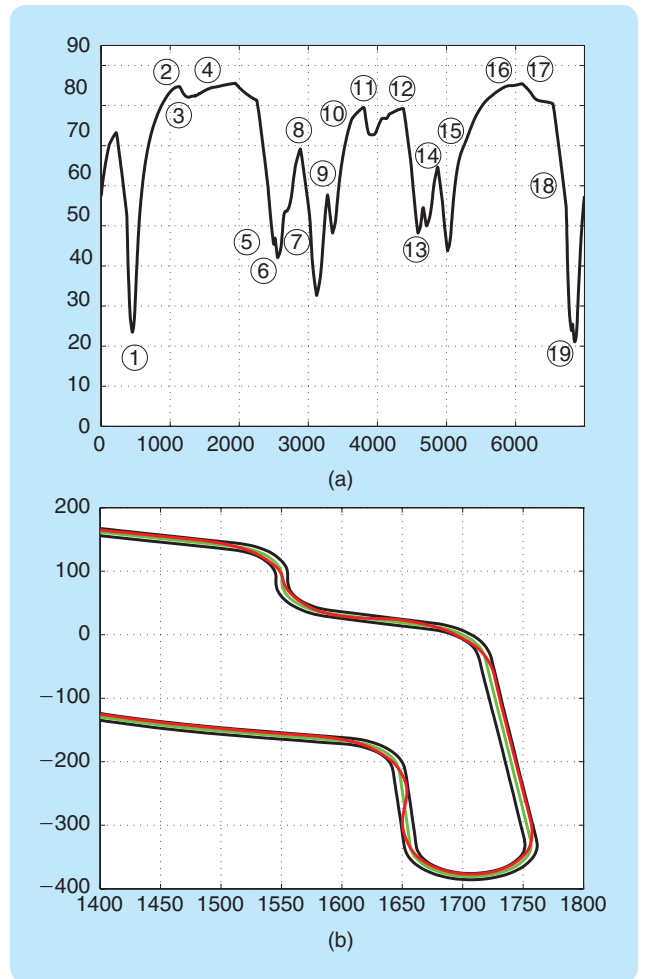


FIGURE 12 Optimized racing performance. Part (a) shows the car’s speed for a racing lap of Spa with the approximate corner positions. Part (b) shows the vehicle’s optimized racing line; the car is traveling clockwise.

shown as black dots. Figure 11(b) shows the refined mesh after 16 mesh-refinement steps. After refinement, the mesh intervals are of unequal length as are the orders of the interpolating polynomials.

Optimal Racing

Figure 12(a) shows the speed profile and racing line of the optimally controlled car. The computed lap time was 1 min, 46.96 s, which compares favorably with the record time of 1 min, 47.26 s, which was set in 2009 in a car with a significantly more powerful engine. This preliminary result shows that the hybrid powertrain with its associated fuel usage restrictions is not compromising the car's performance. Also marked on this figure are the approximate locations of the track's 19 corners.

Figure 12(b) shows the vehicle's racing line. To maintain an optimal racing line, a race car driver will usually take up a position toward the outside edge of a corner before entering it and then turn in following a path through the apex. After "hitting the apex" and upon exiting the corner, the driver will allow the vehicle to return to the outer edge of the track, thus allowing the car to follow a wide radius of curvature path through the bend. The racing line affects the speed into and out of every corner and onto the following straight. A late apex is used to maximize the acceleration onto the following straight. In contrast, an early apex maximizes the use of speed from the incoming straight. This technique is used when approaching a corner, for which the following straight is significantly shorter than that prior to the corner. Hitting the apex mid-corner is a good way to take a corner, while also maximizing the mid-corner minimum speed. Although this is a perennial discussion point for competition drivers, it is "just physics," and an optimal control algorithm knows how to best exploit the track's layout.

Solution times are influenced by many things, including initial conditions, scaling strategies, approximation methods for nonsmooth functions, mesh refinement strategies, gradient calculation techniques, and error tolerances. In our case, these calculations were performed in approximately 20 min each on a standard tower computer based on a 3.5-GHz Intel Core i7 processor using algorithmic derivatives and the NLP solver IPOPT.

Powertrain Usage

Figure 13 shows the optimal utilization of the vehicle's energy resources. The normalized speed (speed/100) is shown (in cyan) as a convenient reference. The first point to note (shown in green) is that the car utilizes fully its fuel quota of 2.273 kg (Spa is a 44-lap race). The MGU-K (shown in blue) generates its maximum quota of 2 MJ per lap, while the maximum quota of 4 MJ that can be transferred from the ES to the MGU-K (shown in magenta) is not fully exploited. This asymmetry in the MGU-K's motoring and generating quotas was introduced to force the car to generate energy with the MGU-H. The state of charge of the

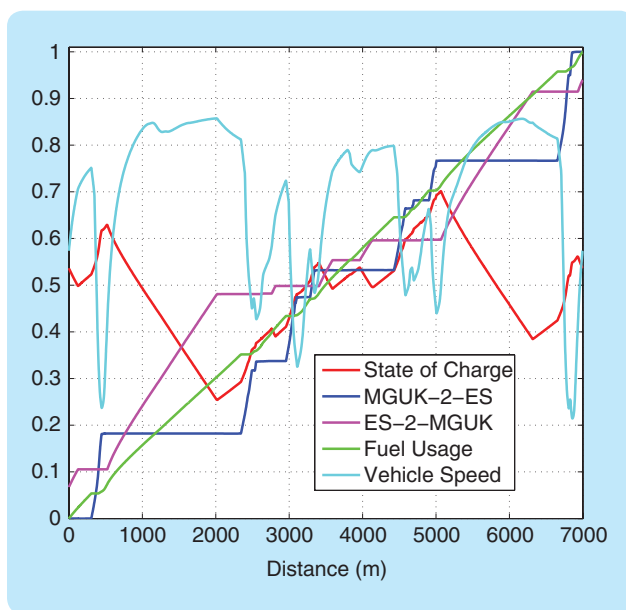


FIGURE 13 Energy usage in the 2014 energy recovery systems under racing conditions. The cyan line is the normalized vehicle speed; the red line is the normalized stored energy; the green line is the normalized fuel usage; the blue line is the normalized energy transfer from the motor-generator-unit-kinetic (MGU-K) to the energy store (ES); and the magenta line is the normalized energy transfer from the ES to the MGU-K.

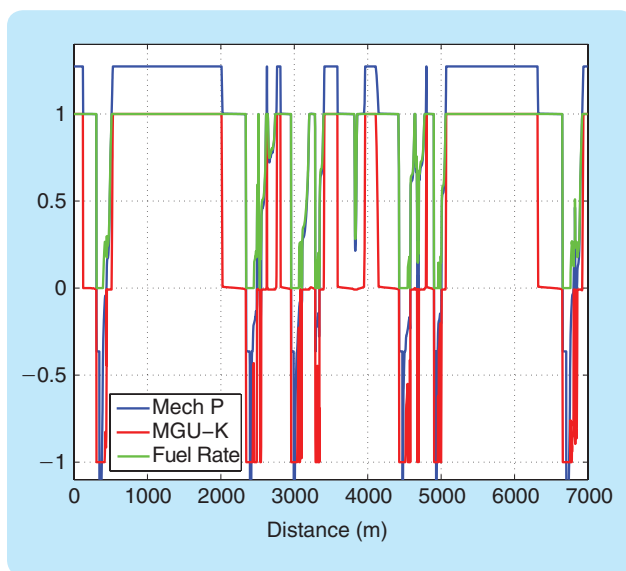


FIGURE 14 Power usage in the 2014 energy recovery systems under racing conditions. The blue line is the normalized mechanical power delivered to the rear wheels; the green line is normalized fuel mass flow rate; the red line is the normalized motor-generator-unit-kinetic (MGU-K) power.

energy storage (shown in red) is constrained to be the same at the start and end of the lap so that the optimal lap can, in principle, be repeated if tire degradation and fuel usage are ignored. Interestingly, the energy storage capability of the

The 2014 Formula One energy-recovery rules are more complex than anything seen previously, with the optimal control problem correspondingly intricate.

ES is only partially utilized. If the plots are examined in detail, it is possible to see the MGU-K driving the car on the fast sections of the track, while it generates under braking on entry into most of the corners. It is not possible to identify easily the contributions being made by the MGU-H, but they will be reflected in discrepancies between the ES's state of charge and MGU-K usage.

Figure 14 shows the power transfers occurring within the powertrain. The green trace shows the fuel usage that operates in a bang-bang manner between its maximum and minimum values. Bang-bang type control behavior is to be expected because the optimal control problem's performance index seeks to minimize the elapsed time and is not a function of the controls. In addition, the controls enter the system dynamics in a linear fashion. If the performance index and the system dynamics are linear in the controls, bang-bang control strategies are likely to result. The bang-bang principle follows immediately from the PMP [35, p. 246–247]. The red trace shows the bang-bang-type operation of the MGU-K, which operates as a generator on the entry to corners, and as a motor on the exit from corners and on fast sections of the track. The car behavior at individual corners can be identified by referring to the distance markers for the Spa circuit given in Table 1. The blue trace, which is the power delivered to the rear wheels, is also bang-bang in character, with a total power delivery of 560 kW possible. Due to the energy efficiencies associated with its use, the waste gate is closed throughout an optimal lap and so is not shown. Under racing conditions, the waste gate is predominantly a safety device that can be used to prevent compressor over speeding under fault conditions.

During heavy braking, the mechanical power delivered to the rear wheels is negative, the MGU-K will be generating, and the fuel flow rate will be cut to zero. As shown in Figure 13, this strategy allows the ES to be recharged. During high-speed driving, such as the section between 400 and 2000 m, both the engine and the MGU-K are used at full capacity, which drains the ES. On the section between 2000 and 2200 m, the MGU-K is unused and the ES is recharged from the MGU-H.

CONCLUSION

Orthogonal collocation methods have been used to solve optimal control problems since the early 1980s. These methods can now solve complex, high-dimension problems by combining the economy of Gaussian quadratures with state-of-the-art, sparse, large-scale, NLP algorithms such as IPOPT,

SNOPT and KNITRO. While the majority of these applications have thus far related to aerospace problems, we have demonstrated that they are also well suited to ground vehicular optimal control problems that may include hybrid powertrains, ERSs, and complex operating constraints.

The results in this article focus on the control of the thermal-kinetic ERSs that have recently been introduced into Formula One racing. In this application, the emphasis is on maintaining established levels of racing performance while simultaneously significantly reducing fuel consumption. The results presented show that the racing performance achieved in 2013 with 150 kg of fuel can now be achieved with 100 kg of fuel when an optimized hybrid powertrain is used. If a modest performance degradation can be accommodated, the fuel consumption can be reduced even further. In racing applications, the powertrain control is essentially a classical bang-bang strategy applied to both the fueling and MGU-K power delivery. A strategy reminiscent of bang-bang control is also employed during braking, although regenerative braking will be limited by the vehicular kinetic energy available, the capacity of the tires, and the restrictions placed on MGU-K usage. The potential for redeploying these technologies in everyday commercial road cars is self-evident.

ACKNOWLEDGMENT

The work was supported by the U.K. Engineering and Physical Sciences Research Council and by the U.S. National Science Foundation.

AUTHOR INFORMATION

David J.N. Limebeer (david.limebeer@eng.ox.ac.uk) received the B.Sc. (Eng.) degree from the University of the Witwatersrand in 1974, the M.Sc. (Eng.) and Ph.D. degrees from the University of Natal in 1977 and 1980, respectively, and the D.Sc. (Eng.) from the University of London in 1992. After completing his Ph.D., he was a post-doc researcher at the University of Cambridge between 1980 and 1984. He then joined the Electrical and Electronic Engineering Department at Imperial College as a lecturer. He was promoted to reader in 1989, professor in 1993, head of the control group in 1996, and head of the department in 1999–2009. In 2009 he moved to Oxford as professor of control engineering and professorial fellow at New College Oxford. His research interests include a range of applied and theoretical problems in control systems and engineering dynamics including robust control, optimal control, process

control, and the control of aeroelastic phenomena in large structures. He is also interested in a variety of problems in multibody mechanics and the dynamics of two- and four-wheeled road vehicles. He is a Fellow of the IEEE (1992), the IET (1994), the Royal Academy of Engineering (1997), and the City and Guilds of London Institute (2002). His consultancy activities have been in a variety of vehicle-related matters; product liability litigation; and patent disputes in optical recording systems, drilling equipment, and high-speed packing machines. He can be contacted at the Department of Engineering Science, University of Oxford, Parks Road, Oxford, OX1 3PJ, United Kingdom.

Anil V. Rao received the A.B. degree in mathematics and the B.S. degree in mechanical engineering (with distinction) from Cornell University in 1988, the M.S.E. in aerospace engineering from the University of Michigan in 1989, and M.A. and Ph.D. degrees from Princeton University in 1992 and 1996, respectively. After completing his Ph.D. degree, he was a senior member of the technical staff at the Aerospace Corporation in Los Angeles, California, from 1996 to 2000. Subsequently, he was a senior member of the technical staff at the Charles Stark Draper Laboratory in Cambridge, Massachusetts. In 2006 he joined the faculty in the Department of Mechanical and Aerospace Engineering at the University of Florida, where he was an assistant professor from 2006 to 2012 and was promoted to associate professor in 2012. The focus of his research program is on computational methods for complex constrained optimal control problems with a primary application focus on aerospace space and atmospheric flight dynamics and secondary application focus on ground vehicle dynamics, underwater dynamics, and biomechanics. In addition to his research, he has won numerous teaching awards throughout his career including Teacher of the Year in the Department of Mechanical Engineering at Boston University (2002 and 2006), College of Engineering Professor of the Year at Boston University (2004), and Teacher of the Year in the Department of Mechanical and Aerospace Engineering at the University of Florida. In 2012 he was awarded the Pramod P. Kharagonekar Junior Faculty award at the University of Florida. He is an associate fellow of the American Institute of Aeronautics and Astronautics.

REFERENCES

[1] J. A. Wilson, "Sweeping flight and soaring albatrosses," *Nature*, vol. 257, pp. 307–308, Sept. 1975.
 [2] M. Deittert, A. Richards, C. A. Toomer, and A. Pipe, "Engineless unmanned aerial vehicle propulsion by dynamic soaring," *J. Guid. Control Dyn.*, vol. 32, no. 5, pp. 1446–1457, 2009.
 [3] A. T. Klesh and P. T. Kabamba, "Solar-powered aircraft: Energy-optimal path planning and perpetual endurance," *J. Guid. Control Dyn.*, vol. 32, no. 4, pp. 1320–1328, 2009.
 [4] D.-M. Ma, J.-K. Shiau, Y.-J. Su, and Y.-H. Chen, "Optimal level turn of solar-powered unmanned aerial vehicle flying in atmosphere," *J. Guid. Control Dyn.*, vol. 33, no. 5, pp. 1347–1356, 2010.
 [5] R. H. Goddard, *A Method of Reaching Extreme Altitudes*, vol. 71. Washington, D.C.: Smithsonian Miscellaneous Collection, 1919, no. 2, p. 54.
 [6] J. S. Meditch, "On the problem of optimal thrust programming for a lunar soft landing," *IEEE Trans. Autom. Contr.*, vol. 9, no. 4, pp. 477–484, 1964.

[7] N. S. Bedrossian, S. Bhatt, W. Kang, and I. M. Ross, "Zero-propellant maneuver guidance," *IEEE Control Syst. Mag.*, vol. 29, no. 5, pp. 53–73, 2009.
 [8] G. Mingotti, F. Toppoto, and F. Bernelli-Zazzera, "Low-energy, low-thrust transfers to the moon," *Celestial Mech. Dyn. Astron.*, vol. 105, nos. 1–3, pp. 61–74, 2009.
 [9] J. T. Betts, "Optimal low-thrust orbit transfers with eclipsing," *Optim. Control Appl. Methods*, Feb. 2014, DOI: 10.1002/oca.2111.
 [10] M. Ehsani, Y. Gao, and J. M. Miller, "Hybrid electric vehicles: Architecture and motor drives," *Proc. IEEE*, vol. 95, no. 4, pp. 719–728, 2007.
 [11] S. G. Wirasingha and A. Emadi, "Classification and review of control strategies for plug-in hybrid electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 60, no. 1, pp. 111–122, 2011.
 [12] R. Lot and S. A. Evangelou, "Lap time optimization of a sports series hybrid electric vehicle," in *Proc. World Congr. Engineering*, London, U.K., vol. III, July 2013, pp. 1711–1716.
 [13] N. Kim, S. Cha, and H. Peng, "Optimal control of hybrid electric vehicles based on pontryagin's minimum principle," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 5, pp. 1279–1287, 2011.
 [14] A. Sciarretta, M. Back, and L. Guzzella, "Optimal control of parallel hybrid electric vehicles," *IEEE Trans. Control Systems Technol.*, vol. 12, no. 3, pp. 352–363, 2004.
 [15] L. V. Perez and E. A. Pilotta, "Optimal power split in a hybrid electric vehicle using direct transcription of an optimal control problem," *Math. Comput. Simul.*, vol. 79, no. 6, pp. 1959–1970, 2009.
 [16] H. Scherenberg, "Mercedes-Benz racing design and cars experience," *Trans. Soc. Automot. Engineers*, no. 580042, pp. 414–420, 1958.
 [17] V. Cossalter, M. D. Lio, R. Lot, and L. Fabbri, "A general method for the evaluation of vehicle manoeuvrability with special emphasis on motorcycles," *Veh. Syst. Dyn.*, vol. 31, no. 2, pp. 113–135, 1999.
 [18] D. Tavernini, M. Massaro, E. Velenis, D. I. Katzourakis, and R. Lot, "Minimum time cornering: The effect of road surface and car transmission layout," *Veh. Syst. Dyn.*, vol. 51, no. 10, pp. 1533–1547, 2013.
 [19] J. M. Hendrikx, T. Meijlink, and R. F. C. Kriens, "Application of optimal control theory to inverse simulation of car handling," *Veh. Syst. Dyn.*, vol. 26, no. 6, pp. 449–461, 1996.
 [20] D. Casanova, "On minimum time vehicle manoeuvring: The theoretical optimal lap," Ph.D. dissertation, School Eng., Cranfield Univ., Cranfield, U.K., 2000.
 [21] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Non-linear Programming*, 2nd ed. Philadelphia, PA: SIAM, 2001.
 [22] D. P. Kelly, "Lap time simulation with transient vehicle and tyre dynamics," Ph.D. dissertation, School Eng., Cranfield Univ., Cranfield, U.K., 2008.
 [23] D. L. Brayshaw and M. F. Harrison, "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. D, J. Automobile Eng.*, vol. 219, no. 6, pp. 725–739, 2005.
 [24] M. Thommypillai, S. A. Evangelou, and R. S. Sharp, "Car driving at the limit by adaptive linear optimal preview control," *Veh. Syst. Dyn.*, vol. 47, no. 12, pp. 1535–1550, 2009.
 [25] J. P. Timings and D. J. Cole, "Minimum maneuver time calculation using convex optimization," *J. Dyn. Syst. Meas. Control*, vol. 135, no. 3, pp. 031 015–031 015–9, 2013.
 [26] G. Perantoni and D. J. Limebeer, "Optimal control for a formula one car with variable parameters," *Veh. Syst. Dyn.*, vol. 52, no. 5, pp. 1–27, Feb. 2014.
 [27] D. J. N. Limebeer, G. Perantoni, and A. V. Rao, "Optimal control of formula one car energy recovery systems," *Int. J. Control*, vol. 87, no. 10, pp. 2065–2080, 2014.
 [28] G. Perantoni and D. J. N. Limebeer, "Optimal control of a formula one car on a three-dimensional track—Part 1: Track modelling and identification," *ASME J. Dyn. Syst. Meas. Control*, to be published, DOI: 10.1115/1.4028253
 [29] D. J. N. Limebeer and G. Perantoni, "Optimal control of a formula one car on a three-dimensional track—Part 2: Optimal control," *ASME J. Dyn. Syst. Meas. Control*, to be published, DOI: 10.1115/1.4029466.
 [30] Fédération Internationale de l'Automobile, "2014 Formula One technical regulations," Tech. Rep., 2013.
 [31] A. Trabesinger, "Power games," *Nature*, vol. 447, pp. 900–903, June 2007.
 [32] A. M. I. Mamat, A. Romagnoli, and R. F. Martinez-Botas, "Design and development of a low-pressure turbine for turbocompounding applications," *Int. J. Gas Turbine, Propul. Power Syst.*, vol. 4, no. 3, pp. 1–8, 2012.
 [33] H. B. Pacejka, *Tyre and Vehicle Dynamics*, 2nd ed. London, U.K.: Butterworth-Heinemann, 2008.

- [34] A. E. J. Bryson and Y.-C. Ho, *Applied Optimal Control: Optimization, Estimation, and Control*. New York: Hemisphere, 1975.
- [35] D. E. Kirk, *Optimal Control Theory: An Introduction*. Englewood Cliffs, NJ: Prentice-Hall, 1970.
- [36] D. G. Hull, *Optimal Control Theory for Applications*. New York: Springer-Verlag, 2003.
- [37] F. L. Lewis and V. L. Syrmos, *Optimal Control*, 2nd ed. New York: Wiley, 1995.
- [38] M. A. Athans and P. L. Falb, *Optimal Control: An Introduction to the Theory and its Applications*. Mineola, NY: Dover, 2006.
- [39] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 2. Belmont, MA: Athena Scientific, 2007.
- [40] H. J. Sussmann and J. C. Willems, "300 years of optimal control: From the brachistochrone to the maximum principle," *IEEE Control Syst. Mag.*, vol. 17, no. 3, pp. 32–44, 1997.
- [41] A. E. Bryson, "Optimal control—1950 to 1985," *IEEE Control Syst. Mag.*, vol. 13, no. 3, pp. 26–33, 1996.
- [42] J. Betts, "Survey of numerical methods for trajectory optimization," *J. Guid. Control Dyn.*, vol. 21, no. 2, pp. 193–207, 1998.
- [43] B. A. Conway. (2012). A survey of methods available for the numerical optimization of continuous dynamic systems. *J. Optim. Theory Applicat.* vol. 152, no. 2, pp. 271–306. [Online]. Available: <http://dx.doi.org/10.1007/s10957-011-9918-z>
- [44] G. A. Bliss, *Lectures on the Calculus of Variations*. Chicago, IL: Univ. Chicago Press, 1946.
- [45] R. Weinstock, *Calculus of Variations*. Mineola, NY: Dover, 1974.
- [46] F. B. Hildebrand, *Methods of Applied Mathematics*. Mineola, NY: Dover, 1992.
- [47] G. Leitmann, *The Calculus of Variations and Optimal Control*. New York: Springer-Verlag, 1981.
- [48] W. H. Fleming and R. W. Rishel, *Deterministic and Stochastic Optimal Control*. Heidelberg, Germany: Springer-Verlag, 1982.
- [49] R. F. Stengel, *Optimal Control and Estimation*. Mineola, NY: Dover, 1994.
- [50] R. Vinter, *Optimal Control (Systems & Control: Foundations and Applications)*. Boston, MA: Birkhäuser, 2000.
- [51] L. S. Pontryagin, *Mathematical Theory of Optimal Processes*. New York: Wiley, 1962.
- [52] N.-X. Vinh, *Optimal Trajectories in Atmospheric Flight*. New York: Elsevier Science, 1981.
- [53] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, 3rd ed. New York: Wiley-Interscience, 2006.
- [54] D. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 2004.
- [55] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [56] U. M. Ascher, R. M. Mattheij, and R. D. Russell, *Numerical Solution of Boundary-Value Problems in Ordinary Differential Equations*. Philadelphia, PA: SIAM Press, 1996.
- [57] H. B. Keller, *Numerical Solution of Two Point Boundary Value Problems*. Philadelphia, PA: SIAM Press, 1976.
- [58] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*. Berlin Heidelberg, Germany: Springer-Verlag, 2002.
- [59] A. V. Rao, "Extension of the computational singular perturbation method to optimal control," Ph.D. dissertation, Dept. Mechan. Aerospace Eng., Princeton Univ., Princeton, NJ, 1996.
- [60] A. V. Rao and K. D. Mease, "Dichotomic basis approach to solving hyper-sensitive optimal control problems," *Automatica*, vol. 35, no. 4, pp. 633–642, Apr. 1999.
- [61] A. V. Rao and K. D. Mease, "Eigenvector approximate dichotomic basis method for solving hyper-sensitive optimal control problems," *Optim. Control Applicat. Methods*, vol. 21, no. 1, pp. 1–19, Jan.–Feb. 2000.
- [62] A. V. Rao, "Application of a dichotomic basis method to performance optimization of supersonic aircraft," *J. Guid. Control Dyn.*, vol. 23, no. 3, pp. 570–573, May–June 2000.
- [63] A. V. Rao, "Riccati dichotomic basis method for solving hyper-sensitive optimal control problems," *J. Guid. Control Dyn.*, vol. 26, no. 1, pp. 185–189, Jan.–Feb. 2003.
- [64] J. S. Logsdon and L. T. Biegler, "Accurate solution of differential-algebraic optimization problems," *Ind. Eng. Chem. Res.*, vol. 28, no. 11, pp. 1628–1639, 1989.
- [65] A. Schwartz, "Theory and implementation of numerical methods based on Runge-Kutta integration for solving optimal control problems," Ph.D. dissertation, Dept. Electr. Eng., Univ. California, Berkeley, CA, 1996.
- [66] A. Schwartz and E. Polak, "Consistent approximations for optimal control problems based on Runge-Kutta integration," *SIAM J. Control Optim.*, vol. 34, no. 4, pp. 1235–1269, July 1996.
- [67] W. W. Hager, "Runge-Kutta methods in optimal control and the transformed adjoint system," *Numer. Math.*, vol. 87, no. 2, pp. 247–282, 2000.
- [68] A. L. Dontchev, W. W. Hager, and K. Malanowski, "Error bounds for the Euler approximation and control constrained optimal control problem," *Numer. Funct. Anal. Applicat.*, vol. 21, nos. 5–6, pp. 653–682, 2000.
- [69] A. L. Dontchev, W. W. Hager, and V. M. Veliov, "Second-order Runge-Kutta approximations in constrained optimal control," *SIAM J. Numer. Anal.*, vol. 38, no. 1, pp. 202–226, 2000.
- [70] A. L. Dontchev, W. W. Hager, and V. M. Veliov, "Uniform convergence and mesh independence of Newton's method for discretized variational problems," *SIAM J. Control Optim.*, vol. 39, no. 3, pp. 961–980, 2000.
- [71] A. L. Dontchev and W. W. Hager, "The Euler approximation in state constrained optimal control," *Math. Comput.*, vol. 70, no. 233, pp. 173–203, 2001.
- [72] A. L. Dontchev and W. W. Hager, "A new approach to Lipschitz continuity in state constrained optimal control," *Syst. Contr. Lett.*, vol. 35, no. 3, pp. 137–143, 1998.
- [73] A. L. Dontchev and W. W. Hager, "Lipschitzian stability for state constrained nonlinear optimal control," *SIAM J. Control Optim.*, vol. 36, no. 2, pp. 696–718, 1998.
- [74] P. E. Gill, W. Murray, and M. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM J. Optim.*, vol. 12, no. 4, pp. 979–1006, 2002.
- [75] R. H. Byrd, J. Nocedal, and R. A. Waltz, *KNITRO: An Integrated Package for Nonlinear Optimization*, vol. 83. Berlin Heidelberg, Germany: Springer-Verlag, 2006, pp. 35–59.
- [76] D. Garg, M. A. Patterson, W. W. Hager, A. V. Rao, D. A. Benson, and G. T. Huntington, "A unified framework for the numerical solution of optimal control problems using pseudospectral methods," *Automatica*, vol. 46, no. 11, pp. 1843–1851, Nov. 2010.
- [77] D. Garg, M. A. Patterson, C. L. Darby, C. Franconin, G. T. Huntington, W. W. Hager, and A. V. Rao, "Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems via a Radau pseudospectral method," *Comput. Optim. Applicat.*, vol. 49, no. 2, pp. 335–358, 2011.
- [78] D. Garg, W. W. Hager, and A. V. Rao, "Pseudospectral methods for solving infinite-horizon optimal control problems," *Automatica*, vol. 47, no. 4, pp. 829–837, 2011.
- [79] S. Kameswaran and L. T. Biegler, "Convergence rates for direct transcription of optimal control problems using collocation at Radau points," *Comput. Optim. Applicat.*, vol. 41, no. 1, pp. 81–126, 2008.
- [80] C. L. Darby, W. W. Hager, and A. V. Rao, "An hp-adaptive pseudospectral method for solving optimal control problems," *Optim. Control Applicat. Methods*, vol. 32, pp. 476–502, Aug. 2011.
- [81] C. L. Darby, W. W. Hager, and A. V. Rao, "Direct trajectory optimization using a variable low-order adaptive pseudospectral method," *J. Spacecraft Rockets*, vol. 48, no. 3, pp. 433–445, 2011.
- [82] M. A. Patterson, W. W. Hager, and A. V. Rao, "A ph mesh refinement method for optimal control," *Optim. Control Applicat. Methods* (published online in early view), Jan. 2014, DOI: 10.1002/oca.2114.
- [83] F. Liu, W. W. Hager, and A. V. Rao, "Adaptive mesh refinement for optimal control using discontinuity detection and mesh size reduction," in *Proc. IEEE Conf. Decision Control*, Los Angeles, CA, Dec. 2014.
- [84] M. A. Patterson and A. V. Rao, "Exploiting sparsity in direct collocation pseudospectral methods for solving continuous-time optimal control problems," *J. Spacecraft Rockets*, vol. 49, no. 2, pp. 364–377, Mar.–Apr. 2012.
- [85] C. C. Franconin, W. W. Hager, and A. V. Rao, "Costate approximation in optimal control using integral Gaussian quadrature collocation methods," *Optim. Control Applicat. Methods* (published online in early view), 2014, DOI: 10.1002/oca.2112.
- [86] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York: Dover, 1965.
- [87] M. A. Patterson and A. V. Rao, "GPOPS-II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive Gaussian quadrature collocation methods and sparse nonlinear programming," *ACM Trans. Math. Softw.*, vol. 39, no. 3, pp. 1:0–1:41, 2013.

