Technische Universität München

TUM School of Engineering and Design

Optimization-based Energy Strategy for Autonomous Electric Race Cars

Thomas Herrmann, M.Sc.

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz:	Prof. Dr.-Ing. Michael W. Gee
Prüfer der Dissertation:	1. Prof. Dr.-Ing. Markus Lienkamp
	2. Assist. Prof. Dr. Ir. Mauro R. U. Salazar

Die Dissertation wurde am 09.05.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 19.09.2022 angenommen.

“[Formula-E is] like playing chess at 200 kph.”

Lucas di Grassi, 2017 [1, p. 50]

Acknowledgment

First, I would like to thank my supervisor, Prof. Dr.-Ing. Markus Lienkamp, for welcoming me to his lab, the Institute of Automotive Technology at the Technical University of Munich. He offered me the opportunity to conduct research in the fascinating field of autonomous driving. He not only provided a remarkable working environment but also formed an impressive team of scientists I was honored to conduct research with. The high level of trust I was met with and the fruitful discussions have been the foundation of this work. I want to say “grazie mille” to Assist. Prof. Dr. Ir. Mauro Salazar for being my co-examiner and for inviting me to his group in the Control Systems Technology section at the Eindhoven University of Technology. His way of quickly interpreting complex mathematical relationships from an engineers point of view has inspired my future work. Furthermore, I would like to thank Prof. Dr.-Ing. Michael Gee for accepting the chairmanship of the examination.

Second, I would like to acknowledge the Bavarian Research Foundation and my industry partners in-tech GmbH, IAV GmbH, and Dräxlmaier GmbH & Co. KG for financing and supporting my research project “Deep Learning for Autonomous Driving on the Race Track – rAIcing”. In addition, a big thank you goes to the company Roborace and all of their former and current employees for providing, maintaining, and operating outstanding, electric prototype race cars. Thank you for organizing the competition events, which gave me the opportunity to put science into practice and to collect tons of measurement data. Also, it was a fascinating experience to internationally connect with the researchers and engineers I have met there.

Third, my gratitude goes to my colleagues at the Institute of Automotive Technology for the fruitful collaborations, the inspiring discussions, and for making long-lasting friendships. I would especially like to thank my former TUM Autonomous Motorsport team members Johannes Betz, Alexander Heilmeier, Leonhard Hermansdorfer, Felix Nobis, Tim Stahl, and Alexander Wischnewski for proofreading this thesis, for the marvelous team work, and the unforgettable time we spent together in preparation for and at the Roborace events. I owe gratitude to my additional proofreaders, Alexander Koch, Francesco Passigato, and Ilse New, who spent a remarkable amount of time to help me improve the clarity of the formulations in this thesis. My thanks also goes to all the students I supervised during my time at the institute. They contributed by valuable insights about the topic of this thesis through the work on their own theses.

Finally, I would like to express my deepest gratitude to my grandparents, Franz and Anna, my parents, Herbert and Sabine, as well as my brother Stefan, Raphaela, and Anna. You not only supported and encouraged me long before this PhD journey had even started, but always created an atmosphere where I have felt welcome.

Garching, April 2022

Thomas Herrmann

Contents

List of Abbreviations

AC	Alternating Current
ACADO	Automatic Control and Dynamic Optimization
ADMM	Alternating Direction Method of Multipliers
ASM	Asynchronous Machine
BLDC	Brushless Direct Current
DAE	Differential Algebraic Equation
DC	Direct Current
DIRECT	DIvided RECTangles
DP	Dynamic Programming
ECM	Equivalent Circuit Model
ECMS	Equivalent Consumption Minimization Strategy
ECOS	Embedded Conic Solver
ECU	Electric Control Unit
EIS	Electrochemical Impedance Spectroscopy
ES	Energy Strategy
EV	Electric Vehicle
FEM	Finite Element Method
FOC	Field-Oriented Control
FONC	First-Order Necessary Condition
GA	Genetic Algorithm
GPOPS	General Purpose OPtimal Control Software
GPOPS-II	General Purpose OPtimal Control Software
GPS	Global Positioning System
GRAMPC	GRAdient-based MPC
HIL	Hardware-in-the-Loop
HPIPM	High-Performance Interior Point Method
ICE	Internal Combustion Engine
IGBT	Insulated-Gate Bipolar Transistor
IP	Interior Point
IPOPT	Interior Point OPTimizer
KKT	Karush-Kuhn-Tucker
LiDAR	Light Detection and Ranging
LP	Linear Program
LTPL	Local Trajectory Planner
MILP	Mixed-Integer Linear Program
MIQP	Mixed-Integer Quadratic Program
MLTP	Minimum Lap Time Problem
MOSFET	Metal Oxide Semiconductor Field Effect Transistor

MPC	Model Predictive Control
mpSQP	multi-parametric Sequential Quadratic Program
MRTP	Minimum Race Time Problem
MUSCOD-II	MUltiple Shooting COde for Direct Optimal Control
NLP	Nonlinear Program
NMPC	Nonlinear Model Predictive Control
NMPP	Nonlinear Model Predictive Planner
NOCP	Nonlinear Optimal Control Problem
NRMSE	Normalized Root Mean Square Error
OCP	Optimal Control Problem
OCV	Open Circuit Voltage
ODE	Ordinary Differential Equation
OS	Open Source
OSQP	Operator Splitting Quadratic Program
PMP	Pontryagin's Minimum Principle
PMSM	Permanent Magnet Synchronous Machine
PSO	Particle Swarm Optimization
PWM	Pulse Width Modulation
QCQP	Quadratically Constrained Quadratic Program
QP	Quadratic Program
qpOASES	QP Online Active SEt Strategy
RADAR	Radio Detection and Ranging
RC	Resistor Capacitor
SA	Simulated Annealing
SNOPT	Sparse Nonlinear OPTimizer
SOC	State of Charge
SOCP	Second Order Conic Program
SONC	Second-Order Necessary Condition
SOSC	Second-Order Sufficient Condition
SQP	Sequential Quadratic Program
SynRM	Synchronous Reluctance Machine
VSI	Voltage Source Inverter

Formula Symbols

The formula symbols used throughout this thesis will be explained following their introduction and in the corresponding publications.

Foreword

The present thesis documents parts of the research carried out by the author during his doctoral studies under the supervision of Prof. Dr.-Ing. Markus Lienkamp at the Institute of Automotive Technology at the Technical University of Munich (TUM) between August 2018 and January 2022. During this time, the author conducted two projects in the field of autonomous driving, which comprise the participation in the Roborace competition and the Indy Autonomous Challenge.

The spectrum of this thesis covers the publications, code, and documentations in [2–11], which are connected to the Roborace project. For the sake of consistency, the additional research at TUM, documented in [12–20] won't be discussed in the remainder of this document. Neither will be the publications in [21, 22], which have been developed in the Control Systems Technology section at the Eindhoven University of Technology (TU/e), nor the content of the book chapter in [23].

1 Introduction and Problem Statement

High-performance racing applications have been serving as a development and testing platform for technological innovations [24]. Since the automation and the electrification of vehicles pertain to the key developments in the automotive domain, racing series combining both technologies have been established [25]. In the following, we will therefore introduce these developments and their challenges in greater detail and derive the underlying motivation and the goal of this thesis therefrom.

1.1 Future Trends in the Automotive Domain

Autonomous Driving

One key trend regarding transportation standards are autonomous vehicles within the next decade [26], which are enhanced with perception, reasoning/decision-making, and controllable actuators [27, p. 1176]. The reasons for this development include the optimization of the traffic flow [28] [29, p. 4], which has already been subject of the well-known “Programme for a European traffic with highest efficiency and unprecedented safety” (Prometheus, 1987–1994) [30, chap. 2]. By improving the traffic flow, the energy efficiency of road vehicles is increased [27, p. 1176] [31] and, through this, the amount of CO_2-emissions can significantly be reduced. Although the number of traffic accidents is constantly decreasing, still 1.35 million people have died in road accidents worldwide in 2018 [32]. Autonomous driving promises to significantly diminish this number [29, chap. 17] [27, p. 1176].

The necessary functional blocks for autonomous driving are the sensing, processing and decision-making, and vehicle control [33, p. 6]. They can also be found in the high-level architecture of an autonomous driving system presented by [34, p. 46], which is depicted similarly in Figure 1.1.

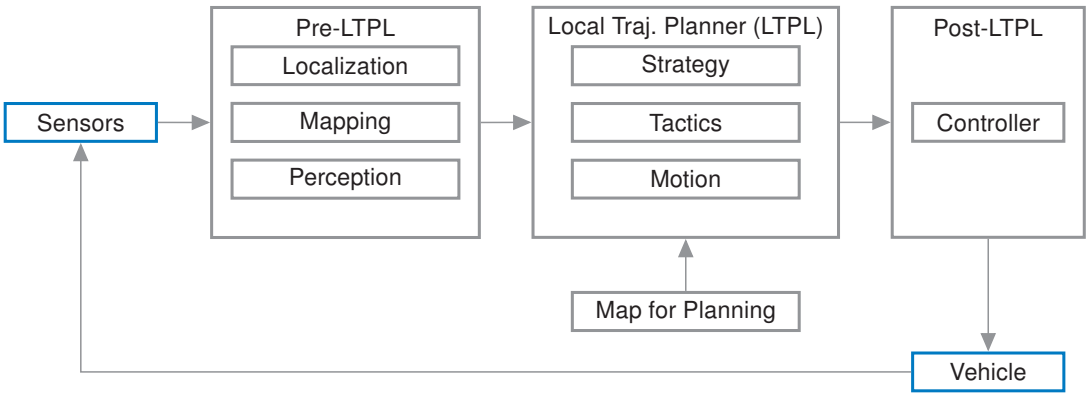


Figure 1.1: High-level architecture of an autonomous driving system, classified according to their position regarding the local trajectory planner (LTPL), which is similar to [34, p. 46].

The autonomous vehicle senses its environment with the help of, e.g., light detection and ranging (LiDAR), radio detection and ranging (RADAR) or Cameras, and determines its state of motion with the additional help of the global positioning system (GPS) and wheel speed sensors. Simultaneously, a map of the environment can be created and saved. Subsequently, the gathered environmental information is used by the local trajectory planning step, which comprises a strategic, tactical, and motion planning part. Briefly, the strategic planning determines a high-level long-term goal, the tactical part identifies reasonable lane changes or overtaking maneuvers, and the motion planning calculates a finely discretized trajectory, consisting of path and velocity, on the available driving surface [34, p. 47]. A dynamics controller operates the actuators (steering, throttle, brake) to have the vehicle perform either a re-optimized version of the planned trajectory (in the case of model predictive control (MPC)) or to make it follow exactly. It is also possible to classify these modules according to the famous scheme “navigation, guidance, and control” [35, p. 183], which we adapted slightly to “pre-local trajectory planner (LTPL), LTPL, and post-LTPL” to emphasize the local planning task in this architecture.

Electrified Vehicles

Alternative powertrains are gaining increasing significance [36]. By 2020, the number of electrified vehicles surpassed 10 million worldwide and is still growing [37, p. 17]. Reasons for the transition from internal combustion engine (ICE)-powered vehicles to electric vehicles (EVs) are diverse. First, it is argued that EVs have a smaller environmental impact regarding global warming and urban air quality through avoiding local emissions, and will therefore boost public health [38–40]. Second, energy security pressures have encouraged manufacturers to increase their development activities in EVs [39].

Despite this transition process, EVs are still facing many technological challenges. These include a limited driving range, a higher charge time compared to the conventional refueling time [38], and smaller thermal operating ranges of the electrical components in comparison to conventional powertrains. Electric machines used for traction applications usually do not exceed the temperatures above $170\,^{\circ}$C [41]. The battery cells in EVs should not be operated above $55\,^{\circ}$C [42].

Electric Race Vehicles Competing Fully Autonomously

In the past, racing series have been a development and testing platform for technological innovations [24]. Arising from both aforementioned developments, it is therefore a logical step that racing series like the Formula E [43], pushing the boundaries of electric powertrain technology, or the Indy Autonomous Challenge [44], aiming to extend the limits of autonomous driving software, have been evolving over the past years. Roborace [25] represents a competition for autonomous race vehicles with electric powertrains and, therefore, serves as a platform for technological advancements in an area combining both domains [45].

1.2 Motivation and Goal

A high-performance racing application reinforces the mentioned challenges of autonomous EVs, since they must meticulously adhere to the small technical operating ranges of their powertrain components to avoid safety shutdowns. Nevertheless, these limits must permanently be exploited to have the vehicle operate at its limits of handling to leverage maximum performance. To reach

this goal, an appropriate Energy Strategy (ES), which seamlessly integrates into an existing autonomous driving software stack, is necessary. By doing so, the race car reaches the minimal race time from a point of view comprising its driving dynamics, its thermodynamics, and the available amount of energy. Thus, the goal of this thesis involves the concept, the development and the implementation of an ES, which is closely related to and part of the local trajectory planning module (Figure 1.1).

The objective of the ES is the calculation of a time-minimal race strategy in a traffic-free environment. Whilst optimizing this goal, the ES must meet the following requirements:
The ES

- must adhere to the technical limitations of the electric powertrain and the driving dynamics. Nevertheless, the ES must exploit the available limited amount of battery energy as “lap-time-efficiently” as possible to achieve a time-minimal result. Lap-time-efficient expresses that the vehicle must travel as fast as possible while adhering to the available energy budget.
- must be real-time-capable whilst simultaneously correctly predicting the powertrain behavior with small physical errors.
- must be implementable in an existing software stack used for autonomous driving.
- must effortlessly be expandable by additional physical constraints whilst being computationally efficient.

Throughout the explanations of the developed algorithms, which realize the ES in this work, we will refer to the aforementioned stated objective and the incorporation of these conflicting constraints.

2 Preliminaries and State of the Art

First, this chapter describes the state of the art in the field of electric powertrain design. The physical working principles of its main components, and their efficient mathematical modeling are explained. Furthermore, we analyze the generation of power losses therein, and additionally discuss thermodynamic characteristics. The powertrain architecture we will focus on is the one used inside the Roborace “DevBot 2.0” [12], which is the unitary race vehicle among the teams participating in the Roborace competition. Roborace itself is the world’s first racing series for autonomous electric race vehicles at full scale. Second, we summarize the mathematical background, which is necessary to select appropriate optimization procedures for the deduction of an online-capable ES. Finally, we depict the current advances in the field of trajectory planning algorithms for autonomous driving.

2.1 Powertrain Design and Modeling

The primary electric propulsion components in a battery-electric powertrain are constituted by an energy storage, namely the battery pack, one or multiple voltage source inverters (VSIs), and one or multiple electric machines [46, 47], [48, p. 6]. These are commonly called “electric motors”, which does not express the generative operation capabilities in the automotive domain [49]. In this section we will introduce the basic working principles of these powertrain components and describe their behavior using mathematical models. Moreover, the mechanisms leading to power losses in each of them will be explained.

Figure 2.1 displays an all-electric, rear-wheel-drive powertrain architecture, which is used inside the Roborace vehicle DevBot 2.0. It can be extended to an all-wheel-drive architecture by adding the propelling components, which will be introduced in the following, also to the front of the powertrain architecture. Its energy storage (B) consists of rechargeable batteries, and is mounted directly underneath the bodywork. In the back, the inverters ($\mathrm{I}_{\mathrm{l/r}}$) on the left and right hand side convert the direct current (DC) power from the energy storage into alternating current (AC) power to feed the electric motors ($\mathrm{M}_{\mathrm{l/r}}$). These are mechanically connected to the wheels ($\mathrm{W}_{\mathrm{rl/rr}}$) by the gearboxes ($\mathrm{G}_{\mathrm{l/r}}$) that translate the motor output torque. We summarize the autonomous driving sensors, the computing units, and other consumers drawing a low-voltage DC power as auxiliaries (A_{x}). The powertrain of the Formula E car “Gen2” is built similarly with a single inverter and electric motor [43].

An active liquid cooling system dissipates the powertrain heat losses. Since the maximum temperature of the inverters and motors is higher compared to the battery, two separate cooling circuits are installed. The radiators (R_{MI} and R_{B}) temper the respective cooling fluids.

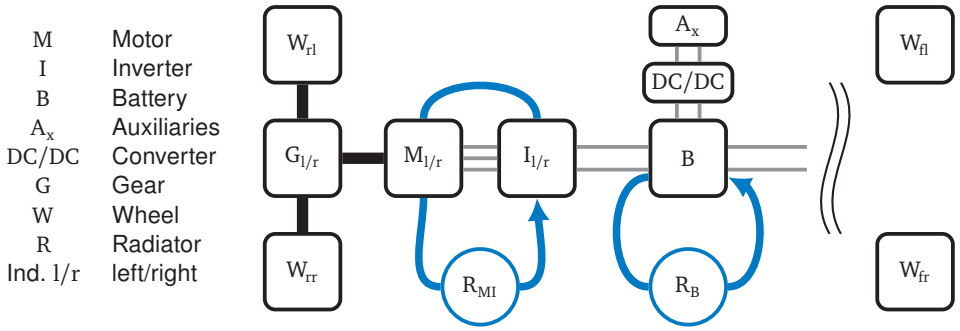


Figure 2.1: All-electric, rear-wheel-drive powertrain architecture including separate cooling circuits and their radiators $\mathrm{R_{MI}}$ and $\mathrm{R_B}$ to temper the combination of motors $\mathrm{M_{l/r}}$ and inverters $\mathrm{I_{l/r}}$ at the rear left and right, and the battery B. Auxiliaries are denoted by $\mathrm{A_x}$ including a DC/DC-converter from high to low voltage, wheels and gears are depicted by the symbols W and G. Bold straight lines indicate mechanical connections, thin lines the electrical ones, and curved arrows the cooling liquids.

2.1.1 Energy Storage

In the Roborace DevBot 2.0, the energy storage consists of lithium-ion batteries, which follows today's trends in EVs [50, p. 50], [51, 52]. When discharging these batteries to produce a usable electrical current, lithium-ions move from the negative to the positive electrode through a solid or liquid electrolyte; this process reverses during charging. To form an energy storage, which fulfills the electrical requirements regarding voltage, current, and energy capacity of an EV, the battery cells are interconnected in series and in parallel, resulting in an XsYp battery pack configuration. According to the big cell-method, the serial combination scales the pack output voltage, and the parallel connection the total capacity [53].

Working Principle

To describe the electrical behavior of lithium-ion cells, the literature mainly uses and differentiates between three major modeling approaches [54], which are

- mechanistic (physico-electrochemical) models [55–57],
- equivalent circuit models (ECMs) [52, 58–62],
- data-driven models: black [18] and gray boxes [63–65].

They can also be combined to hybrid models [66–68].

In [53], the properties of the mentioned modeling approaches are listed. Therein, the model accuracy is emphasized and put into perspective with the corresponding computational complexity, which is important in real-time applications. We summarize the main characteristics of the aforementioned battery modeling approaches in Table 2.1 and assess their applicability for online optimal control problems (OCPs). Mechanistic models require a large set of system parameters, which are often unknown and difficult to measure [56]. Data-driven ones do not provide information about the local rates of change of the system dynamics [54].

Due to their high accuracy [59], practical applicability, and real time capability [70] ECMs are widely used in the automotive domain [59, 71]. They are based on the Thevenin model, which in the simplest case consists of an ideal no-load battery voltage, a constant resistor capacitor (RC) element, and a constant internal resistance [59, 71, 72]. Therein, the electrical components are parameterized using electrochemical impedance spectroscopy (EIS) measurements [73].

Table 2.1: Characteristics and online capability of different modeling approaches of lithium-ion battery cells.

	Mechanistic model	**ECM**	**Data-driven model**
Characteristics	• The modeling is based on the ordinary differential equations ODEs describing the physico-chemical relationships [69]. • Many system parameters are necessary [69], which are often unknown and difficult to measure [56].	• A combination of standard electrical components describes the dominating battery dynamics [54]. • ECMs are easily tuneable, transferable among different battery types due to the lumping of electrochemical processes, and scalable to pack or module level [54]. • The model quality is strongly connected to the measurement data quality [54].	• The nonlinear relationships between the battery states and its performance features are completely described relying on data-driven techniques. • Black box models omit an underlying physical model, gray box models fuse physically motivated models with data-driven algorithms [54].
Applicability in online OCPs	Mechanistic models are too complex for most control applications [53] but provide detailed insights into the battery behavior via sensitivity analyses.	ECMs combine a real time applicability in OCPs including sufficient accuracy for ES control systems [52, 70], especially at low sample rates and long prediction horizons [53].	The governing equations can have rather low computational complexity but they do not provide information about local dynamics, which are necessary for control purposes [54].

For the deduction of an ES for eco-driving applications [50, p. 50] proposes an ECM of 0th order, which consists of an ideal voltage source U_{oc} and a constant internal resistance R_{i}. The model accuracy can be increased by adding a temperature dependency (T_{B}) to the internal resistance $R_{\mathrm{i}}(T_{\mathrm{B}})$ or a state of charge (SOC)-open circuit voltage (OCV)-relationship [71, 74, 75]. Figure 2.2 depicts a widely used version of an ECM for electric vehicle modeling and simulation purposes [71]. It comprises two RC elements [59, 76–78] and follows the dynamics [53]:

$$\frac{\partial u_k(t)}{\partial t} = -\frac{u_k(t)}{R_k C_k} + \frac{i(t)}{C_k}, \tag{2.1a}$$

$$u_{\mathrm{dc}}(t) = u_{\mathrm{oc}}(t) - R_{\mathrm{i}} i(t) - \sum_{k=1}^{2} u_{\mathrm{k}}(t). \tag{2.1b}$$

Here, $k = 2$ denotes the number of RC-elements, $u_k(t)$ the voltage drop over the kth RC-element, R_k and C_k the RC resistance and capacitance, $u_{\mathrm{oc}}(t)$ the OCV, $u_{\mathrm{dc}}(t)$ the terminal voltage, $i(t)$ the battery current, and R_{i} the internal cell resistance.

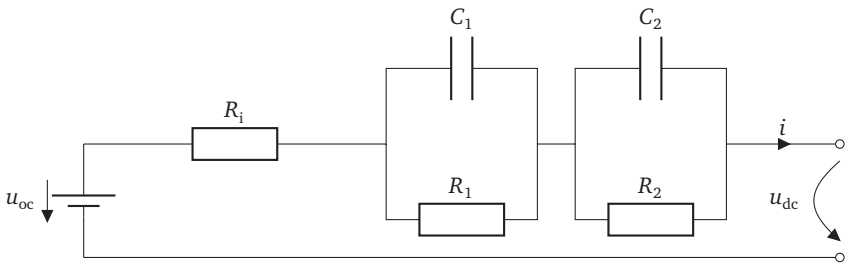


Figure 2.2: ECM of second order of a lithium-ion battery cell [53].

Power Loss

Thermal power in battery cells occurs either through the internal resistance $R_\mathrm{i}(t)$, a change in entropy, or overpotential losses. When considering an ECM of second order (Figure 2.2), the change in heat energy $Q_\mathrm{B}(t)$ can be expressed as [53, 79]

$$\frac{\mathrm{d}Q_\mathrm{B}(t)}{\mathrm{d}t} = \underbrace{R_\mathrm{i}(t)i^2(t)}_{\dot{Q}_\mathrm{B,ih}} + \underbrace{\frac{1}{n_\mathrm{mol}F}T_\mathrm{B}(t)\Delta S\left(\sigma(t)\right)i(t)}_{\dot{Q}_\mathrm{B,re}} + \underbrace{i(t)\sum_{k=1}^{2}u_k(t)}_{\dot{Q}_\mathrm{B,op}}, \tag{2.2}$$

where $\dot{Q}_\mathrm{B,ih}$ denotes the irreversible resistive heating power, $\dot{Q}_\mathrm{B,re}$ the reversible effect of heat generation through a change in entropy ΔS, and $\dot{Q}_\mathrm{B,op}$ the overpotential losses due to kinetic and mass transport effects. To calculate $\dot{Q}_\mathrm{B,re}$, the Faraday constant F, the number of electrons per reaction n_mol, and the battery SOC $\sigma(t)$ are involved. The change in entropy ΔS is

$$\Delta S\left(\sigma\left(t\right)\right) = \frac{\partial u_\mathrm{oc}\left(\sigma(t)\right)}{\partial T_\mathrm{B}} n_\mathrm{mol}F, \tag{2.3}$$

which indicates a strong dependency on the battery SOC $\sigma(t)$. Since the major source of power loss is the irreversible heating power $\dot{Q}_\mathrm{B,ih}$, especially in high power applications, $\dot{Q}_\mathrm{B,re}$ and $\dot{Q}_\mathrm{B,op}$ are often neglected [53]. Assuming an ECM with a time-dependent voltage source $u_\mathrm{oc}(t)$ and an internal resistance $R_\mathrm{i}(t)$, the following equation describes the battery losses $P_\mathrm{l,B}(t)$ directly [50, p. 50]:

$$P_\mathrm{l,B}\left(P_\mathrm{o,B}(t)\right) = \underbrace{\frac{u_\mathrm{oc}^2(t)}{2R_\mathrm{i}(t)} - u_\mathrm{oc}(t)\frac{\sqrt{u_\mathrm{oc}^2(t) - 4P_\mathrm{o,B}(t)R_\mathrm{i}(t)}}{2R_\mathrm{i}(t)}}_{P_\mathrm{i,B}(t)} - P_\mathrm{o,B}(t), \tag{2.4}$$

where $P_\mathrm{i/o,B}(t)$ describe the internal and the output power of the battery cell.

The internal battery power $P_\mathrm{i,B}(t)$ causes the SOC dynamics $\dot{\sigma}$, which can be indicated by the Coulomb counting method [80, p. 71], [50, p. 51], [79]:

$$\frac{\mathrm{d}\sigma(t)}{\mathrm{d}t} = -\frac{P_\mathrm{i,B}(t)}{E_\mathrm{B}}, \tag{2.5}$$

where E_B denotes the battery energy capacity.

The literature reports an operating range between $-20\,^\circ$C to $55\,^\circ$C when discharging and between $0\,^\circ$C to $45\,^\circ$C when charging for automotive lithium-ion cells [42].

2.1.2 Voltage Source Inverter

In general, the VSI in an EV powertrain transforms DC and AC power bidirectionally to propel the electric motors or to recharge the energy storage during recuperation phases [81–83]. In the remainder of this section we drop the time dependency of the battery DC voltage to simplify the explanations of the VSI working principle. VSIs consist of active power semiconductor devices like insulated-gate bipolar transistors (IGBTs) or metal oxide semiconductor field effect transistors (MOSFETs), where they are used as motor load drivers [84, p. 117]. In this work, we focus on IGBTs, since they operate in the Roborace DevBot 2.0 VSI. The basic semiconductor

load switching configurations are classified into the groups, low-side, high-side or half-bridge switching, where the latter one is chosen in automotive powertrain applications [84, p. 118].

Working Principle

VSIs can be classified into two main categories, which are square wave inverters, that only allow controlling the output frequency, and pulse width modulation (PWM) inverters, that additionally enable the control of the output signal amplitude [85, p. 267]. A hardware circuit realizing the DC-AC power conversion is depicted in Figure 2.3. This three-phase bridge circuit comprises six IGBTs T_n and free-wheeling diodes D_n to form the piecewise-constant output pole voltages $u_{\mathrm{U/V/WN}}(t)$ with the levels of $0\,\mathrm{V}$, $-\frac{U_{\mathrm{dc}}}{2}$, or $+\frac{U_{\mathrm{dc}}}{2}$ [85, p. 278]. The freewheeling diodes D_n enable a bidirectional power flow in the three half-bridges and prevent from over-voltage peaks.

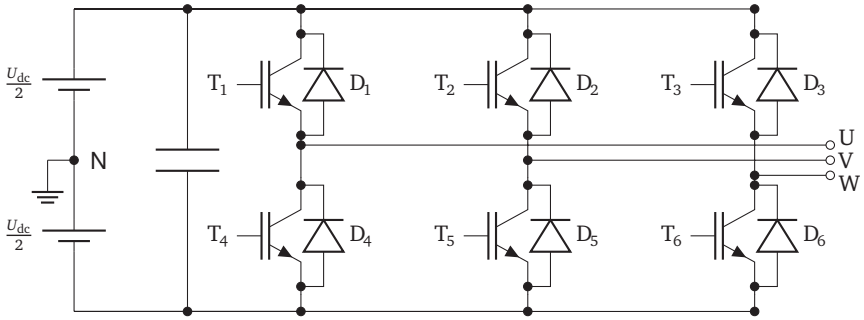


Figure 2.3: B6 three-phase bridge circuit [81, 82].

Figure 2.4 visualizes the working principle of a square wave VSI. In the currently active switching period the transistors T_1, T_3, and T_5 are active, i.e., conducting. Therefore, the poles U and W are connected with the positive potential and pole V with the negative one. After one sixth of the switching period, T_3 is switched off and, alternately, T_6 is switched on [85, p. 277]. In total, each transistor conducts for one half of the switching period [86, 87], i.e., six on-/off-switchings per fundamental period of the output signal $\frac{1}{f_0}$ take place. The motor phase voltages $u_{\mathrm{U/V/WP}}(t)$ result, including the fundamental component $u_{\mathrm{U/V/WP},1}(t)$ as a sinus-wave with an amplitude of the value $\frac{2}{\pi}U_{\mathrm{dc}}$. The line-to-line motor voltages are [85, p. 278], [88, p. 315]

$$u_{\mathrm{UV}}(t) = u_{\mathrm{UN}}(t) - u_{\mathrm{VN}}(t), \quad u_{\mathrm{UW}}(t) = u_{\mathrm{UN}}(t) - u_{\mathrm{WN}}(t), \quad u_{\mathrm{VW}}(t) = u_{\mathrm{VN}}(t) - u_{\mathrm{WN}}(t), \tag{2.6}$$

the motor phase voltages $u_{\mathrm{U/V/WP}}(t)$ can be obtained from the pole line-to-line voltages [85, p. 279]:

$$u_{\mathrm{UP}}(t) = \frac{1}{3}\left(u_{\mathrm{UV}}(t) - u_{\mathrm{WU}}(t)\right), \quad u_{\mathrm{VP}}(t) = \frac{1}{3}\left(u_{\mathrm{VW}}(t) - u_{\mathrm{UV}}(t)\right), \quad u_{\mathrm{WP}}(t) = \frac{1}{3}\left(u_{\mathrm{WU}}(t) - u_{\mathrm{VW}}(t)\right). \tag{2.7}$$

Besides the square wave technique, different PWM methods exist [85, p. 315], [89], which are divided into programmed PWM, carrier-based PWM, and space vector PWM methods, whereby the carrier-based methods are further subdivided into sinusoidal and third harmonic injection PWMs. The space vector PWM methods split into continuous and discontinuous ones.

Figure 2.5 depicts the working principle of the sinusoidal PWM. Here, a carrier signal is permanently compared to the desired reference signal $u_{\mathrm{ref}}(t)$. If the value of the reference signal is higher than the carrier signal, the upper switch of the corresponding VSI pole is turned on.

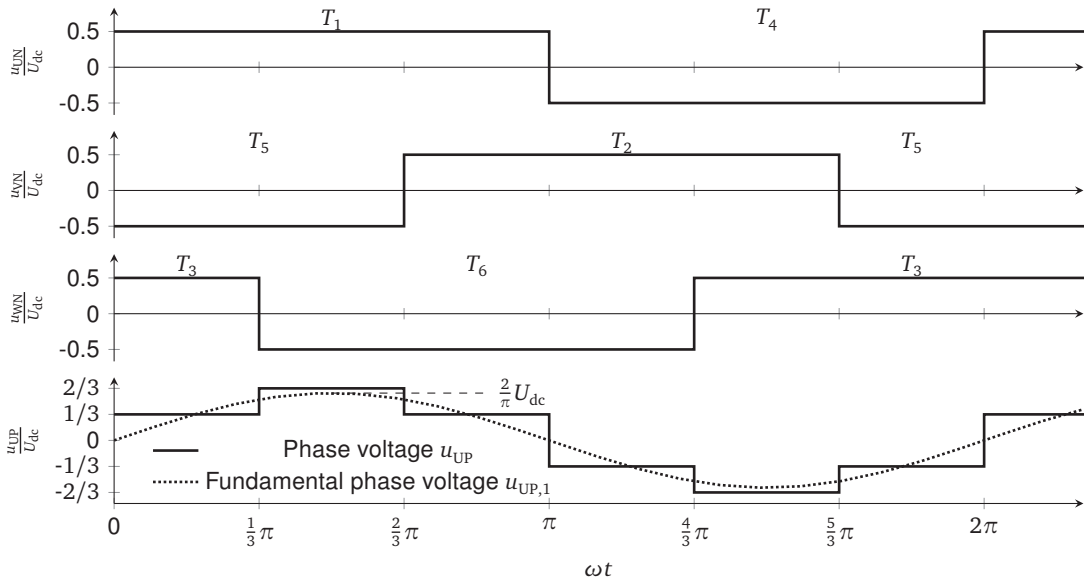


Figure 2.4: Six-step square wave inverter switching scheme and resulting pole and phase voltages. The pole voltages $u_{U/V/WN}(t)$, $u_{VN}(t)$ take the values $-0.5U_{dc}$ or $+0.5U_{dc}$. The motor phase voltages $u_{UP}(t)$, $u_{VP}(t)$, and $u_{WP}(t)$ take the form of a step function, which fundamental voltage $u_{UP,1}(t)$ is a sin-wave with an amplitude of $\frac{2}{\pi}U_{dc}$ [85, p. 278ff.].

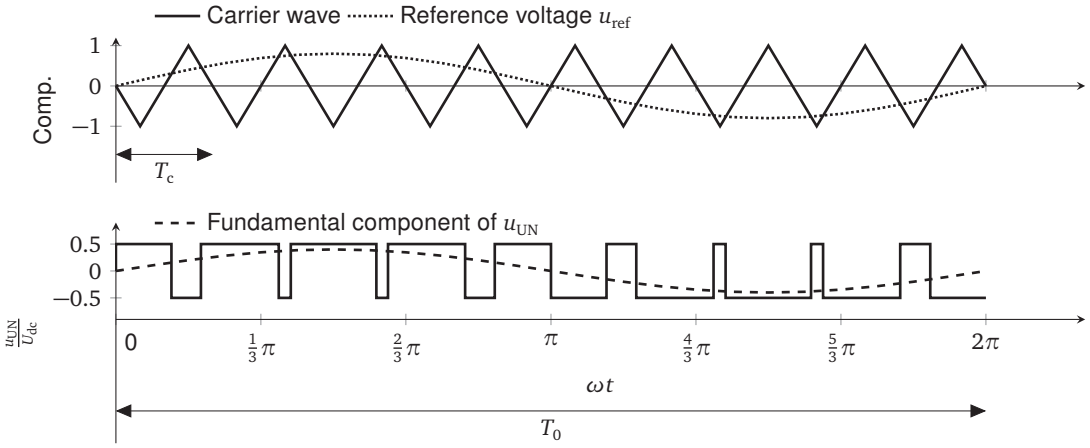


Figure 2.5: PWM switching scheme and resulting pole voltage $u_{UN}(t)$ [85, p. 292].

Therefore, the switching frequency f_{sw} for every transistor is much higher compared to the six-step square wave modulation method (Figure 2.4). The frequency modulation index m_f,

$$m_f = \frac{f_c}{f_0}, \tag{2.8}$$

describes the ratio of the carrier frequency $f_c = \frac{1}{T_c}$, which equals the switching frequency $f_{sw} = \frac{1}{T_{sw}}$ in the sinusoidal PWM method [85, p. 295], and the fundamental output voltage frequency $f_0 = \frac{1}{T_0}$.

The PWM methods have been developed to reduce generating harmonics, which are an inherent disadvantage of the simpler square wave modulation. Through this, undesired noise and heat losses inside the VSI are avoided [85, p. 273]. Conversely, PWM methods, which operate at a higher switching frequency f_{sw}, tend to increase the inverter's switching losses [85, p. 289].

Power Loss

The literature differentiates the power losses generated in a semiconductor between

- the switching losses,
- the conducting losses, and
- the leakage losses,

where the latter are negligible [90, p. 925], [87]. To further analyze the switching and the conducting losses, Figure 2.6 depicts the collector current $i_c(t)$ and the collector-emitter voltage $u_{ce}(t)$ of a transistor during a single switching period of duration T_{sw} [87]. The symbols $\hat{I}_c$ and $\hat{U}_{ce}$ denote the corresponding peak values. Rising or falling edges of the gate-emitter voltage $u_{ge}(t)$, which stems from an external control signal, trigger the measurements of the turn-on and turn-off times $t_{on/off}$, respectively. Therein, the rise time t_r denotes the time span between reaching $10\,\%$ of the maximum stationary value of the collector current (I_l) and of the collector-emitter voltage (U_{dc}) during the IGBT turn-on process. Accordingly, the fall time t_f is defined as the duration between reaching $90\,\%$ and $10\,\%$ of the stationary collector current I_l during the turn-off process. Additionally, the conduction time t_c lies in between the end of t_r and the beginning of t_f.

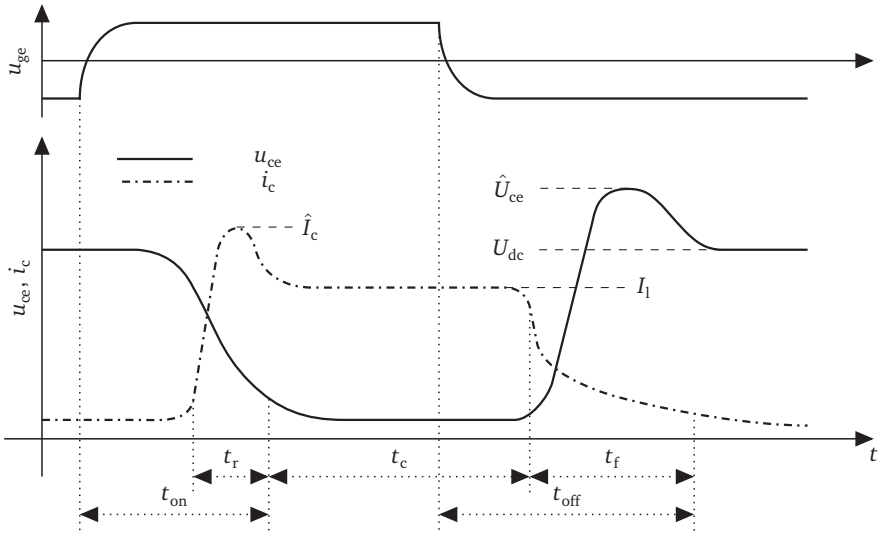


Figure 2.6: IGBT collector current $i_c(t)$ and collector-emitter voltage $u_{ce}(t)$ during a single switching period including their corresponding peak values [81, 87].

Switching losses in an IGBT occur when turning the semiconductor on and off, hence this power loss is directly proportional to the switching frequency when operated by one of the PWM methods [91, p. 82] [87, 92],

$$P_{sw} = \frac{1}{T_{sw}} \left(\underbrace{\int_0^{t_r} i_{c,r}(t) u_{ce,r}(t) \mathrm{d}t}_{W_r} + \underbrace{\int_0^{t_f} i_{c,f}(t) u_{ce,f}(t) \mathrm{d}t}_{W_f} \right), \tag{2.9}$$

where the indices r (rising) and f (falling) denote the signals during turn-on and turn-off, respectively.

To enable a reverse power flow in the VSI half-bridges, external freewheeling diodes are necessary, causing a reverse recovery loss [93] [90, p. 925]. When the collector current $i_c(t)$ of

the switching-on IGBT reaches the load current I_l, the opposing diode in one VSI half bridge starts turning-off [87, 94, 95]. Thereby, the diode releases a reverse recovery current, which results in the collector current peak value $\hat{I}_\mathrm{c}$ (Figure 2.6). In total, the reverse recovery effect causes power losses in the transistor by increasing the collector current $i_\mathrm{c,r}(t)$. Additionally, the excess power P_rr is consumed by the diode itself [87, 95], which adds up to the VSI power losses:

$$P_\mathrm{rr} = \frac{1}{T_\mathrm{sw}} \underbrace{\int_0^{t_\mathrm{rr}} i_\mathrm{d,rr}(t) u_\mathrm{d,rr}(t) \mathrm{d}t}_{W_\mathrm{rr}} \approx \frac{1}{4} \hat{I}_\mathrm{rr} U_\mathrm{dc} t_\mathrm{rr} f_\mathrm{sw}, \tag{2.10}$$

where $i_\mathrm{d,rr}(t)$ and $u_\mathrm{d,rr}(t)$ denote the current and the voltage of the freewheeling diode during reverse recovery. The symbol $\hat{I}_\mathrm{rr}$ indicates the maximum reverse recovery current, and t_rr the time duration of the recovery effect.

When using datasheet numbers, the dissipating energies $W_\mathrm{r/f/rr}$ are given for a typical operating point [93]:

$$P_\mathrm{sw} = (W_\mathrm{r} + W_\mathrm{f} + W_\mathrm{rr}) f_\mathrm{sw}. \tag{2.11}$$

The conducting losses can be written as [81]:

$$P_\mathrm{c} = \frac{1}{T_\mathrm{sw}} \int_0^{t_\mathrm{c}} u_\mathrm{ce,c}(t) i_\mathrm{c,c}(t) \mathrm{d}t \tag{2.12}$$

where $u_\mathrm{ce,c}(t)$ and $i_\mathrm{c,c}(t)$ denote the collector-emitter voltage drop and the emitter current in the on-state operation of the IGBT, i.e., when it is conducting for the time duration of t_c.

The total inverter losses $P_\mathrm{l,I}$ are calculated by

$$P_\mathrm{l,I} = P_\mathrm{sw,I} + P_\mathrm{c,I}, \tag{2.13}$$

where the total switching and conduction losses $P_\mathrm{sw/c,I}$ are determined by scaling the single IGBT losses [87, 96]. By this, the conduction losses turn out to be independent of the switching frequency in the VSI [92].

In order not to damage the semiconductors or risk a limited performance, their temperature measurement and surveillance is crucial [97, p. 245]. The maximum temperature for a VSI operating a permanent magnet synchronous machine (PMSM) is around $120\,^\circ$C [97, p. 459].

2.1.3 Electric Motor

Electric motors can be divided into two main groups [97, p. 122], which are the brushed and the brushless motors (Figure 2.7). Thereby, brushed motors are not relevant for automotive powertrains, since brushless motors combine a higher average efficiency, require less maintenance, and are outstandingly robust [97, p. 135]. The brushless versions can further be subdivided according to the ratio of the stator's electrical and the rotor's mechanical speed, which can either be synchronous or asynchronous. The rotors in asynchronous machines (ASMs) are built in a wound or squirrel-cage design. The synchronous machines can either be commutated by trapezoidal DC power (brushless direct current (BLDC) machines) or by sinusoidal AC power (in

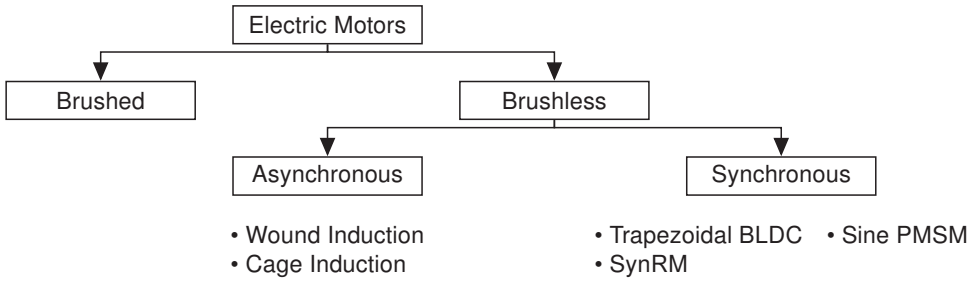


Figure 2.7: Overview of electric motors [97, p. 122]. A further subdivision of the brushed machines is omitted.

the case of PMSMs). An additional construction method are synchronous reluctance machines (SynRMs), which produce torque based on the reluctance principle [97, p. 180].

Especially in racing applications PMSMs are chosen due to their torque-per-weight ratio [98], and their high efficiency level [99].

Working Principle

This section explains the working principle and the control of a three-phase PMSM, which is connected to a VSI through its UVW motor phases as visualized in Figure 2.8.

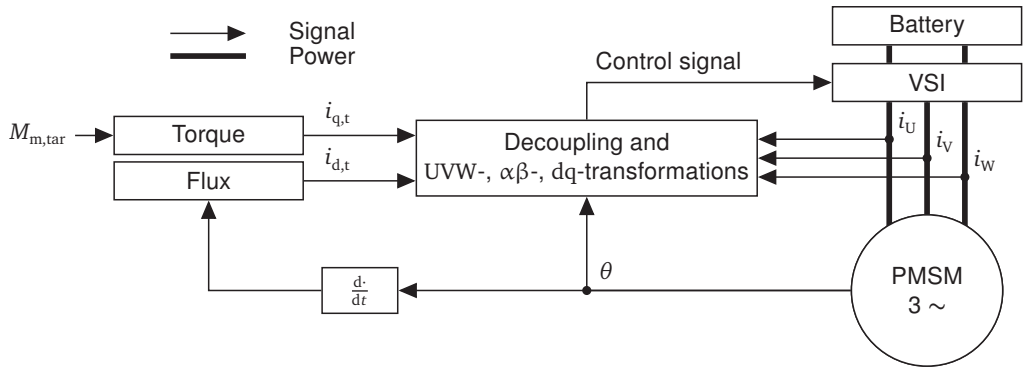


Figure 2.8: Working principle of a modern field-oriented control (FOC) method for the control of a three-phase (3 ∼) PMSM [100, p. 11].

In a vehicle, the motor output torque $M_{\mathrm{m}}(t)$ should follow the target torque $M_{\mathrm{m,tar}}(t)$, where

$$J_{\mathrm{m}} \frac{\mathrm{d}\omega_{\mathrm{m}}(t)}{\mathrm{d}t} = M_{\mathrm{m}}(t) - M_{\mathrm{l}}(t), \tag{2.14}$$

with J_{m} denoting the rotor inertia, $\omega_{\mathrm{m}}(t)$ the mechanical rotational speed, and $M_{\mathrm{l}}(t)$ the time-dependent motor load [101, p. 138]. The torque $M_{\mathrm{m}}(t)$ produced by a PMSM can be written as

$$M_{\mathrm{m}}(t) = \frac{3}{2} z_{\mathrm{p}} \left(\underbrace{\psi_{\mathrm{P}} i_{\mathrm{q}}(t)}_{\text{main}} + \underbrace{i_{\mathrm{d}}(t) i_{\mathrm{q}}(t) \left(L_{\mathrm{d}} - L_{\mathrm{q}} \right)}_{\text{reaction}} \right) \approx \frac{3}{2} z_{\mathrm{p}} \psi_{\mathrm{P}} i_{\mathrm{q}}(t), \tag{2.15}$$

where z_{p} denotes the number of pole pairs, and ψ_{P} a constant pole flux, which can be regarded to be a manufacturer-given system parameter [100, p. 86]. The inductances $L_{\mathrm{d/q}}$ and the current values $i_{\mathrm{d/q}}(t)$ are expressed in a virtual rotor-fixed dq-field-coordinate system. The current signals

$i_{\mathrm{d/q}}(t)$ are computed through a transformation of the stator-fixed UVW motor phase signals via the Clarke ($\alpha\beta$)- [102] and Park (dq)-equations [103]. These conversions simplify the control of an electric motor, since the resulting dq-signals are quasi-constant over time.

The torque $M_{\mathrm{m}}(t)$ comprises the main and the reaction torque. In the case of a non-salient PMSM, which have similar d- and q-inductances ($L_{\mathrm{d}} \approx L_{\mathrm{q}}$ [101, p. 137], [104, chap. 17-3], [105]) the reaction torque contribution is negligible, since the current $i_{\mathrm{d}}(t)$ does not influence the output torque $M_{\mathrm{m}}(t)$ [106]. As a result, the mechanical motor torque $M_{\mathrm{m}}(t)$ is directly proportional to the quadrature current $i_{\mathrm{q}}(t)$ [100, p. 11] [101, p. 138]. The PMSMs empowering the DevBot 2.0 are manufactured with surface-mounted magnets and a wet-wound pre-preg carbon fibre epoxy as a magnet retention sleeve type. They can be classified to be non-salient, i.e., they are “magnetically round” [107, p. 25], [104, chap. 17].

To vary the UVW motor phase currents, a control method like the FOC is necessary, which is depicted in Figure 2.8. The PMSM is fed by a three-phase current $i_{\mathrm{U/V/W}}(t)$, which is converted from the battery DC power by the VSI. Accordingly, a control signal, which can either be a square wave or a PWM signal, switches the VSI transistors. It imprints the target voltages $u_{\mathrm{U/V/W,t}}(t)$ on the motor’s phases which, in turn, result in the UVW-currents $i_{\mathrm{U/V/W}}(t)$. The control signal is calculated taking into account the measured currents $i_{\mathrm{U/V/W}}(t)$, the rotor angle $\theta(t)$, and the target currents $i_{\mathrm{d/q,t}}(t)$ in the virtual, rotor-fixed dq-field-coordinate system. After the Clarke- and Park-transformations of the stator-fixed UVW signals the error between the actual values $i_{\mathrm{d/q}}(t)$ and their targets $i_{\mathrm{d/q,t}}(t)$ can be computed. Finally, using this error, the target voltages $u_{\mathrm{d/q,t}}(t)$ are calculated after decoupling the following first-order ODEs, that are valid for PMSMs [100, p. 85], [101, p. 137]:

$$\frac{\mathrm{d}i_{\mathrm{d}}(t)}{\mathrm{d}t} = -\frac{R_{\mathrm{s}}}{L_{\mathrm{d}}} i_{\mathrm{d}}(t) + \frac{L_{\mathrm{q}}}{L_{\mathrm{d}}} \omega(t) i_{\mathrm{q}}(t) + \frac{1}{L_{\mathrm{d}}} u_{\mathrm{d}}(t), \tag{2.16}$$

$$\frac{\mathrm{d}i_{\mathrm{q}}(t)}{\mathrm{d}t} = -\frac{L_{\mathrm{d}}}{L_{\mathrm{q}}} \omega(t) i_{\mathrm{d}}(t) - \frac{R_{\mathrm{s}}}{L_{\mathrm{q}}} i_{\mathrm{q}}(t) + \frac{1}{L_{\mathrm{q}}} u_{\mathrm{q}}(t) - \frac{\psi_{\mathrm{p}}}{L_{\mathrm{q}}} \omega(t). \tag{2.17}$$

Here, R_{s} denotes the stator resistance, and $\omega = z_{\mathrm{p}} \omega_{\mathrm{m}}$ the electrical rotational speed. After an inverse transformation of the target voltages $u_{\mathrm{d/q,t}}(t)$ to the UVW-system they are converted to the VSI control signal, which imprints the target voltages on the motor phases.

The induced voltage in the motor increases proportionally with its rotational speed $\omega_{\mathrm{m}}(t)$ [104, chap. 17-3]. Therefore, the back electromotive force will limit the current flow into the motor at its rated rotational speed, since the total output voltage of the VSI is technically limited. Therefore, to further increase the rotational speed $\omega_{\mathrm{m}}(t)$ in the field weakening range, two measures are necessary to adhere to the VSI’s voltage constraint in a vector diagram. First, a negative direct current $i_{\mathrm{d}}(t)$ must be applied to enable the compensation of the induced voltage [101, p. 116]. Second, the quadrature current $i_{\mathrm{q}}(t)$ must be reduced by the factor $\frac{1}{\omega_{\mathrm{m}}}$ in the motor’s power limited region [101, p. 117]. A summary of the motor working principle is given in Figure 2.9, where the torque and power outputs, and the dq-current inputs at maximum power are visualized qualitatively.

In a steady-state operation, one can approximate the output torque $M_{\mathrm{m}}(t)$ by the formula [104, chap. 8]

$$T_{\mathrm{m}}(t) = k_{\mathrm{T}} \hat{i}_{\mathrm{s}}(t), \tag{2.18}$$

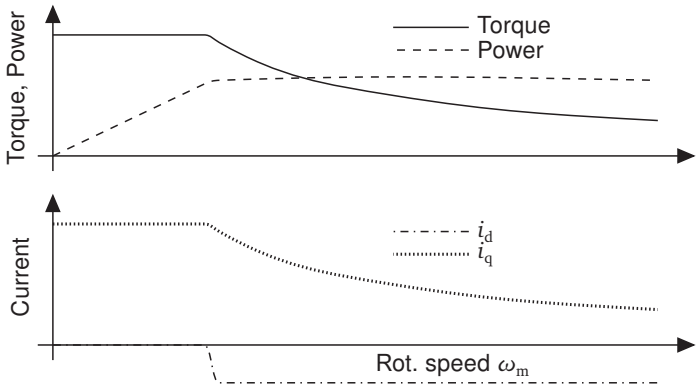


Figure 2.9: Qualitative visualization of the working principle of an electric motor. Depicted are the torque and power outputs, and the dq current inputs.

where k_T is the PMSM's torque constant, and $\hat{i}_s(t)$ is the amplitude of the stator current [108, p. 87].

Power Loss

Three main types of power losses occur in electric motors [109, p. 181], [110], [111, p. 41f.], which are the

- copper losses $P_{l,M,cop}(t)$,
- iron or core losses $P_{l,M,cre}(t)$,
- mechanical losses $P_{l,M,mec}(t)$,

whereby the major part is formed by the copper and iron losses [112], [109, p. 181]. The core losses $P_{l,M,cre}(t)$ can be subdivided into hysteresis and eddy-current losses [105, 113], the mechanical losses $P_{l,M,mec}(t)$ comprise bearing and windage losses.

To study the power losses $P_{l,M}(t)$ in an electric motor, the finite element method (FEM) has attracted growing attention. Many studies that devise analytical formulations to describe the power losses spatially resolved [114–116], or explain the effects of machine design parameters [117], use FEM to validate their results.

To describe the motor losses on a temporal level, equivalent circuits, modeling the electrical behavior of the rotor and the stator, are used. These models allow for the analytical deduction of the governing equations to describe the different types of power losses. Leveraging this approach for PMSMs leads to the following equations [92, 113, 118]:

$$P_{l,M,cop}(t) = \frac{3}{2} R_s \left(i_d^2(t) + i_q^2(t) \right) = \frac{3}{2} R_s \left[\left(\frac{\psi_d(t) - \psi_P}{L_d} \right)^2 + \left(\frac{M_m(t)}{\frac{3}{2}\left[\frac{\psi_P}{L_q} + \left(L_d - L_q \right) \frac{\psi_d(t) - \psi_P}{L_d L_q} \right]} \right)^2 \right], \tag{2.19}$$

$$P_{l,M,cre}(t) = \frac{3}{2} \frac{\omega^2(t)}{R_{Fe} + R_P} \left(\psi_q^2(t) + \psi_d^2(t) \right) = \frac{3}{2} \frac{\omega^2(t)}{R_{cre}} \left[\left(L_q i_q(t) \right)^2 + \left(\psi_P + L_d i_d(t) \right)^2 \right]. \tag{2.20}$$

Here, $\psi_{d/q}(t)$ denotes the pole flux in d- and q-directions. The iron R_{Fe} and magnet resistance R_P add up to the equivalent core resistance R_{cre}. In general, the core losses $P_{l,M,cre}(t)$ are

complicated to predict in comparison to the copper losses $P_{\mathrm{l,M,cop}}(t)$, which mainly depend on the current density [119].

Finally, the machine losses $P_{\mathrm{l,M}}(t)$ can also be approximated by measurement data. An advantage of this method is the relatively low computational complexity of the fitting equations. Still, their accuracy can be adjusted by additional nonlinear terms and the usage of real-world data, which covers physically relevant effects. According to [120], the power losses $P_{\mathrm{l,M}}(t)$ occurring in different types of electric machines can be approximated by polynomials comprising different degrees of the free variables, which are rotational speed $\omega_{\mathrm{m}}(t)$ and torque output $M_{\mathrm{m}}(t)$. In a basic approximation, the copper losses $P_{\mathrm{l,M,cop}}(t)$ are a function of the output torque $M_{\mathrm{m}}(t)$ and thus proportional to the square of the stator current $i_{\mathrm{s}}(t)$. [121, p. 94] suggest a rotational speed-dependent model to describe the iron losses $P_{\mathrm{l,M,iro}}(t)$, which can be refined in the field weakening range by adding the stator voltage to the fitting function [122].

The thermal management of electric motors becomes particularly important in high-performance applications [123]. The temperature hot-spots during designed operation are associated with the specific motor design. However, they are normally located near the motor windings, specifically the end-windings [110, 124–126], [111, p. 42]. Their energy dissipation capabilities are relatively low due to the geometrically complex embedding and additional insulating material [111, p. 41f.]. Moreover, the windings are directly exposed to the exciting current. The admissible maximum temperature values are material-dependent but can, for a PMSM used in a race application, be found in the range of $200\,^{\circ}$C for the motor windings, $170\,^{\circ}$C for the stator and the rotor, and $120\,^{\circ}$C for the permanent magnets [127].

2.2 Mathematical Optimization

In general, mathematical optimization algorithms can be classified into deterministic and stochastic ones [128, p. 20] (Figure 2.10). Prominent deterministic examples are the gradient [129, p. 466] or steepest descent methods [129, p. 475]. Stochastic techniques comprise heuristics and metaheuristics. The metaheuristics combine stochasticity with a local search method and are subdivided into population- or trajectory-based methods [128, p. 21]. Examples of stochastic algorithms are the genetic algorithm (GA), the particle swarm optimization (PSO), and simulated annealing (SA) [130].

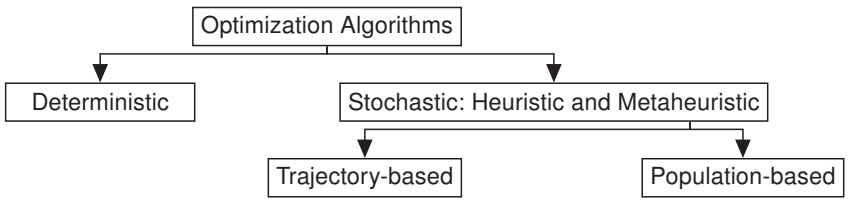


Figure 2.10: Classification of optimization algorithms [128, p. 21].

Stochastic algorithms should be considered if the governing functions are discontinuous and non-differentiable, multimodal, or if a high amount of numerical noise exists [131, p. 141]. (Meta)-heuristics can be used flexibly, since they do not require the optimization problem to exhibit a specific mathematical format [132]. However, these methods are computationally expensive and perform, in general, poorly on large-scale problems [133], since the probability of finding a globally optimal solution decreases with increasing problem size [132]. Therefore, their solutions also lack global optimality guarantees.

Deterministic methods leverage mathematical properties of the formulated optimization problem [132]. They rely on the problem's gradients, which can be analytically devised or accurately approximated in many engineering optimization problems [134, p. vii], [133]. In the following, the characteristics of deterministic methods will be listed. First, they exploit the mathematical structure of the underlying optimization problem and are therefore computationally efficient [131, p. 141]. Moreover, they provide global optimality guarantees when applied to convex optimization problems like, e.g., linear programs (LPs) or positive semidefinite quadratic programs (QPs) [129, p. 138]. Second, their solutions are smooth [135] and adhere to the user-specified constraints if the problem is feasible, which cannot be assured for, e.g., GAs [136, 137], since the constraints must be integrated into their objective function or the solution candidates be repaired. Third, a solver initialization can significantly speed up the iterative procedure, especially in MPC-like applications where local approximations of the theoretical global solution are sufficient or real time requirements must be met [138, p. 571f.]. Fourth, deterministic algorithms converge in a finite amount of iterations when applied to specific types of mathematical programming problems [139, p. 477]. Finally, they require little problem-specific parameter tuning [133].

In the remainder of this section we will focus on deterministic algorithms, since we will leverage them to solve OCPs in a direct fashion in the main part of this thesis. We will start by introducing the necessary and sufficient criteria to assess the solution quality of deterministic algorithms.

2.2.1 Optimality Criteria

A vector $\boldsymbol{o}^*$ is said to be an unconstrained global minimum of the objective function $J : \mathbb{R}^n \mapsto \mathbb{R}$ if [140, p. 6]

$$J(\boldsymbol{o}^*) \leq J(\boldsymbol{o}), \quad \forall \boldsymbol{o} \in \mathbb{R}^n, \tag{2.21}$$

where $\boldsymbol{o} \in \mathbb{R}^n$ defines the vector of optimization variables of dimension n.

A global algorithm will find the best possible solution $\boldsymbol{o}^* \in \mathbb{R}^n$ of a nonlinear program (NLP) independently of its initialization. An example of such an algorithm is the Shubert-Piyavskii method [141, 142]. The drawback of this algorithm is that it requires knowing or estimating an appropriate Lipschitz constant to obtain accurate results [143, p. 46]. In a similar approach, called divided rectangles (DIRECT) [144], specifying the Lipschitz constant is not necessary. However, the original formulation of DIRECT only handles upper and lower bounds on the state variables. Moreover, its convergence rate towards a high-accuracy solution is low, since it does not exploit local characteristics of the governing equations to speed up the optimization process. In addition, its computational effort increases significantly with the number of optimization variables [145].

Due to the backdrops of global algorithms, and to solve practical non-convex NLPs in real time, a compromise between finding the globally best solution and efficiency must be obtained [129, p. 10]. Therefore, many well-studied local algorithms are available, which are widely used in practical applications where finding a "good point" is sufficient [129, p. 9], [138, p. 147]. To validate the solution quality of a local algorithm compared to the global optimum, [140, p. 231 f.] suggests restarting the algorithms from a broad variety of initial points. Additionally, different solver strategies can be implemented to check for improvements in the result quality.

The vector $\boldsymbol{o}^*$ is an unconstrained local minimum of J if

$$J(\boldsymbol{o}^*) \leq J(\boldsymbol{o}), \quad \forall \boldsymbol{o} \in \mathbb{R}^n \text{ with } \left\| \boldsymbol{o} - \boldsymbol{o}^* \right\| < \epsilon, \tag{2.22}$$

and a constrained local minimum if

$$J(\boldsymbol{o}^*) \leq J(\boldsymbol{o}), \quad \forall \boldsymbol{o} \in O \text{ with } \left\| \boldsymbol{o} - \boldsymbol{o}^* \right\| < \epsilon, \tag{2.23}$$

where $O \subset \mathbb{R}^n$ is a constraint set, and ϵ the neighborhood of $\boldsymbol{o}^*$ [140, p. 5 f.].

Conditions for Unconstrained Problems

In an unconstrained NLP,

$$\min_{\boldsymbol{o} \in \mathbb{R}^n} J(\boldsymbol{o}), \tag{2.24}$$

the candidates for a local optimum, i.e., the "stationary points" [140, p. 8], can be determined by the first-order necessary condition (FONC) [140, p. 15],

$$\nabla J(\boldsymbol{o}^*) = 0, \tag{2.25}$$

which was originally discovered by Pierre de Fermat in 1637 [146]. If the objective function $J(\boldsymbol{o})$ is convex, the FONC is not only necessary but also sufficient to determine optimality [140, p. 8]. To further distinguish whether the stationary point $\boldsymbol{o}^*$ describes a local minimum, maximum, or stationary point of the nonlinear function $J(\boldsymbol{o})$, the second-order necessary condition (SONC) must be evaluated [140, p. 15]:

$$\nabla^2 J(\boldsymbol{o}^*) : \text{positive semidefinite.} \tag{2.26}$$

Figure 2.11 depicts an edge case where both, the FONC and the SONC, will fail to determine a maximum or minimum correctly, since they interpret inflection points as optima.

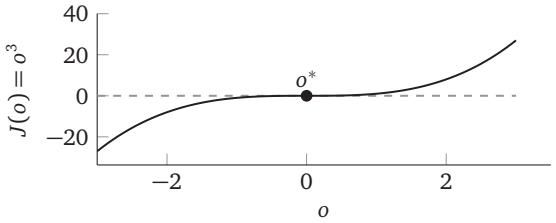


Figure 2.11: At the inflection point o^* both, the FONC and the SONC are fulfilled: $\nabla J(o^*) = 3(o^*)^2 = 3 \cdot 0^2 = 0$, and $\nabla^2 J(o^*) = 6o^* = 0$. Nevertheless, o^* does not depict a minimum or a maximum [140, p. 9].

Hence, to reliably detect a local optimum of the nonlinear function $J(\boldsymbol{o})$, the second-order sufficient condition (SOSC) must be fulfilled. It basically extends the FONC by the following sufficient condition [140, p. 21]:

$$\nabla^2 J(\boldsymbol{o}^*) : \text{positive definite.} \tag{2.27}$$

Conditions for Constrained Problems

In this section we will extend the FONC (2.25), and the SOSC (2.27) for constrained optimization problems. In general, a constrained NLP can be written as [134, p. 215]

$$\min_{\boldsymbol{o} \in \mathbb{R}^n} J(\boldsymbol{o}) \tag{2.28a}$$
$$\text{s.t. } h_1(\boldsymbol{o}) = 0, ..., h_m(\boldsymbol{o}) = 0 \tag{2.28b}$$
$$g_1(\boldsymbol{o}) \leq 0, ..., g_r(\boldsymbol{o}) \leq 0, \tag{2.28c}$$

comprising m equality and r inequality constraints. More compactly, the formulation is summarized to [140, p. 377]

$$\min_{\boldsymbol{o} \in \mathbb{R}^n} J(\boldsymbol{o}) \tag{2.29a}$$
$$\text{s.t. } \boldsymbol{H}(\boldsymbol{o}) = 0 \tag{2.29b}$$
$$\boldsymbol{G}(\boldsymbol{o}) \leq 0, \tag{2.29c}$$

where $\boldsymbol{H}(\boldsymbol{o}) = (h_1(\boldsymbol{o}), ..., h_m(\boldsymbol{o})) : \mathbb{R}^n \mapsto \mathbb{R}^m$ and $\boldsymbol{G}(\boldsymbol{o}) = (g_1(\boldsymbol{o}), ..., g_r(\boldsymbol{o})) : \mathbb{R}^n \mapsto \mathbb{R}^r$. To obtain candidates $\boldsymbol{o}^*$ solving this NLP analytically, one must search for an optimal set which fulfills the first-order necessary Karush-Kuhn-Tucker (KKT) conditions [134, p. 197]. They can be written as [134, p. 258], [140, p. 379]

$$\nabla_{\boldsymbol{o}} \mathcal{L}(\boldsymbol{o}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \nabla J(\boldsymbol{o}^*) + \sum_{i=1}^{m} \lambda_i^* \nabla h_i(\boldsymbol{o}^*) + \sum_{j=1}^{r} \mu_j^* \nabla g_j(\boldsymbol{o}^*) = 0 \tag{2.30a}$$
$$h_i(\boldsymbol{o}^*) = 0 \tag{2.30b}$$
$$g_j(\boldsymbol{o}^*) \leq 0 \tag{2.30c}$$
$$\mu_j^* \geq 0 \tag{2.30d}$$
$$\mu_j^* g_j(\boldsymbol{o}^*) = 0, \tag{2.30e}$$

where $\boldsymbol{o}$ are also called the "primal" variables and $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$ the "dual" variables or the Langrange multipliers. The Lagrangian function $\mathcal{L}$ is defined as [138, p. 543], [147, p. 292]

$$\mathcal{L}(\boldsymbol{o}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = J(\boldsymbol{o}) + \boldsymbol{\lambda}^T \boldsymbol{H}(\boldsymbol{o}) + \boldsymbol{\mu}^T \boldsymbol{G}(\boldsymbol{o}). \tag{2.31}$$

To fulfill the KKT conditions (2.30) a solution must be stationary (2.30a). Additionally, primal ((2.30b), (2.30c)) and dual feasibility (2.30d) are required. Moreover, either the inequality constraints g_j must be active, i.e., hold with equality at the optimal solution $\boldsymbol{o}^*$ or, if the inequality constraints g_j are not active, the corresponding Lagrange multipliers μ_j^* must be zero [143, p. 176 f.]. The complementary slackness condition (2.30e) enforces this behavior and can compactly be written as

$$\left.\begin{aligned} &\mu_j^* \geq 0 \\ &\mu_j^* g_j(\boldsymbol{o}^*) = 0 \end{aligned}\right\} \mu_j^* \geq 0 \perp g_j(\boldsymbol{o}^*) \leq 0. \tag{2.32}$$

Strict complementarity is achieved if

$$\mu_j^* = 0, \; g_j(\boldsymbol{o}^*) \neq 0 \tag{2.33}$$

or

$$\mu_j^* \neq 0,\ g_j(\boldsymbol{o}^*) = 0 \tag{2.34}$$

holds and no combination of $\mu_j^* = 0,\ g_j(\boldsymbol{o}^*) = 0$ exists [148, p. 11]. In general, complementarity conditions are challenging for a numerical optimization solver, since they require solving an underlying combinatorial problem [139, p. 424]. To handle inequalities in an NLP, the solver determines whether they actively constrain the optimal solution $\boldsymbol{o}^*$. If not, they can be excluded from the optimization problem via the Lagrange multipliers.

To guarantee whether the candidate $\boldsymbol{o}^*$ depicts a local optimum, the KKT conditions (2.30) need to be extended by [140, p. 383]

$$\boldsymbol{y}^T \nabla^2_{\boldsymbol{oo}} \mathcal{L}(\boldsymbol{o}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \boldsymbol{y} > 0, \quad \text{for all } \boldsymbol{y} \neq 0 \text{ such that} \tag{2.35a}$$

$$\nabla h_i(\boldsymbol{o}^*)^T \boldsymbol{y} = 0 \tag{2.35b}$$

$$\nabla g_j(\boldsymbol{o}^*)^T \boldsymbol{y} = 0, \quad \forall j \in A(\boldsymbol{o}^*), \tag{2.35c}$$

resulting in the SOSCs for equality- and inequality-constrained problems [140, p. 383]. Here, $A(\boldsymbol{o}^*)$ denotes the set of active constraints, i.e., the inequalities $g_j(\boldsymbol{o}^*)$ which hold with equality at the optimal solution $\boldsymbol{o}^*$. The vector $\boldsymbol{y}$, which has to be orthogonal to the gradients of the equality and the active inequality constraints, is used to determine the positive definiteness of the Hessian matrix of the problem's Lagrangian $\nabla^2_{\boldsymbol{oo}} \mathcal{L}(\boldsymbol{o}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$.

2.2.2 Numerical Algorithms

Solving the KKT conditions analytically can result in large systems of equations and a massive combinatorial search [134, p. 287], [139, p. 424], since the activity of the inequality constraints g_j at the optimum $\boldsymbol{o}^*$ must be determined. Therefore, iterative algorithms for constrained NLPs have emerged, which will be presented in the remainder of this section.

Table 2.2 organizes methods, frameworks, and solvers, which have been developed for general NLPs or specifically for QPs.

Table 2.2: Iterative numerical optimization algorithms.

	NLP		
Method	*IP*		*SQP*
Framework	CasADi		acados
Solver	IPOPT		Any QP solver
	QP		
Method	*Active-set*	*ADMM*	*IP*
Solver	qpOASES	OSQP	HPIPM

To solve NLPs, two widely used Newton-type methods have been developed, which are the nonlinear interior point (IP) and the sequential quadratic program (SQP) methods [138, p. 551], [148, p. 14]. To formulate NLPs a framework, which supports symbolic expressions and directly interfaces the optimization solver, is helpful. Therefore, we further introduce the row "Framework", which comprises both CasADi [149] and acados [150, 151]. An IP method can be used via CasADi in combination with, e.g., the interior point optimizer (IPOPT) solver [152]. Acados, which

was primarily developed for nonlinear model predictive control (NMPC) applications, implements an SQP method and interfaces a broad variety of standard QP solvers.

In particular, QPs can be flagged up as infeasible or be solved in a finite amount of iterations. The absolute number of iterations depends on the mathematical form of the objective function and the number of inequality constraints [139, p. 449]. This advantage is especially helpful in real-time-critical embedded systems.

NLP Methods

In this section two well-known NLP solution methods will be introduced.

NLP - Interior Point Methods

The following paragraph considers the general NLP problem (2.29) and explains the general principles of nonlinear IP methods. To mitigate the combinatorial difficulty of identifying the active inequality constraints at the optimal solution $\boldsymbol{o}^*$, a barrier parameter ϵ is introduced,

$$0 < \epsilon^{k+1} < \epsilon^k, \quad k = 0, 1, ..., \quad \epsilon^k \to 0, \tag{2.36}$$

which is used to transform the inequality constraints into penalty terms. Subsequently, these are added to the objective function [153, p. 488], [154, p.108]. Symbol k denotes the sequence iteration number [140, p. 447]. Through a logarithmic barrier function, the original problem (2.29) turns into the equality-constrained version [138, p. 553], [139, p. 424]

$$\min_{\boldsymbol{o} \in \mathbb{R}^n,\, \boldsymbol{s} \in \mathbb{R}^r} J(\boldsymbol{o}) - \epsilon^k \sum_{j=1}^{r} \ln s_j = \min_{\boldsymbol{o} \in \mathbb{R}^n,\, \boldsymbol{s} \in \mathbb{R}^r} \tilde{J}(\boldsymbol{o}, \boldsymbol{s}, \epsilon) \tag{2.37a}$$

$$\text{s.t. } \boldsymbol{H}(\boldsymbol{o}) = 0 \tag{2.37b}$$

$$\boldsymbol{G}(\boldsymbol{o}) + \boldsymbol{s} = 0. \tag{2.37c}$$

In (2.37) the slack variables $\boldsymbol{s} > 0$ transform the inequality constraints $\boldsymbol{G}$ into equalities. The logarithmic barrier function $-\sum \ln s_j$ enforces adhering to the inequality constraints $g_j(\boldsymbol{o})$. For increasing values of $\boldsymbol{s}$ the smaller the objective function value $\tilde{J}(\boldsymbol{o}, \boldsymbol{s}, \epsilon)$ becomes. For $\boldsymbol{s} \to 0$, i.e., when the inequality constraints $\boldsymbol{G}$ tend towards their upper bound, the objective function grows to infinity.

The format of the KKT-conditions for problem (2.37) differs slightly compared to that of the original problem in (2.30). The major difference is that the complementarity condition $\mu_j^* g_j(\boldsymbol{o}^*) = 0$ (2.30e) is replaced by the smooth version $\mu_j^* s_j^* = \epsilon^k$ [138, p. 553]:

$$\nabla J(\boldsymbol{o}^*) + \sum_{i=1}^{m} \lambda_i^* \nabla h_i(\boldsymbol{o}^*) + \sum_{j=1}^{r} \mu_j^* \nabla g_j(\boldsymbol{o}^*) = 0 \tag{2.38a}$$

$$h_i(\boldsymbol{o}^*) = 0 \tag{2.38b}$$

$$g_j(\boldsymbol{o}^*) + s_j^* = 0 \tag{2.38c}$$

$$\mu_j^* \geq 0 \tag{2.38d}$$

$$\mu_j^* s_j^* = \epsilon^k. \tag{2.38e}$$

The demonstrated principle of an IP algorithm is implemented in the well-known solver IPOPT. Many other variants of nonlinear IP algorithms exist, which can also be tailored to linear or quadratic programs, see for example [140, p. 447], [153, p. 488].

NLP - Sequential Quadratic Programming Methods

In an SQP method [155], NLP (2.29) is linearized in every iteration k resulting in an inequality-constrained QP of the form [139, p. 423]

$$\min_{\boldsymbol{z}_{\mathrm{qp}} \in \mathbb{R}^n} \frac{1}{2}\boldsymbol{z}_{\mathrm{qp}}^T \boldsymbol{P}(\boldsymbol{o}^k, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k)\boldsymbol{z}_{\mathrm{qp}} + \nabla J(\boldsymbol{o}^k)^T \boldsymbol{z}_{\mathrm{qp}} \tag{2.39a}$$

$$\text{s.t. } \nabla h_i(\boldsymbol{o}^k)^T \boldsymbol{z}_{\mathrm{qp}} + h_i(\boldsymbol{o}^k) = 0 \tag{2.39b}$$

$$\nabla g_j(\boldsymbol{o}^k)^T \boldsymbol{z}_{\mathrm{qp}} + g_j(\boldsymbol{o}^k) \leq 0. \tag{2.39c}$$

Here, $\boldsymbol{P}(\boldsymbol{o}^k, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) \in \mathbb{R}^{n \times n}$ denotes the problem's Hessian, and $\boldsymbol{o}^k$ the current linearization point. In every iteration k, the QP in (2.39) is solved. The subsequent iterate k is updated according to [155]

$$\boldsymbol{o}^{k+1} = \boldsymbol{o}^k + \alpha \boldsymbol{z}_{\mathrm{qp}} \tag{2.40a}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}_{\mathrm{qp}} \tag{2.40b}$$

$$\boldsymbol{\mu}^{k+1} = \boldsymbol{\mu}_{\mathrm{qp}}, \tag{2.40c}$$

where $\boldsymbol{z}_{\mathrm{qp}}$ denotes the search direction, $\boldsymbol{\lambda}_{\mathrm{qp}}$ and $\boldsymbol{\mu}_{\mathrm{qp}}$ the Lagrange multipliers of the QP, and α the step size, which can be computed by, e.g., the Armijo rule [153, p. 230]. The Hessian is either calculated exactly by $\nabla^2_{\boldsymbol{oo}} \mathcal{L}(\boldsymbol{o}^k, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k)$ or approximated by, e.g., the Broyden-Fletcher-Goldfarb-Shanno or the Gauss-Newton algorithm [148, p. 16].

Compared to nonlinear IP algorithms, the SQP method requires a smaller number of iterations to converge, whereby one iteration is computationally more expensive. This stems from the fact that IP methods solve one linear system per iteration resulting in a smaller computation time. SQP methods have convergence advantages in MPC-applications where an initial guess, which is close to the optimal solution of the subsequent optimization problems, is available [138, p. 554 f.], [140, p. 549].

Solving the quadratic subproblems in (2.39) requires a QP solver. In the section below we will therefore discuss the algorithmic background of three widely-used QP solver methods [156], [139, p. 422].

QP Methods

In this section three famous QP solution methods will be explained.

QP - Interior Point Methods

A convex QP has the form [156]

$$\min_{\boldsymbol{z}_{\mathrm{qp}} \in \mathbb{R}^n} \frac{1}{2}\boldsymbol{z}_{\mathrm{qp}}^T \boldsymbol{P} \boldsymbol{z}_{\mathrm{qp}} + \boldsymbol{q}^T \boldsymbol{z}_{\mathrm{qp}} \tag{2.41a}$$

$$\text{s.t. } \boldsymbol{l} \leq \boldsymbol{A}\boldsymbol{z}_{\mathrm{qp}} \leq \boldsymbol{u}, \tag{2.41b}$$

where $\boldsymbol{P} \in \mathbb{R}^{n \times n}$ is positive semidefinite, $\boldsymbol{q} \in \mathbb{R}^n$, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ contains the linear constraints and $\boldsymbol{l}, \boldsymbol{u} \in \mathbb{R}^m$ their upper and lower boundaries, with $l_i \in \{-\infty\} \cup \mathbb{R}$ and $u_i \in \{+\infty\} \cup \mathbb{R}$. For convenience, we rewrite (2.41) and split the constraints into equality and inequality constraints, indicated by the indices E and I, respectively,

$$\min_{\boldsymbol{z}_{\mathrm{qp}} \in \mathbb{R}^n} \frac{1}{2} \boldsymbol{z}_{\mathrm{qp}}^T \boldsymbol{P} \boldsymbol{z}_{\mathrm{qp}} + \boldsymbol{q}^T \boldsymbol{z}_{\mathrm{qp}} \tag{2.42a}$$

$$\text{s.t. } \boldsymbol{A}_{\mathrm{E}} \boldsymbol{z}_{\mathrm{qp}} - \boldsymbol{b}_{\mathrm{E}} = 0 \tag{2.42b}$$

$$\boldsymbol{A}_{\mathrm{I}} \boldsymbol{z}_{\mathrm{qp}} - \boldsymbol{b}_{\mathrm{I}} \geq 0, \tag{2.42c}$$

where $\boldsymbol{b}_I \in \mathbb{R}^{m_{\mathrm{I}}}$, $\boldsymbol{b}_E \in \mathbb{R}^{m_{\mathrm{E}}}$ and $m_{\mathrm{I}} + m_{\mathrm{E}} = m$.

To solve the QP (2.41) by an interior point algorithm, the FONC in the form of the KKT conditions must be deduced [129, p. 567], [138, p. 553]. Since QP (2.41) is convex, these are also sufficient to determine an optimal solution [139, p. 481]. The resulting KKT conditions look similar compared to those arising in nonlinear IP methods applied to NLPs (2.38),

$$\boldsymbol{P} \boldsymbol{z}_{\mathrm{qp}}^* + \boldsymbol{q} - \boldsymbol{A}_{\mathrm{E}}^T \boldsymbol{\lambda}^* - \boldsymbol{A}_{\mathrm{I}}^T \boldsymbol{\mu}^* = 0 \tag{2.43a}$$

$$-\boldsymbol{A}_{\mathrm{E}} \boldsymbol{z}_{\mathrm{qp}}^* + \boldsymbol{b}_{\mathrm{E}} = 0 \tag{2.43b}$$

$$-\boldsymbol{A}_{\mathrm{I}} \boldsymbol{z}_{\mathrm{qp}}^* + \boldsymbol{b}_{\mathrm{I}} + \boldsymbol{s}^* = 0 \tag{2.43c}$$

$$\left(\mu_j^*, s_j^*\right) \geq 0 \tag{2.43d}$$

$$\mu_j^* s_j^* = \epsilon^k. \tag{2.43e}$$

The KKT conditions lead to a linear system, which can effortlessly be solved to obtain Newton steps into the direction of the globally optimal solution $\boldsymbol{z}_{\mathrm{qp}}^*$.

IP methods deliver good performance across a broad range of practical problems [139, p. 480], [153, p. 487], [156]. Moreover, they can efficiently deal with a large number of inequality constraints [152]. However, IP methods cannot easily be warm-started [138, p. 555], [139, p. 485], which means that a good initial guess does not speed up the solution process as much as in active-set methods. An IP method tailored to QPs is implemented in, e.g., the open-source solver high-performance interior point method (HPIPM) [157].

QP - Active-set Methods

We consider the QP (2.41) with a quadratic objective function $J(\boldsymbol{o})$ and linear inequality constraints $g_j(\boldsymbol{o})$,

$$\min_{\boldsymbol{o} \in \mathbb{R}^n} J(\boldsymbol{o}) \tag{2.44a}$$

$$\text{s.t. } g_1(\boldsymbol{o}) \leq 0, ..., g_j(\boldsymbol{o}) \leq 0. \tag{2.44b}$$

If one identified the subset of binding inequality constraints at the optimal solution $\boldsymbol{o}^*$ correctly, the resulting equality-constrained problem would directly lead to the optimal solution $\boldsymbol{o}^*$. Therefore, the main idea behind active-set methods is to split the inequality constraints into two groups, which shall be treated as active and inactive ones in every phase of the algorithm. The FONC

for this approach can be written as [153, p. 363], [139, p. 467]

$$\nabla J(\boldsymbol{o}^*) + \sum_{j \in A} \mu_j^* \nabla g_j(\boldsymbol{o}^*) = 0 \tag{2.45a}$$

$$g_j(\boldsymbol{o}^*) = 0, \quad j \in A \tag{2.45b}$$

$$g_j(\boldsymbol{o}^*) < 0, \quad j \notin A \tag{2.45c}$$

$$\mu_j^* > 0, \quad j \in A \tag{2.45d}$$

$$\mu_j^* = 0, \quad j \notin A. \tag{2.45e}$$

Here, (2.45c) guarantees that the inactive constraints are satisfied and their corresponding Lagrange multipliers are zero (2.45e). At the same time, the Lagrange multipliers of the active constraints must be nonnegative (2.45d).

An active-set method assumes a working set W, which is a subset of the active constraints A, in each phase of the algorithm [153, Section 12.3]: This working set W spans a "working surface", on which the algorithm moves to improve the solutions of its iterations. If during the movement on the working surface new constraints become active, they are added to the working set W. Finally, if the stationarity condition (2.45a) is fulfilled within the working set W, and its Lagrangian multipliers are nonnegative, the optimal point $\boldsymbol{o}^*$ is found. However, if some of the Lagrange multipliers are negative, the corresponding constraints are removed from the working set W and the procedure is restarted.

Negative Lagrange multipliers μ_j directly indicate that the objective function $J(\boldsymbol{o})$ can further be improved by dropping the corresponding constraint from the active set W. We illustrate this principle in Figure 2.12 [153, p. 365]. In the current phase of the active-set algorithm, $\boldsymbol{o}$ denotes the minimum point of the objective function $J(\boldsymbol{o})$ which additionally adheres to the constraint $g_1(\boldsymbol{o}) = 0$. The Lagrange multiplier μ_1 is negative. The gradient of $J(\boldsymbol{o})$ suggests moving into the feasible region for further improvement. By this, $g_1(\boldsymbol{o})$ will not hold with equality, i.e., $g_1(\boldsymbol{o})$ will become inactive but will still be fulfilled.

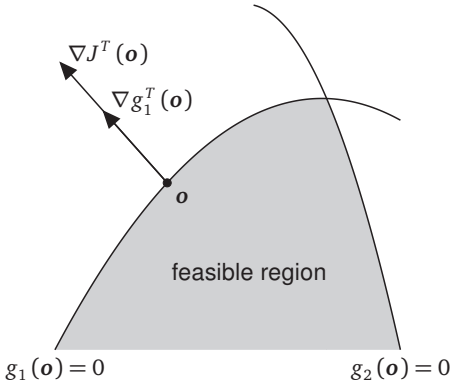


Figure 2.12: Moving into the feasible region and along the opposite direction of the gradient of the objective function $J(\boldsymbol{o})$ improves the solution. Constraint $g_1(\boldsymbol{o})$ should therefore be dropped from the working set W [153, p. 365].

Active-set methods can efficiently be warm-started to reduce the computation time compared to IP methods [139, p. 490], [156, 158]. However, in the worst case, i.e., when the algorithm has to explore all possible combinations of binding constraints, its solver time can rise enormously [139, p. 477]. Active-set methods are in general slower on large QP problems compared to IP

methods [139, p. 490], [158]. Normally, they require more iterations to converge, whereas one iteration is computationally cheaper compared to IP methods [159]. In addition, active-set algorithms can get stuck during cycling/zigzagging when repeatedly dropping and adding the same constraints to the active set W [153, p. 367], [139, p. 477]. A primal-dual active-set method is implemented in the open-source solver QP online active set strategy (qpOASES) [159]. It is particularly well-suited for, e.g., MPC applications, since the optimal solutions of subsequent optimization problems are often close to the previous ones [148, p. 32].

QP - Alternating Direction Method of Multipliers

To formulate the alternating direction method of multipliers (ADMM) we consider the general QP in (2.41). It can be rewritten to [156], [140, p. 693]

$$\min_{\boldsymbol{z}_{\mathrm{qp}} \in \mathbb{R}^n} \frac{1}{2} \boldsymbol{z}_{\mathrm{qp}}^T \boldsymbol{P} \boldsymbol{z}_{\mathrm{qp}} + \boldsymbol{q}^T \boldsymbol{z}_{\mathrm{qp}} \tag{2.46a}$$

$$\text{s.t. } \boldsymbol{A} \boldsymbol{z}_{\mathrm{qp}} = \tilde{\boldsymbol{z}} \tag{2.46b}$$

$$\tilde{\boldsymbol{z}} \in \mathcal{C}, \tag{2.46c}$$

where the convex set $\mathcal{C} = \{\tilde{\boldsymbol{z}} \in \mathbb{R}^m \mid l_i \leq \tilde{z}_i \leq u_i,\ i = 1, ..., m\}$. The augmented Lagrangian of (2.46) can be written as

$$\mathcal{L}_{\mathrm{A}}\left(\boldsymbol{z}_{\mathrm{qp}}, \tilde{\boldsymbol{z}}, \boldsymbol{\lambda}\right) = \frac{1}{2} \boldsymbol{z}_{\mathrm{qp}}^T \boldsymbol{P} \boldsymbol{z}_{\mathrm{qp}} + \boldsymbol{q}^T \boldsymbol{z}_{\mathrm{qp}} + \boldsymbol{\lambda}^T \left(\boldsymbol{A} \boldsymbol{z}_{\mathrm{qp}} - \tilde{\boldsymbol{z}}\right) + \frac{c}{2} \left\| \boldsymbol{A} \boldsymbol{z}_{\mathrm{qp}} - \tilde{\boldsymbol{z}} \right\|_2^2 \tag{2.47}$$

with $\boldsymbol{\lambda}$ representing the Lagrangian multipliers of the equality constraints, and c denoting a penalty factor. A new iterate $\left(\boldsymbol{z}_{\mathrm{qp}}^{k+1}, \tilde{\boldsymbol{z}}^{k+1}, \boldsymbol{\lambda}^{k+1}\right)$ is calculated according to the update rules [140, p. 693]

$$\boldsymbol{z}_{\mathrm{qp}}^{k+1} \in \arg\min_{\boldsymbol{z}_{\mathrm{qp}} \in \mathbb{R}^n} \mathcal{L}_{\mathrm{A}}\left(\boldsymbol{z}_{\mathrm{qp}}, \tilde{\boldsymbol{z}}^k, \boldsymbol{\lambda}^k\right) \tag{2.48a}$$

$$\tilde{\boldsymbol{z}}^{k+1} \in \arg\min_{\tilde{\boldsymbol{z}} \in \mathbb{R}^m} \mathcal{L}_{\mathrm{A}}\left(\boldsymbol{z}_{\mathrm{qp}}^{k+1}, \tilde{\boldsymbol{z}}, \boldsymbol{\lambda}^k\right) \tag{2.48b}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + c\left(\boldsymbol{A} \boldsymbol{z}_{\mathrm{qp}}^{k+1} - \tilde{\boldsymbol{z}}^{k+1}\right). \tag{2.48c}$$

This preceding procedure illustrates that the ADMM uses alternate minimizations to decouple connected variables, which makes it well-suited for parallel computing for large problems [140, p. 691]. Additionally, an ADMM provides solutions of modest accuracy in a small number of computationally cheap iterations. The method is therefore also ideally suited for practical implementations on embedded processors with limited computing power as [156] summarizes. An open-source solver which has an ADMM implemented is operator splitting quadratic program (OSQP) [156].

2.2.3 Application to Optimal Control Problems

Practical problems in engineering like the computation of a maximum velocity profile under the constraints stemming from a vehicle's driving dynamics, finding a time-optimal ES or global race trajectory for an electric race car, are OCPs. In general, OCPs can be solved by three different methods, namely dynamic programming (DP), indirect, and direct methods. Applying DP results in the Hamilton-Jacobi-Bellman equation, which suffers from the course of dimensionality after a tabulation in state space when solved numerically. Indirect methods derive a boundary value problem, which comprises differential equations that are difficult to solve due to a high degree of

nonlinearity and instability. Finally, direct methods transform the OCP into a finite-dimensional NLP [160], which can be solved with off-the-shelf algorithms. Direct methods handle inequality constraints well in comparison to indirect techniques. According to [161], [162], direct methods are the most successful methods to solve OCPs, especially in practical applications. Therefore, we will focus on them in the remainder of this thesis.

In the previous sections we have introduced the necessary and the sufficient optimality criteria and explained their specific form in different iterative optimization methods. These criteria remain valid to compute the optimal control strategies in nonlinear optimal control problems (NOCPs), since their mathematical form is directly connected to NLPs after discretization. The general, time-continuous form of an OCP can be written as [138, p. 493], [148, p. 23]

$$\min_{\boldsymbol{o}(t)} \int_0^T l(\underbrace{\boldsymbol{x}(t), \boldsymbol{u}(t)}_{\boldsymbol{o}(t)})\mathrm{d}t + l_\mathrm{f}(\boldsymbol{x}(T)) \tag{2.49a}$$

$$\text{s.t. } \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{2.49b}$$

$$\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = \dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right), \qquad t \in [0, T] \tag{2.49c}$$

$$\boldsymbol{g}\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right) \leq 0, \qquad t \in [0, T] \tag{2.49d}$$

$$\boldsymbol{g}_\mathrm{f}(\boldsymbol{x}(T)) \leq 0. \tag{2.49e}$$

In this formulation, $l : \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{u}} \mapsto \mathbb{R}$ denotes the running cost, $l_\mathrm{f} : \mathbb{R}^{n_\mathrm{x}} \mapsto \mathbb{R}$ the final stage cost, $\boldsymbol{g} : \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{u}} \mapsto \mathbb{R}^\mathrm{r}$ the path constraints, and $\boldsymbol{g}_\mathrm{f} : \mathbb{R}^{n_\mathrm{x}} \mapsto \mathbb{R}^{\mathrm{r_f}}$ the terminal constraints. The vector $\boldsymbol{f} : \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{u}} \mapsto \mathbb{R}^{n_\mathrm{x}}$ contains the system dynamics, and T marks the final time. In the following sections we will introduce several discretization techniques to transform OCPs into an NLP.

Single Shooting

Single shooting discretizes the control trajectory $\boldsymbol{u}(t)$ of an OCP by the parameters $\boldsymbol{q}_k$, which are, e.g., piecewise constant, $\boldsymbol{u}(t) = \boldsymbol{q}_k$ for $t \in [t_k, t_{k+1}]$ with $k = 0, ..., N-1$. The control input per time step k has the dimension n_q, therefore $\boldsymbol{q} \in \mathbb{R}^{N-1 \times n_\mathrm{q}}$. The system states $\boldsymbol{x}$ can then be expressed as dependent variables on the time t and the control parameters $\boldsymbol{q}$ and the governing dynamics $\boldsymbol{f}$, i.e., $\boldsymbol{x}(t; \boldsymbol{q})$, when an initial value $\boldsymbol{x}_0$ is provided [161, p. 7],

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}\,(t; \boldsymbol{q})), \quad \forall t \in [t_0, t_N] \tag{2.50a}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0. \tag{2.50b}$$

The notation $\cdot(t; \cdot)$ expresses that t is a free variable, and there are additional parameters noted after the semicolon. To numerically integrate $\boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}\,(t; \boldsymbol{q}))$, schemes like the Euler or Heun method can be applied. Different explicit and implicit Runge Kutta methods of higher order deliver more accurate and stable results at the expense of computational steps to perform [138, p. 496f.].

Figure 2.13 illustrates the single shooting approach for a problem with one system state $x(t; \boldsymbol{q})$ and a one-dimensional control input $u(t; \boldsymbol{q})$. Finally, the NLP of an OCP discretized by single

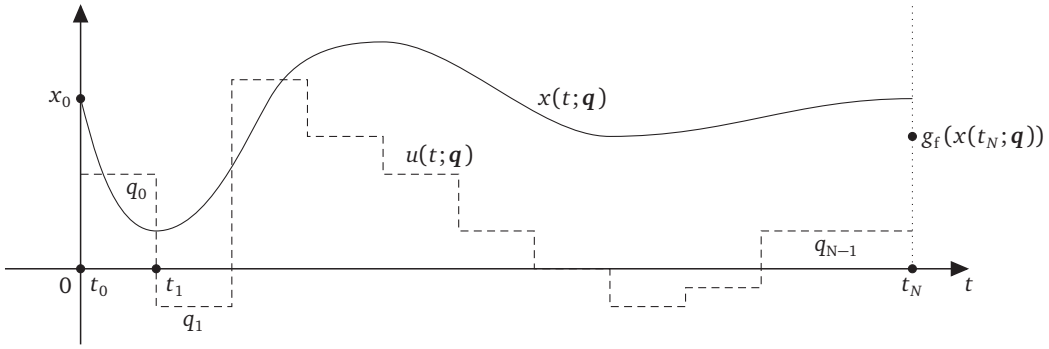


Figure 2.13: Single shooting discretization method for a problem with a single system state $x(t;\boldsymbol{q})$, a one-dimensional control input $u(t;\boldsymbol{q})$, and a terminal constraint $g_\mathrm{f}(x(t_N;\boldsymbol{q}))$ [161, p. 7].

shooting reads

$$\min_{\boldsymbol{q}\,\in\,\mathbb{R}^{N-1\times n_\mathrm{q}}} \sum_{k=0}^{N-1} l\left(\boldsymbol{x}(t_k;\boldsymbol{q}),\boldsymbol{q}_k\right) + l_\mathrm{f}(\boldsymbol{x}(t_N;\boldsymbol{q})) \tag{2.51a}$$

$$\text{s.t. } \boldsymbol{g}(\boldsymbol{x}(t_k;\boldsymbol{q}),\boldsymbol{q}_k) \leq 0, \quad k=0,\ldots,N-1 \tag{2.51b}$$

$$\boldsymbol{g}_\mathrm{f}(\boldsymbol{x}(t_N;\boldsymbol{q})) \leq 0. \tag{2.51c}$$

An advantage of single shooting are the relatively few optimization variables and constraints. Moreover, an initial guess is only necessary for the control variables, which makes this approach simple to use. On the other hand, in single shooting the user cannot provide prior knowledge of the state trajectory $\boldsymbol{x}(t;\boldsymbol{q})$. Additionally, this technique can in general not handle unstable systems [161, p. 8].

Multiple Shooting

As in single shooting, the control trajectory $\boldsymbol{u}(t)$ is parametrized by the discrete values $\boldsymbol{q}_k$, which are constant between the discretization points, $\boldsymbol{u}(t)=\boldsymbol{q}_k$ for $t\in[t_k,t_{k+1}]$. However, in multiple shooting the governing ODE system $\boldsymbol{f}$, representing the physical behavior of the controlled system, is solved independently in each of the discretization intervals. Therefore, artificial state variables $\boldsymbol{\chi}_k$ are necessary, which must be constrained to equal the integrated values of the ODEs [161, p. 10]:

$$\dot{\boldsymbol{x}}_k(t) = \boldsymbol{f}\left(\boldsymbol{x}_k(t),\boldsymbol{q}_k\right), \quad t\in[t_k,t_{k+1}] \tag{2.52a}$$

$$\boldsymbol{x}_k(t_k) = \boldsymbol{\chi}_k. \tag{2.52b}$$

By solving the initial value problem in (2.52) the trajectory pieces $\boldsymbol{x}_k\left(t;\boldsymbol{\chi}_k,\boldsymbol{q}_k\right)$ result (Figure 2.14).

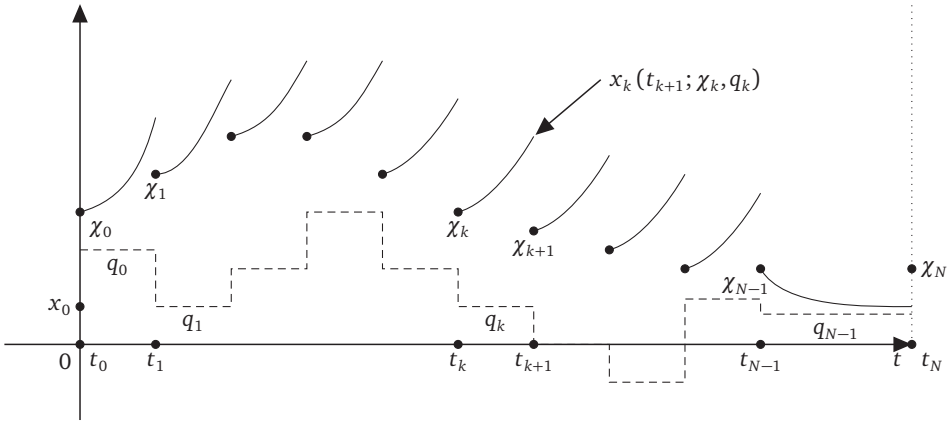


Figure 2.14: Discretization using multiple shooting for a problem with a single discretized system state $x_k(t;\chi_k,q_k)$, and a one-dimensional control input $u(t)=\boldsymbol{q}$ [161, p. 11].

The formulation of the resulting sparse NLP reads [163]

$$\min_{\boldsymbol{\chi}\in\mathbb{R}^{N\times n_x},\ \boldsymbol{q}\in\mathbb{R}^{N-1\times n_q}} \sum_{k=0}^{N-1} l_k\left(\boldsymbol{\chi}_k,\boldsymbol{q}_k\right)+l_f\left(\boldsymbol{\chi}_N\right) \tag{2.53a}$$

$$\text{s.t. } \boldsymbol{\chi}_0=\boldsymbol{x}_0 \tag{2.53b}$$

$$\boldsymbol{\chi}_{k+1}=\boldsymbol{x}_k\left(t_{k+1};\boldsymbol{\chi}_k,\boldsymbol{q}_k\right), \qquad k=0,\ldots,N-1 \tag{2.53c}$$

$$\boldsymbol{g}\left(\boldsymbol{\chi}_k,\boldsymbol{q}_k\right)\le 0, \qquad k=0,\ldots,N-1 \tag{2.53d}$$

$$\boldsymbol{g}_f\left(\boldsymbol{\chi}_N\right)\le 0. \tag{2.53e}$$

The objective function (2.53a) sums over the running costs per interval $l_k\left(\boldsymbol{\chi}_k,\boldsymbol{q}_k\right)$, thus

$$l_k\left(\boldsymbol{\chi}_k,\boldsymbol{q}_k\right)=\int_{t_k}^{t_{k+1}} l\left(\boldsymbol{x}_k\left(t_k;\boldsymbol{\chi}_k,\boldsymbol{q}_k\right),\boldsymbol{q}_k\right)\mathrm{d}t. \tag{2.54}$$

Further, (2.53b) constrains the initial value of the first artificial state vector $\boldsymbol{\chi}_0$, and (2.53c) ensures continuity between the artificial states $\boldsymbol{\chi}_{k+1}$ and the physical ones $\boldsymbol{x}\left(t_{k+1};\boldsymbol{\chi}_k,q_k\right)$. The discretized path and terminal constraints are denoted by (2.53d) and (2.53e), respectively.

By solving the ODE system f independently on every discretization interval, the numerical stability of the optimization process is significantly improved [161, p. 10], since the growth of an error stemming from a poor initialization of the optimization problem is prevented. Additionally, inherent instabilities of the ODE system are propagated in a minimal way [163]. Prior knowledge of the state trajectory can be used during initialization of the optimization algorithm when the multiple shooting technique is applied. However, the resulting optimization problems is larger compared to single shooting, since additional artificial states $\boldsymbol{\chi}_k$ need to be introduced.

Direct Collocation

Direct collocation discretizes the states and the controls simultaneously [161, p. 8 f.]. The system behavior for $t\in[0,T]$,

$$\begin{aligned} \dot{\boldsymbol{x}} \quad &- \quad \boldsymbol{f}\left(\boldsymbol{x}(t),\boldsymbol{u}(t)\right) \quad &&=0 \\ \frac{\boldsymbol{\chi}_{k+1}-\boldsymbol{\chi}_k}{t_{k+1}-t_k} \quad &- \quad \boldsymbol{f}\left(\frac{\boldsymbol{\chi}_k+\boldsymbol{\chi}_{k+1}}{2},\boldsymbol{q}_k\right) \quad &&=0 \quad =\boldsymbol{h}_k\left(\boldsymbol{q}_k,\boldsymbol{\chi}_k,\boldsymbol{\chi}_k',\boldsymbol{\chi}_{k+1}\right), \end{aligned} \tag{2.55}$$

is integrated into the optimization problem by finitely many equality constraints $\boldsymbol{h}_k$. The symbols $\boldsymbol{\chi}'_k$ denote intermediate states between the main discretization points k. We will neglect the dependency on $\boldsymbol{\chi}'_k$ in the remainder of this section.

The resulting NLP can be written as [161, p. 9 f.]

$$\min_{\boldsymbol{\chi} \in \mathbb{R}^{N \times n_\mathrm{x}},\, \boldsymbol{q} \in \mathbb{R}^{N-1 \times n_\mathrm{q}}} \sum_{k=0}^{N-1} l_k\left(\boldsymbol{q}_k, \boldsymbol{\chi}_k, \boldsymbol{\chi}_{k+1}\right) + l_\mathrm{f}\left(\boldsymbol{\chi}_N\right) \tag{2.56a}$$

$$\text{s.t. } \boldsymbol{\chi}_0 = \boldsymbol{x}_0 \tag{2.56b}$$

$$\boldsymbol{h}_k\left(\boldsymbol{q}_k, \boldsymbol{\chi}_k, \boldsymbol{\chi}_{k+1}\right) = 0, \qquad k = 0, ..., N-1 \tag{2.56c}$$

$$\boldsymbol{g}\left(\boldsymbol{\chi}_k, \boldsymbol{q}_k\right) \leq 0, \qquad k = 0, ..., N-1 \tag{2.56d}$$

$$\boldsymbol{h}_\mathrm{f}\left(\boldsymbol{\chi}_N\right) \leq 0, \tag{2.56e}$$

where the running costs can approximately be integrated by

$$l_k\left(\boldsymbol{\chi}_k, \boldsymbol{\chi}_{k+1}, \boldsymbol{q}_k\right) = \int_{t_k}^{t_{k+1}} l\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right) \mathrm{d}t \approx l\left(\frac{\boldsymbol{\chi}_k + \boldsymbol{\chi}_{k+1}}{2}, \boldsymbol{q}_k\right)\left(t_{k+1} - t_k\right). \tag{2.57}$$

The benefits of direct collocation include the sparsity of the resulting NLP. However, it comprises a higher dimensionality compared to multiple shooting. Direct collocation shows fast local convergence during the iterative solving procedure. Moreover, the state trajectory $\boldsymbol{x}(t)$ can be initialized prior to the optimization process, and inherently unstable systems can be handled by this discretization technique [161, p. 10].

2.3 Trajectory Planning

The task of trajectory planning with regards to the driving dynamics of a vehicle, is a widely studied field. There are real-time-capable algorithms, which are usually developed to be operated in an autonomous driving software stack and therefore tailored to a specific task. Offline methods are commonly used to perform, e.g., sensitivity analyses of vehicle parameters. A prominent application regarding passenger vehicles are the eco-driving algorithms, which are used offline for the analysis of a globally optimal route selection and driving behavior. In an online fashion, they can evaluate the driving pattern regarding, e.g., the driver's energy efficiency. In racing, the calculation of a globally time-optimal race trajectory is rather performed offline. In contrast, the planning and control of a fully autonomous race vehicle must be calculated by sufficiently fast algorithms online. In rare cases, pure planning algorithms consider the behavior of an electric powertrain, since electrified autonomous racing series are just evolving. Nevertheless, the SOC of the energy storage, and the temperatures of the powertrain components are crucial to operate at the global performance limits. In the remainder of this section we will therefore give a brief overview of the related literature (Table 2.3), focusing on planning algorithms suited for racing applications. Further analyses can be found in the main part of this thesis, which is framed by our prior publications that also incorporate detailed topic-specific literature reviews.

Table 2.3: Literature overview of trajectory planning algorithms, subdivided into groups according to their online capability and the incorporation of the electric powertrain behavior.

	No el. powertrain behavior	**Incorporated el. powertrain behavior**
Offline	[160, 164–170]	[127, 171–180]
Online	[24, 181–201]	[14, 202–210]

2.3.1 Offline Computation Without Powertrain Behavior

In the following we will describe offline trajectory optimization algorithms, which do not consider a detailed powertrain behavior. They are used for sensitivity analyses of the vehicle parameters and to find global time-optimal race trajectories.

The theoretically optimal maneuvering behavior of a Formula 1 car has been studied via formulation of NLPs. These have subsequently been solved using either an SQP method [164] that is implemented in the sparse nonlinear optimizer (SNOPT) solver package [211], or a nonlinear IP method as in IPOPT [212]. These minimum lap time problems (MLTPs) are extended by the three-dimensional geometry of race tracks in [169]. To solve them, the general purpose optimal control software (GPOPS-II) [213] is used, which transcribes the formulated OCP via a direct pseudo-spectral collocation method. Further extensions include the thermodynamic tire behavior [170], and the interaction of the suspension system and the aerodynamics [168]. By this means, the influence of tire wear and the aerodynamic setup on the lap time performance was analyzed. The work in [166] increases the vehicle dynamics model complexity to 14 degrees of freedom. A GP2 car is modeled as a multi-body system, which is capable of calculating wheel spin, suspension travel, and the full chassis motion. This detailed description allows to tune, e.g., the aerodynamic setup or gearing settings. The resulting OCP is solved by the indirect method PINS [214]. Tire-specific friction coefficients are considered in [165]. The formulated MLTP is transcribed via the direct orthogonal collocation discretization technique. In turn, the resulting NLP is fed into the nonlinear IPOPT solver [152].

A less computational-intense approach approximates the time-optimal global race path [167]. To do so, a minimum-curvature QP formulation, based on an occupancy grid map, is deduced.

In order to optimize the vehicle movement during a double-lane change manoeuvre [160] formulates an OCP, which comprises the vehicle dynamics of a single track model and includes integer variables to represent gear shifts. The applied solver package multiple shooting code for direct optimal control (MUSCOD-II) [215] implements a multiple shooting discretization technique, and allows to select various SQP-type solution strategies [216].

2.3.2 Offline Computation With Powertrain Behavior

When it comes to offline trajectory planning algorithms that also consider the electric powertrain behavior, the majority of the available literature has been published most recently.

For solar race cars, velocity profiles [174] and energy management strategies [171] have been optimized. The strategies in both aforementioned publications compute the time-optimal race trajectories whilst the latter one takes also the electric motor efficiency and battery constraints into account. As solvers for the resulting NLPs, stochastic algorithms and MATLAB's general purpose solver "fmincon" were leveraged.

An energy management strategy is also vital for hybrid electric vehicles. The presented algorithm in [179] minimizes the usage of fuel energy. The resulting optimization problem is relaxed to obtain a convex form and subsequently solved by the embedded conic solver (ECOS) [217]. Also, [176] deduces the optimal velocity profile and energy management strategies for a hybrid Formula 1 powertrain architecture by a convex formulation. The optimized control inputs are the engine input and the mechanical motor generator unit power, which lead to a minimum lap time. This formulation was also adapted to describe qualifying scenarios [180]. For the same race format, [175] optimizes the energy management and gearshift strategies, iteratively combining algorithms based on Pontryagin's minimum principle (PMP), DP, and convex optimization.

Finally, fully electric powertrain architectures have been considered in the frame of MLTPs. The convex [173] and the quasi-convex OCP formulations in [127, 172] derive minimum time velocity profiles for the Le Mans race circuit. They compare fixed gear and continuously variable transmission systems including different levels of transmission efficiency and available battery energy. Moreover, [127] incorporates the thermodynamic behavior of the electric motor such that the optimal velocity profiles do not violate the critical admissible machine temperatures. A nonlinear OCP to deduce an energy strategy for the Formula E racing format is presented in [178], where GPOPS-II was leveraged as the NLP solver. The presented results discuss the effects of different energetic and thermal powertrain constraints. For the same problem, also artificial neural networks have been used [177]. In this approach the battery temperature and its SOC span the decision variables. The presented method provides optimal control predictions for horizons of up to 32 race laps. Conversely, subsequent extensions of the neural network by, e.g., different physical effects, require a computationally intensive re-training phase.

2.3.3 Online Planning Without Powertrain Behavior

In online-capable trajectory planning approaches the computation time plays a crucial role. Specifically, its peak values can cause serious safety issues for an autonomous vehicle operating at the limits of handling. Therefore, different concepts have been formulated in the literature to efficiently compute local trajectories in an autonomous driving software stack.

Decoupled approaches are presented in [187, 193, 201]: First, a physically drivable path is selected and, second, a feasible velocity profile is added. Therein, the paths are generated by graph search and mesh methods. For the isolated task of velocity profile planning, tailored packages for ground vehicles like MTSOS [191] are available. Similar graph- and sampling-based approaches can be found in [185, 186, 188, 189, 192, 196, 199]. Specifically, [196] sets up a multi-layered graph-based path planner tackling non-convex scenarios, and generates feasible solutions during car following or overtaking maneuvers. The algorithm generates drivable splines offline and selects them online through solving a shortest path problem.

Since decoupling the path and velocity planning problem can lead to suboptimal solutions, research has also been done on combined approaches. Computational efficient algorithms use techniques of iteratively linearizing the underlying optimization model. Based on this method, an MPC was implemented [183], which also avoids obstacles, leveraging the least squares solver LSSOL [218]. Its real time capability was validated in a test vehicle. A similar approach is shown in [181] which also considers velocity-dependent friction maxima. The resulting problem was solved by the IBM "cplexqp" solver routine [219]. A stochastic MPC is proposed in [200], which is solved on a powerful GPU to operate a model car. To iteratively increase the controller performance [182] proposes a repetitive learning MPC, which solves the underlying optimization problem in real time using IPOPT.

Safety trajectories are computed in [197] by an OCP, where the working principle was validated in simulation. The experiments were conducted dealing with objects traveling at city driving speeds. The optimality of the numerical results calculated using the automatic control and dynamic optimization (ACADO) toolkit [220] was additionally validated against a DP algorithm.

Static obstacles were also dealt with in [24] by replanning given reference trajectories at spatially fixed points on the race track. The formulation is based on a point mass including longitudinal weight shifting, resulting in a quadratically constrained quadratic program (QCQP). Leveraging the nonlinear solver FORCES Pro [221], the computation times allow to evade static obstacles at experimentally validated velocities above $100\,\mathrm{km\,h^{-1}}$. The studies in [198] combine a trajectory pre-sampling method and a subsequent optimization, which is solved using Gurobi [222], to select the cost-minimal side to pass an obstacle. Thereby, a varying tire road friction potential is included. Instead of pre-sampling trajectories to cope with the logical constraints of where to overtake other cars, [190, 195] formulate mixed-integer quadratic programs (MIQPs). These formulations are particularly promising to find solutions with global optimality guarantees but they can demand large peak computation times.

Additional safety features through intermediate milestones were included in [184], which addresses the problem of finite computation times on limited hardware resources. Therein, a randomized incremental roadmap planning algorithm, which considers dynamic constraints on the vehicle's motion, is proposed. The approach was validated on an autonomous ground robot and a model helicopter. To operate in uncertain environments including dynamic obstacles, [194] sets up an OCP to navigate the vehicle through the remaining free space, which is limited by the obstacle movement predictions. The approach was validated experimentally on a model-sized KUKA robot. The resulting trajectory planning NLP is solved using CasADi and IPOPT.

2.3.4 Online Planning With Powertrain Behavior

Currently, there are few publications available which deal with the task of online trajectory planning while simultaneously incorporating the powertrain behavior in the modeling approach. Therefore, we will also present related eco-driving algorithms that are not necessarily well-suited for racing applications. However, some of their general ideas can be reused.

Analytical solutions to eco-driving problems for combustion, hybrid, and electric powertrain architectures are summarized in [209].The work in [202] proposes the formulation of an OCP, which objective function strives to minimize the energy drawn from the battery storage. The algorithm can be used online as a driver assistance system to inform about the optimal velocity profile. Therein, the powertrain losses are assumed to be quadratically dependent on the PMSM torque. Similarly, [210] implements an MPC to predictively adapt the vehicle velocity profile for energy saving purposes. The underlying system model is based on a point mass description including a loss map of the electric motor. The gradient-based MPC (GRAMPC) solver was chosen [223], which is specifically designed for OCPs. Most recent literature also formulates numerically efficient eco-driving algorithms leveraging IPOPT for different powertrain architectures [14], and to incorporate driver preferences [203].

When it comes to racing applications, [204] proposes a multi-stage optimization method for a solar-powered vehicle. Thereby, an ES comprising several race days is computed. In addition to planning this high-level strategy, the detailed, continuous vehicle control inputs are simultaneously optimized on the low level. A nonlinear loss model of the BLDC motor is included. With the help of a pseudo-spectral discretization technique and the general purpose optimal

control software (GPOPS), competitive calculation times were achieved. For Formula 1, a series of publications dealing with real-time-capable ESs are available: In [206] the optimal power distribution for hybrid high-performance powertrains is studied. The work derives an ES analytically leveraging PMP and nonsmooth analysis. In a subsequent work, a two-level MPC scheme is introduced to recalculate the formulated ES, thus reacting to unforeseen events in real time [207]. Therein, an upper level is implemented, which runs a convex MPC that is solved by ECOS [217] to frequently update the lap-time-optimal strategy. The additional low-level controller tracks the state trajectories computed by the high-level algorithm by iteratively solving an LP via the commercial package FORCES Pro [221]. Also, a feedback controller, inspired by the equivalent consumption minimization strategy (ECMS), was implemented to achieve a minimum lap time [208]. The presented approach is able to handle stochastic disturbances by tracking and adapting an offline-generated ES. Moreover, this algorithm drastically reduces the computational effort compared to the MPC approach. To jointly optimize the race trajectory and the hybrid powertrain's power flow, [205] proposes an indirect method operating in real time.

3 Proposed Energy Strategy Architecture

As stated in the previous Section 2.3, many algorithms, which deal with the task of local trajectory planning for vehicles, incorporating the driving dynamics, are available. However, these algorithms seldom consider the combination of thermal and energetic constraints, stemming from the electric powertrain operating at the performance maximum. By doing so, their computational complexity would increase substantially and they may therefore lose the sufficiency for an online operation. Some offline-suited global race trajectory algorithms partially include the electric powertrain behavior. Nevertheless, these are not designed for an online integration on an embedded vehicle electric control unit (ECU). With the evolution of autonomous electric vehicles, algorithms dealing with the efficient consumption of energy, while simultaneously adhering to the thermodynamic powertrain constraints, become vital. Moreover, these algorithms must seamlessly be integrable in an existing autonomous driving software stack, which operates the vehicle at the handling limits. Therefore, the following research questions arise and will be addressed throughout this thesis.

3.1 Research Questions

- Which software module architecture is appropriate to realize an ES, building on an existing software stack used for autonomous driving?
- How can an ES be structured to adhere to the technical limitations of the electric powertrain and the driving dynamics? How can the ES still use the available, limited amount of battery energy as lap-time-efficiently as possible to achieve a time-minimal result in a racing application?
- Which mathematical and numerical formulations of the ES algorithms are real-time-capable but will simultaneously predict small physical errors regarding the powertrain behavior?
- Which mathematical formulation of the ES is adequate for future extensibility by additional physical constraints whilst keeping its computational complexity small?

3.2 Energy Strategy for Battery Electric Race Vehicles

Following up on the first two research questions, we introduce the software stack used in the autonomous driving competition Roborace in Figure 3.1 [12, 224]. The structure of the

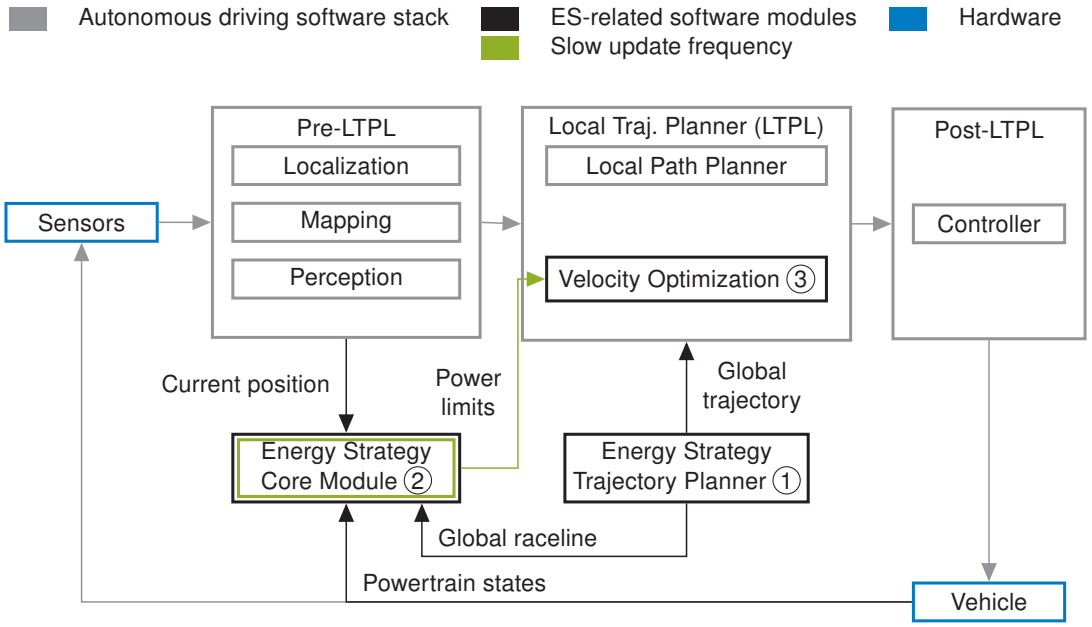


Figure 3.1: Software stack of an autonomous, electric car. The software modules which have been created or extended by ES-related content, are marked by black rectangles, the ES Core module is additionally highlighted in green to express its slower update frequency in comparison with the rest of the software stack [9, 12, 224].

software modules basically follows the scheme of navigation, guidance, and control as well as the high-level architecture introduced in Chapter 1.

The software modules that extend this software stack by energy strategy-related functionality are marked in black. To clarify the basic ideas of the proposed ES architecture, we will briefly introduce the local trajectory planner and its requirements in greater detail in the following paragraph, since energy-related considerations must be taken into account during the planning of the vehicle movement. Subsequently, we will explain the key ideas of the energy strategy-related software modules to justify their functionality and position in the proposed architecture. We introduce the final software architecture here, since it is important to understand the interconnection and the interplay of the single software modules before diving into their specific functionality. The software architecture in Figure 3.1, which is also part of the results of this thesis, will be presented in Chapter 5 in greater detail.

3.2.1 Local Trajectory Planner in an Autonomous Driving Software Stack

A local trajectory must be [225] [226]

- collision-free,
- physically drivable with respect to the vehicle parameters,
- within the identified and free driving space,
- continuous in its heading, and – depending on the type of vehicle motion controller – be steering-continuous [167, 196],
- recursively feasible regarding the planning horizon by restricting the final state of every local trajectory to a safe one.

Furthermore, and especially in autonomous racing, there are several performance requirements a local trajectory planner must meet. First, it must be able to solve combinatorial problems to identify the best suitable path avoiding or overtaking obstacles [227]. To deal with this task, an approach capable of discrete decision making is necessary. Second, tight real-time requirements must be met. Operating a vehicle at speeds of up to $220\,\mathrm{km\,h^{-1}}$ as in the Roborace competition [25] or even $270\,\mathrm{km\,h^{-1}}$ as in the Indy Autonomous Challenge [44] requires the trajectory planner to operate at a frequency of at least $10\,\mathrm{Hz}$ [196]. Smaller update rates would result in an uncontrollable vehicle motion, since the reaction time regarding static or dynamic obstacles would be too high. In the following section we will therefore propose an ES architecture, which tackles these challenges.

3.2.2 Energy Strategy Architecture

To meet the introduced requirements of the trajectory planning algorithm and those of its output in real time, we propose the ES architecture given in black rectangles in Figure 3.1. It considers the energetic and thermodynamic powertrain behavior by extending the autonomous driving software stack by three contributions. These are the “ES Trajectory Planner”, the “ES Core Module” that adapts the ES in real time, and a “Velocity Optimization” algorithm. Their functionality will be introduced in the paragraphs below.

① Energy Strategy Trajectory Planner (Global, Offline)

This module builds upon the offline race trajectory optimization algorithm of [165]. We extend the nonlinear double track model by the description of a fully electric powertrain to deduce the power losses occurring in each of its components. Therewith, we compute the powertrain components’ thermodynamic behavior and include constraints on the energetic and thermodynamic state variables, which must not be exceeded when following the global race trajectory. By this, the influence of the powertrain thermodynamics on the optimal raceline, i.e., the geometry of the trajectory, and the corresponding time-optimal velocity profile can be studied. Furthermore, the generated racelines serve as input into the ES Core Module.

Publications: [2, 3]; open source (OS) Code: [6].

② Energy Strategy Core Module (Global, Online)

The ES Core Module includes an online-capable algorithm, which iteratively reoptimizes the power usage of the vehicle for a horizon of the remaining race distance and thereby solves a minimum race time problem (MRTP). To do so, it adapts the ES to unforeseen events and disturbances to permanently adhere to the constraints arising from the driving dynamics, energy, and thermodynamics. Finally, the calculated power limitations are forwarded to the LTPL. The power loss descriptions of the powertrain components, that translate into temperature contributions, are based on physically detailed models. The component losses and their temperature profiles have been validated with the help of measurement data of a full-scale electric race vehicle. The corresponding publication additionally presents a justification of the selected numerical solver to iteratively compute the ES, and validates the module’s results for a single race lap by a comparison to a state-of-the-art MLTP algorithm.

The ES Core Module’s computation time lies in the range of approx. $0.1\,\mathrm{Hz}$. Since an autonomous vehicle’s planning frequency must be significantly faster, the module cannot be integrated in

series with the other software modules of the autonomous driving software stack. Therefore we propose the dual architecture given in Figure 3.1. Here, the ES Core Module is running in parallel to the Pre-, Post-, and LTPL modules. The outer software loop provides local trajectories with a sufficiently high update rate of about 10 Hz to 20 Hz. The inner software loop (connections to and from the ES Core Module, which is marked in green) is permitted to replan the ES with a frequency of about 0.1 Hz, since the proposed architecture decouples the task of fast trajectory planning and vehicle control from the ES recalculations. Once a valid ES for the remaining race distance is available, it will be updated with a frequency which is in accordance with the slower change rate of the powertrain thermodynamics. Thus the power limitations can be independently retrieved by the velocity optimization module.

Publication: [9]; OS Code: [10]; Manual [11].

③ Velocity Optimization (Local, Online)

As part of the trajectory planning module, the velocity optimization algorithm acts as the interface between both, the inner and the outer software loop in the architecture. In real time, it computes speed profiles, leading to a time-minimal racing behavior on the locally selected paths [196]. To achieve this goal, its computation time is kept low by setting up the underlying optimization problem as numerically efficient as possible. However, the multi-parametric formulation enables to incorporate external information stemming from, e.g., the ES Core Module, which provides spatially discretized maximum power values. The real-time capability of the formulated algorithm was proven on an embedded vehicle ECU in a high-fidelity nonlinear hardware-in-the-loop (HIL) simulator in the corresponding publication. Additionally, this module's output speed profiles have been validated by a comparison to different mathematical problem formulations and optimization solvers.

Publication: [4]; OS Code: [7, 8]; Manual: [5].

4 Energy Strategy Trajectory Planner (Global, Offline)

The offline calculation of the ES is vital to study the impact of energetic and thermodynamic constraints of the powertrain components on the global race trajectory. The following sections will quantify their influence, and derive conceptual ideas for the implementation of an online-capable algorithm for the dynamic replanning of the ES.

4.1 Concept and the Influence of Energetic Constraints – ITSC 2019

Summary

The corresponding publication presents the concept for the static calculation of an ES for autonomous electric race cars. The proposed OCP identifies the time-minimal global trajectories, consisting of path and velocity, along the race track. The algorithm builds upon the MLTP depicted in [165], which is extended by constraints imposed by energetic limitations and recuperation through braking. Moreover, we deduce and interpret their influence on the geometry of the optimal global race path and the related vehicle velocities.

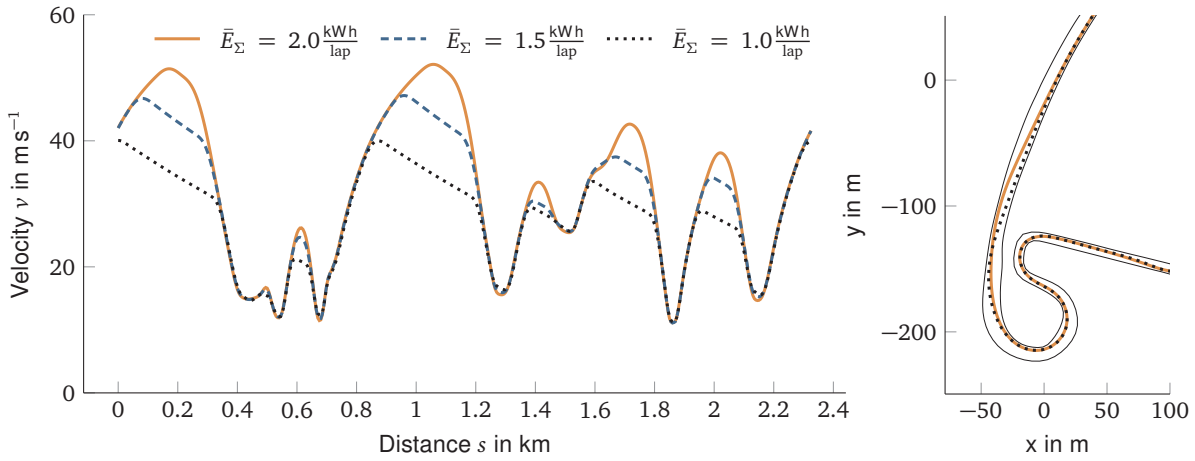
The literature review testifies to a broad variety of publications dealing with the optimal control of road vehicles. They differ in the applied mathematical and numerical approaches, ranging from analytical solutions, mixed-integer linear programs (MILPs), MIQPs, graph search methods, second order conic programs (SOCPs), towards NLPs [160, 164, 169, 176, 179, 184, 188, 195, 202, 208, 209, 228]. We choose to solve the energy-constrained MLTP using an NLP, since this formulation allows for a subsequent extension by the powertrain component behavior, delivers results which have proven to cohere with real-world applications, and promises competitive computation times.

The presented OCP models the vehicle dynamics as a nonlinear double track model, including the tire behavior using the Pacejka formulation [229]. The space-dependent state variables comprise the vehicle dynamics with the reference line forming the basis of the trajectory optimization. The control inputs are the acceleration and brake forces as well as the steering angle.

After the transcription via direct collocation to an NLP within the CasADi framework [149], the optimization problem is solved by IPOPT [152] (Section 2.2). IPOPT is well-known for its accurate, high-quality results [151]. The results present the significant influence of energetic constraints on the time-optimal, global race trajectories, and are summarized in Figure 4.1. We vary the allowed amount of energy per lap consumption $\bar{E}_{\Sigma}$ manually: By decreasing this design parameter the vehicle reduces the velocity maxima through shorter acceleration phases combined with coasting. Thus the race car rolls off into the braking zones, leveraging the aerodynamic drag to slow down.

Moreover, this behavior influences the optimal race path. Trajectories constrained by tight energy limitations are shorter in distance to compensate for a time-minimal operation. The vehicle is still able to follow the suggested path geometries, since the combination of sharper radii in front of the curve entries in combination with reduced velocities still adhere to the tire constraints.

The correlation between the energy demand per lap and the achievable lap time is displayed as a Pareto curve. Its hyperbolic shape suggests that a significant reduction of the energy consumption relates to a small increase in lap time in a certain area of the plot. Powertrain recuperation improves the Pareto front and thereby allows for a more beneficial compromise between both opposing objectives.



(a) Velocities. (b) Racelines.

Figure 4.1: Energy-constrained time-optimal race trajectories. The race path for $\bar{E}_\Sigma = 1.5\frac{\text{kWh}}{\text{lap}}$ is not displayed, © 2019 IEEE.

Contributions

T. H. initiated the idea of the paper and contributed significantly to the concept and the presented ES. F. C. drew up the formulation of the minimum lap-time problem as an OCP including the numerical solver in his Master's thesis. J. B. contributed to the whole concept of the paper. M. L. provided a significant contribution to the concept of the research project. He revised the paper critically for important intellectual content. M. L. gave final approval for the publication of this version and is in agreement with all aspects of the work. As a guarantor, he accepts responsibility for the overall integrity of this paper.

Copyright notice

Energy Management Strategy for an Autonomous Electric Racecar using Optimal Control

Thomas Herrmann[1], Fabian Christ[2], Johannes Betz[1] and Markus Lienkamp[1]

***Abstract*— The automation of passenger vehicles is becoming more and more widespread, leading to full autonomy of cars within the next years. Furthermore, sustainable electric mobility is gaining in importance. As racecars have been a development platform for technology that has later also been transferred to passenger vehicles, a race format for autonomous electric racecars called *Roborace* has been created.**

As electric racecars only store a limited amount of energy, an Energy Management Strategy (EMS) is needed to work out the time as well as the minimum energy trajectories for the track. At the same time, the technical limitations and component behavior in the electric powertrain must be taken into account when calculating the race trajectories. In this paper, we present a concept for a special type of EMS. This is based on the Optimal Control Problem (OCP) of generating a time-minimal global trajectory which is solved by the transcription via direct orthogonal collocation to a Nonlinear Programming Problem (NLPP). We extend this minimum lap time problem by adding our ideas for a holistic EMS. This approach proves the fundamental feasibility of the stated ideas, e.g. varying race-paths and velocities due to energy limitations, covered by the EMS. Also, the presented concept forms the basis for future work on meta-models of the powertrain's components that can be fed into the OCP to increase the validity of the control output of the EMS.

I. INTRODUCTION

In 2018, the Technical University of Munich (TUM) participated in the first Roborace event [1]. Roborace stages the first race series for autonomous vehicles (*Robocars*) and is a support series for Formula E. The software stack developed by the team at TUM used to operate the Robocar [2] is already partially publicly available [3]. This paper presents a concept and the main ideas for an EMS that extends the software module calculating the global trajectories leading to the minimum lap time [4]. The EMS is crucial as it considers component behavior and the inherent limitations of the all-electric powertrain. This is necessary because of the significant influence of components on the minimum achievable race time in total over all laps.

According to [5], the results from research in the field of autonomous motorsport provide information on future autonomous road vehicles for the following three reasons:

- The algorithms developed for and tested in an autonomous racecar must be capable of being calculated using limited computational resources in real-time and must be exceptionally robust.
- As the Robocars have electric powertrains, the development of an EMS that enables energy-saving and energy recovery ensures progress beyond established technologies, including series production vehicles.
- Since it is also included in the presented EMS, the choice of trajectory is of major interest for autonomous passenger vehicles as well as racecars for reasons of safety, range or a minimal lap time.

For these reasons, the EMS is being developed in the context of autonomous motorsport for testing under extremely tough conditions in an enclosed environment, where technical effects and correlations of the powertrain components can be clearly seen.

The structure of this paper is as follows: In Section II, the state of the art is being summarized. Section III contains the concept of the presented EMS. Section IV describes the methods used together with formulation of the OCP, including its states and control inputs as well as the powertrain architecture of an electric rear wheel drive vehicle. The results of the presented OCP are described in Section V, Section VI summarizes the results obtained and also states the direction of future work and how the presented OCP will be extended.

II. STATE OF THE ART

The optimal control of vehicles is a complex field that has been dealt with in prior publications. In the following, several different mathematical approaches are summarized in order to plan velocities, paths or entire trajectories with optimal control based approaches. Their specific advantages and drawbacks are described.

Within this section we do not distinguish between the specific objectives that are being optimized in these approaches. They are classified according to the mathematical optimization methods, to solve the OCP.

1) Explicit solution: The authors of [6] and [7] calculate one-dimensional velocity profiles for predefined paths to minimize the energy demand of the traveling vehicle. The vehicle's dynamics are expressed using a simplified point mass model [8]. The driving resistance is described applying Newton's second law resulting in

$$m\ddot{x} = \frac{1}{r} M_\mathrm{e} i_\mathrm{g} - \frac{1}{2}\rho_\mathrm{a} A c_\mathrm{w} \dot{x}^2 - mgc_\mathrm{r} - mg\sin\left(\alpha(x)\right) \quad (1)$$

where x is the vehicle's position, m the vehicle mass, M_e the output torque of the electric machine, i_g the gear

[1]Thomas Herrmann, Johannes Betz and Markus Lienkamp are with the Chair of Automotive Technology, Faculty of Mechanical Engineering, Technical University of Munich, 85748 Garching b. Muenchen, Germany thomas.herrmann@tum.de

[2]Fabian Christ is with the Chair of Automatic Control, Department of Mechanical Engineering, Technical University of Munich, 85748 Garching b. Muenchen, Germany

transmission, r the wheel radius, ρ_a the air density, A the vehicle's front surface, c_w the aerodynamic drag coefficient, g the gravitational acceleration, c_r the rolling resistance coefficient and $\alpha(x)$ the road slope.
To bring the energy consumption E_Σ into play, [6] introduces a second order polynomial of the form

$$E_\Sigma = \int b_1 \tau + b_2 \dot{x}(t) \tau^2 \mathrm{d}t \tag{2}$$

with b_1, b_2 being constant fitting coefficients and τ the required torque. Neglecting aerodynamic drag, [7] approximates the energy demand by

$$E_\Sigma = \int F_\mathrm{w} \dot{x}(t) \mathrm{d}t \tag{3}$$

where F_w depicts the traction force deduced from (1). The Hamiltonian is devised to deduce an explicit solution to the stated OCP. As one can distinguish, only simple model equations can be formulated to be able to determine a solution using an explicit approach.

2) Mixed Integer Linear/Quadratic Programming: Mixed Integer Linear Programming (MILP) or Mixed Integer Quadratic Programming (MIQP) is used widely in order to generate optimal trajectories for specific driving maneuvers. In [9], an MIQP formulation is used to solve problems like vehicle overtaking, obstacle avoidance or lane changes. To plan global trajectories, [10] uses MILP to find the time-minimal path while simultaneously avoiding static obstacles. Furthermore, [11] uses MILP to calculate the optimal energy distribution within an electric powertrain during driving to control fail-operational power nets. However, the driving kinematics modeled within the enumerated publications are based on simplified point mass models.

3) Graph search: To account for more complex models of vehicle dynamics, nonlinearities in objective functions or in the boundary conditions of the optimization problem, graph search approaches are used widely. The architecture used in [12] addresses the dynamic constraints on the vehicle's motion in real-time. An RRT*-algorithm is implemented to solve the minimum lap-time problem using a half-car dynamic model to find its local steering input in [13].

4) Convex optimization: There are several approaches to relax an OCP of electric vehicles to a convex optimization problem [14], [15]. The main advantage of these reformulations is the almost negligible calculation time needed to solve the defined problem. On the other hand, these problem formulations suffer from subsequent extension, since any additional equality or inequality constraint must also be formulated to fit into the existing framework. Furthermore, [15] and [16] assume a fixed driving path as well as a point mass to reduce the complexity of the optimization problem. This in turn enables calculation of a time-optimal velocity profile exploiting the convexity of the problem formulation.

5) Nonlinear Programming: [17] uses Nonlinear Programming (NLP) to solve a minimal lap time problem for a Formula 1 racecar modeled as nonlinear double track model including a detailed tire model. The problem formulation is extended in [18] taking gear shifts into account. Track-specific parameter optimization is done in [19]. [20] also considers three-dimensional track courses when solving the minimal lap time problem. As one can recognize, complex dynamic scenarios are modeled using an NLP approach. Unfortunately, the computing times for solving these are relatively long in comparison with other mathematical problem formulations.

III. Concept

This paper covers the concept and the main ideas for an EMS for autonomous electric cars. This EMS aims to find the minimum race time by optimizing global lap trajectories (path & velocity) while taking the technical constraints of the components in the all-electric powertrain into account. The results in this paper demonstrate the overall feasibility of the stated OCP for one lap trajectory. The formulation of this optimal control based approach enables us to extend the problem formulation by component behaviors in the future. Furthermore, we will be able to consider multiple consecutive race laps.
On the racetrack, highly dynamic driving scenarios, including maximum velocities and peak positive as well as negative accelerations, occur. These put enormous stress on electric powertrain components. Additionally, environmental conditions vary, depending on race locations. Therefore, extreme heat or cold or humidity and aridness can occur. Due to these facts, component behavior must be considered when solving the minimal race time OCP.
Our approach is based on our previous work, as presented in [4]. Therein, an OCP for planning time-optimal trajectories was formulated that allows for easy subsequent extension via the components' behavior within the electric powertrain. In this way, we can consider technical constraints and include energetic considerations in the OCP. This extension enables planning of the global race trajectories for all race laps that need to be completed.
Effects within the powertrain leading to unexpected component behavior or limited available power can be due to:

- Reaching the maximum permitted battery temperature.
- The maximum permitted temperature of the electric machines are reached, especially during qualifying, as higher peak power is allowed there compared to the race itself.
- A decreased level of efficiency of the motor inverters due to the battery's voltage drop to equal the input voltage of the motor inverters and due to increased inverter temperature.
- Quadratically higher thermal power loss ($P_{\mathrm{l,T}} \propto I^2$) due to higher current I within the powertrain owing to the continuous voltage decrease in accordance with the State of Charge (SOC) of the main battery.

The most important components of the electric powertrain with rear wheel drive are depicted in Fig. 1 with

- Energy storage represented as the battery (B).

- Power electronics converting the battery's Direct Current (DC) into Alternating Current (AC) at rear left (I_l) and rear right (I_r).
- Synchronous permanent electric machines at the rear left (M_l) and rear right (M_r).
- Gears attached to the electric machines ($G_{\mathrm{l/r}}$) to transform the motor torque into drive torque.
- Sensors for autonomous driving (A_x) that need to be powered by the battery and must not be neglected, since they consume a major amount of the vehicle's total energy demand.

The wheels are W_rl and W_rr; displayed is only the rear part of the whole powertrain.

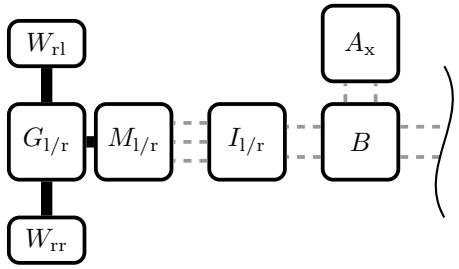


Fig. 1. Electric powertrain architecture of a rear wheel drive vehicle

IV. METHODOLOGY

Section III presented a concept and the main ideas for the EMS for an autonomous electric racecar. This strategy will lead to the minimum race time T_Σ totaled over all the race lap times T_i,

$$T_\Sigma = \sum_{i=1} T_i. \tag{4}$$

The vehicle's dynamics are described using a nonlinear double track model that includes longitudinal, lateral and yaw freedoms. This means that a quasi-steady state wheel load transfer is permitted whenever the car is accelerating or cornering. For a detailed description of the required first-order ordinary differential equations describing the model dynamics, we refer to [4].

An OCP including equality and inequality constraints to be solved by the EMS is defined by [21, p. 478], [22, p. 127], [23, p. 215]:

$$\min \; l(\boldsymbol{x}) \tag{5}$$

$$s.t. \; \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}s} = f(\boldsymbol{x}(s), \boldsymbol{u}(s)) \tag{6}$$

$$h_\mathrm{i} = 0 \tag{7}$$

$$g_\mathrm{j} \leq 0 \tag{8}$$

with $i = 1, ..., m$ and $j = 1, ..., r$, where s as the independent variable within the OCP denotes the distance along the reference line of the racetrack (Fig. 2).
The following summarizes the formulation of the time-minimal OCP defined in our previous work [4] and is extended by energy-related considerations.
The state vector $\boldsymbol{x}(s)$ within the OCP is defined as

$$\boldsymbol{x}(s) = \begin{pmatrix} v & \beta & \dot{\psi} & n & \xi \end{pmatrix}^T \tag{9}$$

with the vehicle's states v, β as the side slip angle and ψ the yaw angle of the vehicle as well as the path model's states n denoting the lateral distance to the reference line and ξ being the relative angle of the vehicle's longitudinal axis to the tangent tan on the reference line. Fig. 2 visualizes the used states of the OCP. θ denotes the angle between tangent and local x-axis. The gray rectangle with rounded corners represents the vehicle heading north-east.

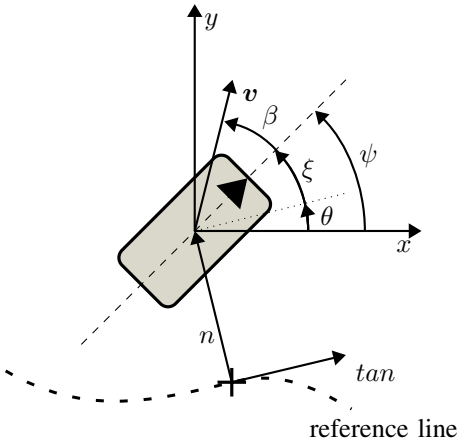


Fig. 2. Vehicle and path model used in the OCP [4]

The control input vector $\boldsymbol{u}(s)$ is defined by

$$\boldsymbol{u}(s) = \begin{pmatrix} F_\mathrm{d} & F_\mathrm{b} & \delta & \gamma \end{pmatrix}^T \tag{10}$$

with F_d being the driving force, F_b the braking force, δ the steering input and γ the wheel load transfer according to the nonlinear double track model [4].
The independent variable s changes according to

$$\dot{s} = \frac{v \cos(\xi + \beta)}{1 - n\kappa}. \tag{11}$$

The transitions of the states in $\boldsymbol{x}(s)$ can be summarized as follows:

$$\dot{v} = \pi_1(v, \beta, \delta, F_\mathrm{tire}) \tag{12}$$

$$\dot{\beta} = \pi_2(v, \beta, \delta, \dot{\psi}, F_\mathrm{tire}) \tag{13}$$

$$\ddot{\psi} = \pi_3(\delta, F_\mathrm{tire}) \tag{14}$$

$$\dot{n} = v \sin(\xi + \beta) \tag{15}$$

$$\dot{\xi} = \dot{\psi} - \kappa \frac{v \cos(\xi + \beta)}{1 - n\kappa} \tag{16}$$

where $\kappa = \frac{1}{R}$ describes the geometrical curvature by the inverse radius R of a local curve in the reference line. The $\pi_\mathrm{i}(\cdot)$ denote a mathematical function. For a detailed description of these equations, we refer to our previous work [4].

Here, $f(\boldsymbol{x}(s), \boldsymbol{u}(s))$ describes the system dynamics and contains (12) - (16), as well as a functional correlation of the

wheel load transfers according to the used nonlinear double track model [4].
The objective function $l(\boldsymbol{x})$ that minimizes the race time can be written as

$$l(\boldsymbol{x}) = \int_0^{S_\Sigma} L(\boldsymbol{x}(s), \boldsymbol{u}(s))\mathrm{d}s \tag{17}$$

with $L(\cdot)$ being the Langrangian cost function defined in [24]

$$L = \frac{\mathrm{d}t}{\mathrm{d}s} = \frac{1 - n\kappa}{v\cos(\xi + \beta)} \tag{18}$$

and S_Σ the entire race distance cumulated over all race laps. The Lagrangian is also needed to transform the time-dependent Ordinary Differential Equations (ODEs) into space-dependent ones.
The equality constraints are

$$h_1 = \gamma - \Pi \tag{19}$$
$$h_2 = F_\mathrm{d} \cdot F_\mathrm{b} \tag{20}$$

with Π denoting the lateral wheel load transfer. The inequality constraints are

$$g_1 = \sqrt{F_\mathrm{x}^2 + F_\mathrm{y}^2} - 1 \tag{21}$$
$$g_2 = F_\mathrm{d} v - P_\mathrm{max} \tag{22}$$
$$g_3 = F_\mathrm{d} - F_\mathrm{d,max} \tag{23}$$
$$g_4 = -F_\mathrm{d} \tag{24}$$
$$g_5 = F_\mathrm{b} \tag{25}$$
$$g_6 = \delta - \delta_\mathrm{max} \tag{26}$$
$$g_7 = \delta_\mathrm{min} - \delta \tag{27}$$
$$g_8 = \frac{\Delta F_\mathrm{d}}{L\Delta s} - \frac{F_\mathrm{d,max}}{T_\mathrm{d}} \tag{28}$$
$$g_9 = \frac{F_\mathrm{b,max}}{T_\mathrm{b}} - \frac{\Delta F_\mathrm{b}}{L\Delta s} \tag{29}$$
$$g_{10} = \frac{\Delta\delta}{L\Delta s} - \frac{\delta_\mathrm{max}}{T_\delta} \tag{30}$$
$$g_{11} = \frac{\delta_\mathrm{min}}{T_\delta} - \frac{\Delta\delta}{L\Delta s}, \tag{31}$$

where F_x and F_y are the longitudinal and the lateral forces set down by the tire. P_max is the maximum traction power of the vehicle, $F_\mathrm{d,max}$ and $F_\mathrm{d,min}$ describe the maximum drive and break force, respectively, δ_max and δ_min denote the maximum positive and the minimal negative steering angle input. With T_j, appropriate time constants are defined to restrict the actuator dynamics [24].
The energy consumption during driving E_Σ is calculated using

$$E_\Sigma = \int_0^{T_\Sigma} P_\mathrm{d} + \bar{f}_\mathrm{r} P_\mathrm{b}\mathrm{d}t \tag{32}$$
$$= \int_0^{T_\Sigma} (F_\mathrm{d} + \bar{f}_\mathrm{r} F_\mathrm{b}) v \mathrm{d}t \tag{33}$$

with P_d being the driving power, P_b the braking power and $\bar{f}_\mathrm{r}$ denoting a mean recuperation factor between $0\,\%$ and $100\,\%$. The braking energy can then partially be re-stored in the battery (Fig. 1).
The inequality constraint

$$g_{12} = E_\Sigma - \bar{E}_\Sigma \tag{34}$$

results, limiting the maximum available amount of energy ($\bar{E}_\Sigma$) for the entire race.

V. Results

This section presents the results obtained with the primal-dual interior-point method IPOPT called by CasADi [25]. The execution time of the solver for the NLP for one parameter set was always less than a minute on a PC with an i7-7820HQ CPU and 16GB of memory. The discretization step size is $\Delta s = 5.0\,\mathrm{m}$.
In all of the following plots, the states $\boldsymbol{x}(s)$ as well as the control inputs $\boldsymbol{u}(s)$ are optimized to obtain the minimum race time. To show the feasibility of the presented concept, the race consists of a single lap on the Formula E track in Berlin, Germany.

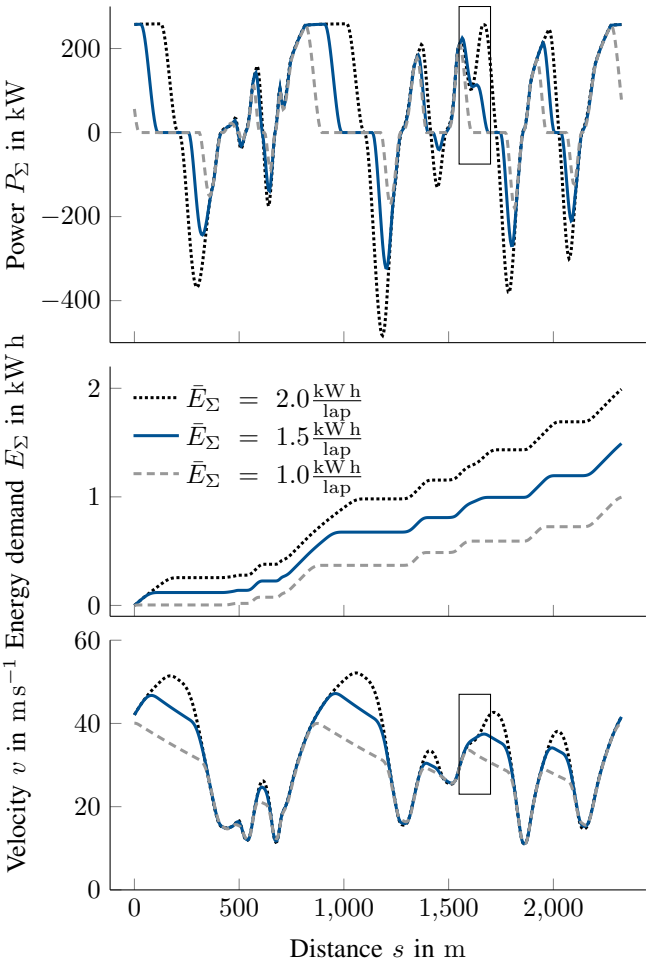


Fig. 3. Power P_Σ, energy demand E_Σ and velocity $v(s)$ over the raceline distance s on the Berlin Formula E racetrack

In Fig. 3, the effects of different limitations of the maximum allowed amount of energy per lap $\bar{E}_\Sigma$ are shown. The influence of $\bar{E}_\Sigma$ can clearly be seen in a constantly

repeating pattern: Peak velocity values are avoided if $\bar{E}_\Sigma$ gets decreased due to the driving resistance growing quadratically with the velocity ($E_\Sigma \propto v^2$). Peak power demands $P_\Sigma(s)$ are requested on shorter distances and coasting phases ($P_\Sigma(s) = 0\,\mathrm{kW}$) increase with a smaller E_Σ. The behavior at $s \approx 1600\,\mathrm{m}$ differs a little from the described pattern but reinforces this explanation (marked with rectangles in the plots): At this position on the racetrack, a curve to the right follows a short straight part. With the limited available amount of energy of $\bar{E}_\Sigma = 1.0\frac{\mathrm{kW\,h}}{\mathrm{lap}}$ no positive acceleration takes place before the curve. In contrast, with twice the amount of energy allowed for the lap $\bar{E}_\Sigma = 2.0\frac{\mathrm{kW\,h}}{\mathrm{lap}}$, a positive acceleration occurs resulting in a higher velocity on the straight part. These described patterns are similar to the technique of "lift and coast" in Formula E: Shortly before a curve, the optimized controlling policy suggests reduction of the peak power request and keeping the vehicle rolling for a short distance Δs. This behavior helps to lose kinetic energy without braking while simultaneously also reducing total energy demand.

In the middle of Fig. 3, the cumulated energy demand $\bar{E}_\Sigma$ is depicted. As the recuperation factor f_r was set to $0\,\%$ for these experiments, these graphs either ascend or remain constant. As already stated in the concept in Section III, the time minimal path of the raceline varies, depending on the limited amount of every $\bar{E}_\Sigma$ available per lap. In Fig. 4, both displayed paths vary, especially from $(25\,\mathrm{m}, 0\,\mathrm{m})$ to $(-40\,\mathrm{m}, -200\,\mathrm{m})$ corresponding to the interval of $s = [0\,\mathrm{m}, 500\,\mathrm{m}]$ of the path distance. In this part of the racetrack the time-minimal path ($\bar{E}_\Sigma = 2.0\frac{\mathrm{kW\,h}}{\mathrm{lap}}$) is a little longer than the one for $\bar{E}_\Sigma = 1.0\frac{\mathrm{kW\,h}}{\mathrm{lap}}$. This is because of the lower curvature κ following $s_{\bar{E}_\Sigma = 2.0\frac{\mathrm{kW\,h}}{\mathrm{lap}}}$, which simultaneously allows a higher peak velocity and therefore a smaller lap time.

A summary of the results presented is given in Fig. 5. Two Pareto fronts containing the correlation between the minimal reachable lap time T_Σ with a limited amount of available energy $\bar{E}_\Sigma$ including an average recuperation factor $\bar{f}_\mathrm{r}$ are shown. The hyperbolic form of the two fitted curves expresses the following: reducing the lap time leads to a squared increase in the energy demand E_Σ for the same lap. When introducing an average recuperation factor, converting negative braking force F_b to charge the battery (Fig. 1), the Pareto front bends into the direction of the origin of the plot. With the help of these Pareto fronts, the decision on how to set up the race strategy can be made: the effects of a fast lap on energy consumption can be distinguished and the consequences for subsequent laps can be deduced. The future integration of powertrain components' behavior into the OCP, as well as extension of the optimization problem to multiple race laps, help to determine a realistic race strategy for the entire race.

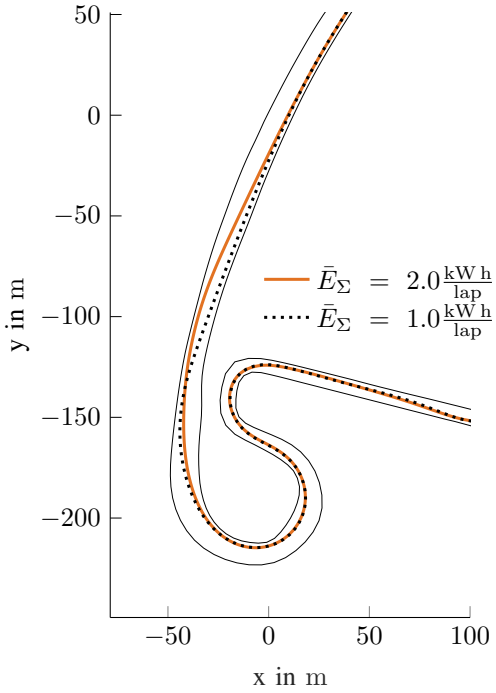


Fig. 4. Time optimal raceline paths for different energy limitations on the Formula E racetrack in Berlin, Germany

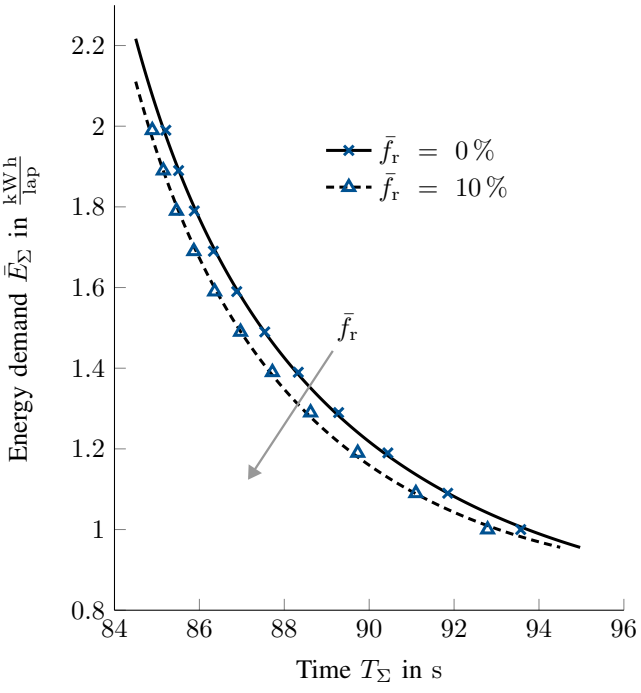


Fig. 5. Pareto fronts containing the minimum reachable lap time T_Σ versus a limited energy demand $\bar{E}_\Sigma$ including an average recuperation factor $\bar{f}_\mathrm{r}$

VI. CONCLUSION & OUTLOOK

In this paper, we presented an optimal control based approach for determining time-optimal lap trajectories for a rear wheel drive all-electric autonomous racecar, taking energy limitations into account. The solver IPOPT was able to find a feasible solution to the formulated OCP in less than one minute on a standard user PC. The results show that

the EMS can determine the relationships between requested power, energy demand and path velocity when energy restrictions need to be met. Furthermore, the assumption that the optimal driving path differs from the time-minimal path when energetic limitations are considered, is confirmed.
This paper defines the basis for future work: The OCP presented will be extended to multiple race laps instead of one. Furthermore, the behavior of the powertrain's components will be described using meta-models. These can then be included in the OCP to achieve results taking the powertrain's states, e.g. temperatures of the energy storage or the electric machines, into account. With the help of these meta-models, a complete race strategy for a given powertrain configuration and an entire race can be determined.

CONTRIBUTIONS

Thomas Herrmann initiated the idea of the paper and contributed significantly to the concept and the presented EMS. Fabian Christ drew up the formulation of the minimum lap-time problem as an OCP including the numerical solver in his Master's thesis. Johannes Betz contributed to the whole concept of the paper. Markus Lienkamp provided a significant contribution to the concept of the research project. He revised the paper critically for important intellectual content. Markus Lienkamp gave final approval for the publication of this version and is in agreement with all aspects of the work. As a guarantor, he accepts responsibility for the overall integrity of this paper.

ACKNOWLEDGMENT

We would like to thank the Roborace team for giving us the opportunity to work with them and for the use of their vehicles for our research project. We would also like to thank the *Bavarian Research Foundation (Bayerische Forschungsstifung)* for funding us in connection with the "rAIcing" research project.
We would also like to thank Alexander Wischnewski and Leonhard Hermansdorfer for valuable discussions about the basic ideas behind this paper.

REFERENCES

[1] Roborace, "Roborace Ltd." 2018. [Online]. Available: https://roborace.com/

[2] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, T. Stahl, L. Hermansdorfer, and M. Lienkamp, "A Software Architecture for an Autonomous Racecar," in *89th Vehicular Technology Conference*, IEEE Intelligent Transportation Systems Society, Ed., In Press, 2019.

[3] Chair of Automotive Technology, "Autonomous Driving Control Software of TUM Roborace," 2019. [Online]. Available: https://github.com/TUMFTM/veh_passenger

[4] F. Christ, A. Wischnewski, A. Heilmeier, and B. Lohmann, "Time-Optimal Trajectory Planning for an Autonomous Race Car with variable friction coefficients," *Vehicle System Dynamics*, submitted, 2019.

[5] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, T. Stahl, L. Hermansdorfer, B. Lohmann, and M. Lienkamp, "What can we learn from autonomous level-5 motorsport?" in *9th International Munich Chassis Symposium 2018*, ser. Proceedings, P. Pfeffer, Ed. Wiesbaden: Springer Fachmedien Wiesbaden, 2019, pp. 123–146.

[6] W. Dib, A. Chasse, P. Moulin, A. Sciarretta, and G. Corde, "Optimal energy management for an electric vehicle in eco-driving applications," *Control Engineering Practice*, vol. 29, pp. 299–307, 2014.

[7] A. Sciarretta, G. de Nunzio, and L. L. Ojeda, "Optimal Ecodriving Control: Energy-Efficient Driving of Road Vehicles as an Optimal Control Problem," *IEEE Control Systems*, vol. 35, no. 5, pp. 71–90, 2015.

[8] N. Petit and A. Sciarretta, "Optimal drive of electric vehicles using an inversion-based trajectory generation approach," in *Proceedings of the 18th IFAC World Congress, 2011*, vol. 44. s.l.: IFAC-PapersOnLine, 2011, pp. 14 519–14 526.

[9] X. Qian, F. Altche, P. Bender, C. Stiller, and A. de La Fortelle, "Optimal Trajectory Planning for Autonomous Driving Integrating Logical Constraints: An MIQP Perspective," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. Piscataway, NJ: IEEE, 2016.

[10] T. A. Ademoye and A. Davari, "Trajectory Planning of Multiple Autonomous Systems Using Mixed-Integer Linear Programming," in *Proceedings of the 38th Southeastern Symposium on System Theory*. Piscataway, NJ: IEEE Operations Center, 2006, pp. 260–264.

[11] K. Gorelik, A. Kilic, R. Obermaisser, and N. Müller, "Modellprädiktives Energiemanagement mit Steuerung der Fahrzeugführung für automatisiertes Fahren," *at - Automatisierungstechnik*, vol. 66, no. 9, pp. 735–744, 2018.

[12] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-Time Motion Planning for Agile Autonomous Vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 116–129, 2002.

[13] J. h. Jeon, R. V. Cowlagi, S. C. Peters, S. Karaman, E. Frazzoli, P. Tsiotras, and K. Iagnemma, "Optimal Motion Planning with the Half-Car Dynamical Model for Autonomous High-Speed Driving," in *American Control Conference (ACC)*. Piscataway, NJ: IEEE, 2013, pp. 188–193.

[14] N. Murgovski, L. Johannesson, X. Hu, B. Egardt, and Sjoeberg J., "Convex relaxations in the optimal control of electrified vehicles," in *American Control Conference (ACC), 2015*. Piscataway, NJ: IEEE, 2015, pp. 2292–2298.

[15] S. Ebbesen, M. Salazar, P. Elbert, C. Bussi, and C. H. Onder, "Time-optimal Control Strategies for a Hybrid Electric Race Car," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 233–247, 2018.

[16] M. Salazar, C. Balerna, E. Chisari, C. Bussi, and C. H. Onder, "Equivalent Lap Time Minimization Strategies for a Hybrid Electric Race Car," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 6125–6131.

[17] D. Casanova, "On Minimum Time Vehicle Manoeveuring: The Theoretical Optimal Lap," PhD Thesis, Cranfield University, 2000.

[18] C. Kirches, S. Sager, H. G. Bock, and J. P. Schlöder, "Time-optimal control of automobile test drives with gear shifts," *Optimal Control Applications and Methods*, vol. 31, no. 2, pp. 137–153, 2010.

[19] M. Imani Masouleh and D. J. N. Limebeer, "Optimizing the Aero-Suspension Interactions in a Formula One Car," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 3, pp. 912–927, 2016.

[20] D. J. N. Limebeer and G. Perantoni, "Optimal Control of a Formula One Car on a Three-Dimensional Track—Part 2: Optimal Control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 5, p. 051019, 2015.

[21] D. P. Bertsekas, *Dynamic programming and optimal control*, fourth ed. ed., ser. Athena scientific optimization and computation series. Belmont, Mass.: Athena Scientific, 2016, vol. 3.

[22] S. P. Boyd and L. Vandenberghe, *Convex optimization*, ser. Safari Tech Books Online. Cambridge, UK and New York: Cambridge University Press, 2004. [Online]. Available: http://proquest.safaribooksonline.com/9781107385924

[23] R. Sioshansi and A. J. Conejo, *Optimization in engineering: Models and algorithms*, ser. Springer optimization and its applications. Cham: Springer, 2017, vol. volume 120.

[24] I. Gundlach, U. Konigorski, and J. Hoedt, "Zeitoptimale Trajektorienplanung fuer automatisiertes Fahren im fahrdynamischen Grenzbereich," *VDI Reports*, no. 2292, pp. 223–234, 2017.

[25] J. A. E. Andersson, J. Gillis, and G. Horn, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, In Press, 2018.

4.2 Power Losses and the Influence of Thermodynamic Constraints – ITSC 2020

Summary

In contrast to ICE-powered cars, electric powertrains face tighter temperature limitations. Especially in electric racing applications these thermodynamic limits must often be exploited to achieve a minimum race time totaled over the subsequently driven laps. The DevBot 2.0, which is used as the unitary race car in the Roborace competition among the teams [25], is equipped with an all-electric powertrain. Its energy storage, consisting of lithium-ion batteries, the VSIs, and the temperature hotspots in the PMSMs, which are the windings, must not exceed $55\,°$C, $100\,°$C and $180\,°$C, respectively. In this publication we therefore show the significant influence of the powertrain thermodynamics on the race time performance. When the vehicle follows the deduced global static ES trajectory, the powertrain components operate at their technical limits without hazarding safety shutdowns while simultaneously achieving a minimum race time. Note that no energetic constraints are active in this problem formulation. By this, we isolate the resulting effects on the static ES due to the thermodynamic constraints, and are able to discuss them independently.
Apart from the algorithmic implementation of the OCP, this publication defines the three levels that form the proposed ES of this thesis, which was introduced in greater detail in Section 3. The global static level comprises the offline computation of the global race trajectories as explained in Sections 4.1 and 4.2. These plannings need to be dynamically adapted during a race on the global level to react to disturbances or unforeseen events, which are studied in Chapter 5. Finally, a local path and velocity planner is necessary to realize the globally (re-)planned strategies, which are the content of Chapter 6.

A literature study reveals several publications dealing with MLTPs [160, 169, 176, 183, 188, 206, 210]. However, these do not consider the behavior of electric powertrains. Therefore we extend the OCP presented in Section 4.1 [2], comprising the race car's driving dynamics as a nonlinear double-track model, by the powertrain component thermodynamics. The powertrain architecture and its cooling circuits are depicted in Figure 2.1. We approximate the power losses of the PMSMs and the VSIs as second order polynomials. These are deduced using measurement data from our nonlinear HIL-simulator [230]. The description of the battery pack follows an open circuit model, consisting of a constant OCV and a constant internal resistance [50, p. 51].

The calculated power losses are directly connected to a temperature contribution in the electric components and their cooling fluids. We deduce the governing ODEs of first order, which express the temperature dynamics, with the help of a lumped thermal network of the entire powertrain. Herein, the thermal resistance of the PMSMs (Figure 4.2) is modeled in detail to accurately describe the heat flow $P_{\mathrm{col,M}}$ from their temperature hotspots, which are the stator end windings (temperature T_{W}) [124]. The thermodynamic power $P_{\mathrm{col,M}}$ is concurrently conducted to the rotor and the exterior of the electric motor (temperatures $\bar{T}_{\mathrm{F1,M}}$).

After the transcription of the OCP to an NLP via direct collocation using the CasADi modeling language [149], the discretized optimization problem was solved by the primal-dual IP method IPOPT [152].

The powertrain's influence on the time-optimal race strategy is shown by comparing two scenarios, spanning two race laps each, where the initial values of the thermodynamic ODEs have been varied (Figure 4.3): The baseline scenario called "cold" (index "-") is necessary to calculate the reference race strategy without active thermodynamic constraints. The scenario "hot" (index

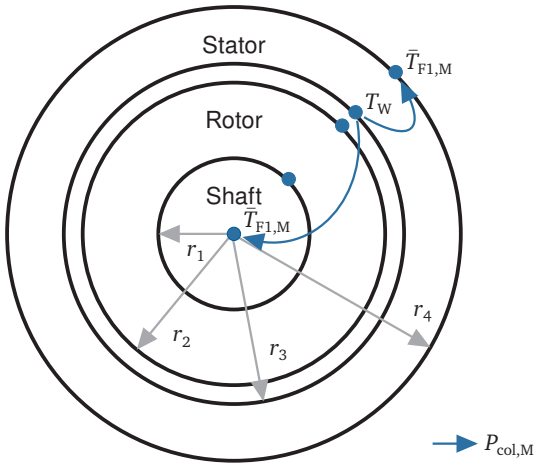


Figure 4.2: Thermal resistance model of the PMSM, © 2020 IEEE.

"+") reveals valuable insights on how to achieve a minimum race time, while simultaneously adhering to the technical powertrain constraints. Here, we set the initial battery temperature $T_{\mathrm{B}}(s)$ equal to almost its allowed maximum. By following the corresponding optimal race strategy "hot", the critical battery temperature $T_{\mathrm{B,max}}$ is reached directly when crossing the finish line. This result can be achieved by a strategy that is different to the lift-and-coast strategy, which was identified when energetic constraints were active in Section 4.1: In this experiment, the vehicle linearly decreases the requested driving power on the straights. Through this, the absolute temperature of the battery can nearly be kept constant on the straights, resulting in an optimal cooling through the fluid temperature $T_{\mathrm{F2}}(s)$ which, in turn, allows for a high driving velocity $v(s)$. The temperatures of the electric motors, inverters, and the motor-inverter cooling liquid can be neglected in this experiment, since they stay far below their limitations.

The influence of the thermodynamic constraints on the geometry of the race path, which is a free optimization variable in the presented OCP, turned out to be minor.

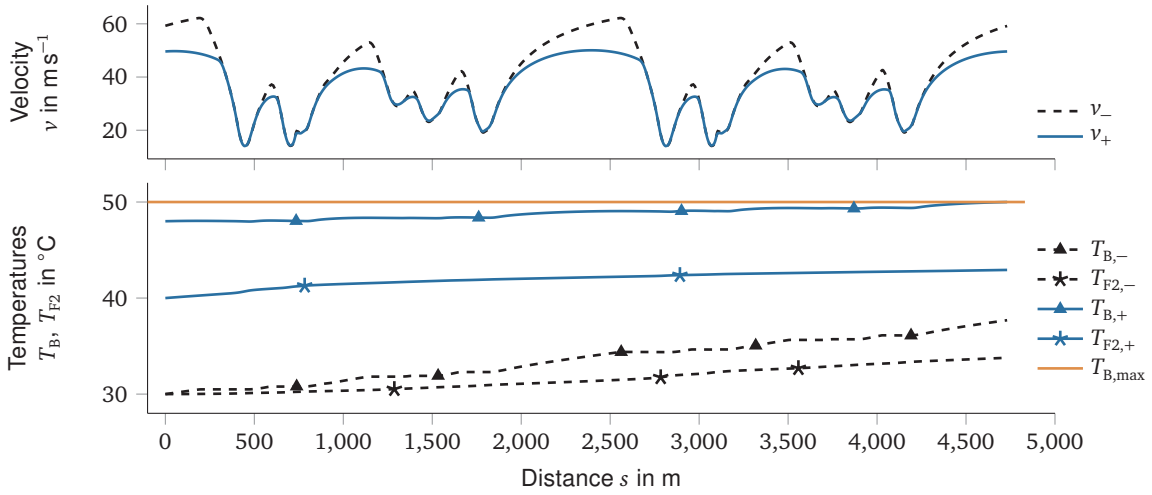


Figure 4.3: Thermodynamically constrained time-optimal race profiles over two consecutive laps, © 2020 IEEE.

Contributions

T. H. initiated the idea of the paper and contributed significantly to the concept, modeling, and results. F. P. modeled the powertrain thermodynamics in his Master's thesis. J. B. contributed to the whole concept of the paper. M. L. provided a significant contribution to the concept of the research project. He revised the paper critically for important intellectual content. M. L. gave final approval for the publication of this version and is in agreement with all aspects of the work. As a guarantor, he accepts responsibility for the overall integrity of this paper.

Copyright notice

Minimum Race-Time Planning-Strategy for an Autonomous Electric Racecar

Thomas Herrmann[1], Francesco Passigato[1], Johannes Betz[1] and Markus Lienkamp[1]

***Abstract*— Increasing attention to autonomous passenger vehicles has also attracted interest in an autonomous racing series. Because of this, platforms such as Roborace and the Indy Autonomous Challenge are currently evolving.**
Electric racecars face the challenge of a limited amount of stored energy within their batteries. Furthermore, the thermodynamical influence of an all-electric powertrain on the race performance is crucial. Severe damage can occur to the powertrain components when thermally overstressed. In this work we present a race-time minimal control strategy deduced from an Optimal Control Problem (OCP) that is transcribed into a Nonlinear Problem (NLP). Its optimization variables stem from the driving dynamics as well as from a thermodynamical description of the electric powertrain. We deduce the necessary first-order Ordinary Differential Equations (ODE)s and form simplified loss models for the implementation within the numerical optimization. The significant influence of the powertrain behavior on the race strategy is shown.

I. INTRODUCTION

The first autonomous race series for all-electric racecars is called *Roborace*. Its main goal is to be a platform for the development of software powering self-driving cars. Therefore, Roborace is a race format, bringing cars to their limits of handling [1]. The special requirements for the algorithms regarding computational resources, real-time capability, and robustness are thus outstanding [2], [3]. The race format of autonomous motor-sports delivers perfect conditions for testing under tough conditions in an enclosed environment. We, a team from the Technical University of Munich (TUM), are participating in this race series. Most parts of our software stack are available online [4]. This paper describes the extension of our software by a control strategy calculating the minimum race-time, taking into account energetic and thermal constraints arising from the powertrain architecture. The minimum time control strategy is one of three parts of our race strategy (Fig. 1). As discussed in the results section of this paper, the powertrain thermodynamics have a major impact on the entire race strategy.
This paper is based on the ideas for an energy management strategy for autonomous electric cars as stated in our previous work [5]. We extend the state of the art by taking into account multiple race laps as well as the thermodynamics of the all-electric powertrain in the OCP that needs to be solved.

[1]Thomas Herrmann (corresponding author), Francesco Passigato, Johannes Betz and Markus Lienkamp are with the Chair of Automotive Technology, Faculty of Mechanical Engineering, Technical University of Munich, 85748 Garching (Munich), Germany thomas.herrmann@tum.de

A. State of the Art

OCPs dealing with trajectory optimization, which is equivalent to solving a Minimum Lap Time Problem (MLTP) in motor-sports, are well known in the literature. Different mathematical approaches are used to solve an MLTP. The variety ranges from graph search methods [6] over Sequential Quadratic (SQP) [7] to Second Order Conic Problems (SOCP) [8]. By the transformation of the OCP into an NLP, the MLTP can be solved with detailed and complex double-track vehicle and tire models for the purpose of, e.g., vehicle parameter optimization for Internal Combustion Engine (ICE) powered cars [9]–[11]. Latest publications in the field of trajectory optimization also consider optimal power distributions within hybrid powertrains [12] or use Model Predictive Control (MPC)-approaches for the planning of energy-saving trajectories [13].
However, none of these sources consider the thermodynamics of the powertrain during their optimizations. Unless the temperature limits of a conventional ICE, the electric machines of the Roborace cars must not reach temperatures beyond $180\,^{\circ}\mathrm{C}$, the inverter's limit is $100\,^{\circ}\mathrm{C}$ [14]. Additionally, the efficiency level of an electric machine decreases as it heats up, leading to further reduction in efficiency [15], [16]. Furthermore, the energy storage, a lithium-ion battery in our case, must reduce its output power from $50\,^{\circ}\mathrm{C}$ to $0\,\%$ output power when reaching $55\,^{\circ}\mathrm{C}$ for safety reasons [14]. In order to therefore prevent the unwanted power loss, the thermal behavior of the powertrain components must be considered for consecutive race laps when dealing with all-electric racecars.
This paper is organized as follows: In Subsection I-B, an overview of the structure of our race strategy is given. Section II describes the powertrain architecture including power loss descriptions, thermodynamical models as well as the formulation of the optimization problem. Section III explains the results in detail. A summary of the presented work is given in Section IV.

B. Structure of the race strategy

The race strategy is split into three levels. All of these levels have a different optimization horizon as well as a different problem size stemming from the combination of their optimization horizon as well as their model complexities (Fig. 1).
Before the race starts, the global time-minimal trajectories per lap for the entire race are calculated offline. Here, we can use a non-linear double track model describing the driving dynamics as well as a detailed thermal powertrain

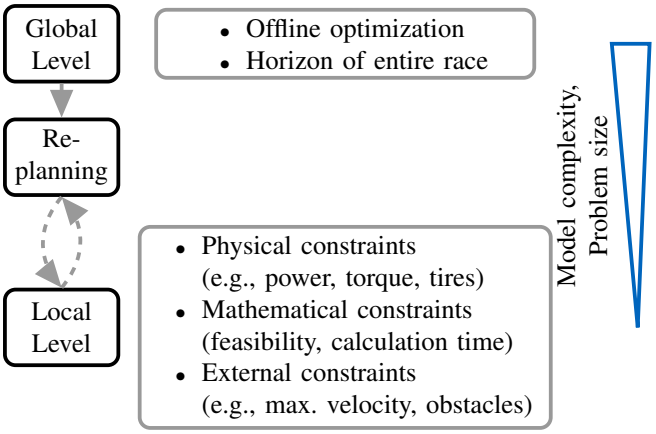


Fig. 1: Three levels of the proposed race strategy.

model to consider all necessary physical effects in detail. The pre-computed global trajectories are recalculated during the race, reacting to external disturbances, e.g., overtakes, using a significantly reduced optimization problem formulation. For example, the path can often be completely removed during the replanning phase as it is almost equal for all the race laps. The global trajectories are then fed into the local trajectory planner that transforms all given physical constraints from the global trajectory (e.g., max. power, max. torque) as well as mathematical requirements (e.g., guaranteed feasibility, calculation time) into locally optimal paths and velocities [17]. Furthermore, the local planner considers external influences, e.g., overtakes opponent cars or reacts to speed limits. In this paper we focus on the global level of the proposed race strategy and describe the offline optimization.

II. Powertrain Architecture & Modeling

The all-eletric powertrain of the racecar (Fig. 2) consists of

- A battery (B).
- Two power electronics/inverters at rear left and right ($\mathrm{I_{l/r}}$).
- Two synchronous permanent electric machines ($\mathrm{M_{l/r}}$).

Two separate cooling circuits are necessary in order to control the component temperatures T_c. Circuit 1 is responsible for machines $\mathrm{M_{l/r}}$ and inverters $\mathrm{I_{l/r}}$ leveraging radiator $\mathrm{R_{MI}}$. The same is true for circuit 2, battery B and radiator $\mathrm{R_B}$. For the sake of completeness, the gears ($\mathrm{G_{l/r}}$), sensors required for autonomous driving ($\mathrm{A_x}$), as well as the wheels $\mathrm{W_{rr/rl}}$, are also displayed within the rear part of the whole powertrain.

In this work, the index c indicates the powertrain component, i.e., $c \in \{\mathrm{M, I, B, R_{MI}, R_B}\}$. The second index d of the temperature symbols of both cooling liquids $T_{\mathrm{F1/2},d}$ enumerates the components the fluids are entering.

A. Power loss models

Meta-models of the powertrain are used to describe the internal losses of its components within the OCP. Mathematically, the meta-models for the electric machines as well as the inverters can be formulated as second order polynomials with the output power P_{out} as free variable (1):

$$P_{\mathrm{in,fit}}(P_{\mathrm{out}}) = a_{\mathrm{fit}} P_{\mathrm{out}}^2 + b_{\mathrm{fit}} P_{\mathrm{out}} + c_{\mathrm{fit}}. \tag{1}$$

The Mean Square Error (MSE) $e_{\mathrm{MS,fit}}$,

$$e_{\mathrm{MS,fit}} = \frac{1}{N} \sum_{i}^{N} \left(P_{\mathrm{in,fit},i} - P_{\mathrm{in,mes},i}\right)^2 \tag{2}$$

is minimized by fitting the constant parameters a_{fit}, b_{fit}, and c_{fit}. The input power P_{in} into the single components is a function depending on the requested output power P_{out} [8]. $P_{\mathrm{in,mes}}$ stems from measurement data from our Hardware-in-the-Loop (HiL)-Simulator [18] where detailed non-linear powertrain models, that are based on real-world measurement data from the Roborace cars, are implemented. The index i denotes a counter variable in the range $[1\ ..\ N]$.

Fig. 3 displays a polynomial fit to simulated data of an electric machine. The probability distributions of data on both axis indicate that mainly max- or minimal power is requested by the racecar. Therefore, the parabolic fit, showing high accuracy at these points, results in low MSEs. These are $e_{\mathrm{MS,fit,M}} = 3.19\,\%$ for the electric machine and $e_{\mathrm{MS,fit,I}} = 4.16\,\%$ for the inverter. The diagonal indicates $100\,\%$ efficiency.

We use an open circuit model to describe the internal battery power [19, p. 51] $P_{\mathrm{in,B}}$,

$$P_{\mathrm{in,B}}(P_{\mathrm{out,B}}) = \frac{U_{\mathrm{OCV}}^2}{2R_{\mathrm{i}}} - U_{\mathrm{OCV}} \frac{\sqrt{U_{\mathrm{OCV}}^2 - 4P_{\mathrm{out,B}}R_{\mathrm{i}}}}{2R_{\mathrm{i}}}. \tag{3}$$

The open circuit voltage is U_{OCV} and R_{i} is the internal battery resistance.

The component power loss $P_{\mathrm{los},c}$ can now be described using (1) and (3) that can easily be implemented within the numerical optimization,

$$P_{\mathrm{los},c} = P_{\mathrm{in},c} - P_{\mathrm{out},c}. \tag{4}$$

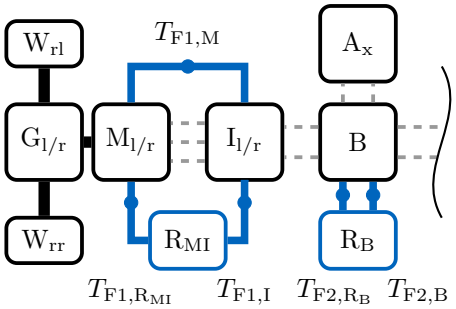


Fig. 2: Electric powertrain architecture of a rear-wheel drive vehicle including two separate cooling circuits.

B. Thermal models

As introduced in Section II, the thermal model of the powertrain is split into two circuits, one cooling the electric machines $\mathrm{M_{l/r}}$ and both inverters $\mathrm{I_{l/r}}$ in series using the radiator $\mathrm{R_{MI}}$, the other one being responsible for the battery

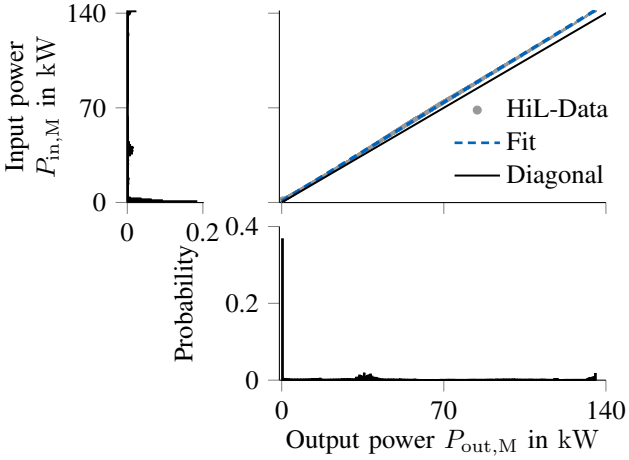


Fig. 3: Parabola fit to measurement data to deduce a polynomial expression of the electric machine's efficiency.

temperature T_{B} leveraging radiator $\mathrm{R_B}$ (Fig. 2). Power losses (4) from the electric components c are directly fed into the thermal circuits.

In the following, the ODEs describing the powertrain thermodynamics, i.e., the heat transfer from the powertrain components to the cooling liquids, are deduced. In general, the product of the thermal heat capacities C and the temperature gradients $\frac{\mathrm{d}T}{\mathrm{d}t}$ equal loss P_{los} and cooling power P_{col} for electric machines, inverters and battery:

$$C_{\mathrm{M}}\frac{\mathrm{d}T_{\mathrm{M}}}{\mathrm{d}t} = P_{\mathrm{los,M}} - \frac{T_{\mathrm{M}} - T_{\mathrm{M},\infty}}{R_{\mathrm{M}}} = \\ = P_{\mathrm{los,M}} - \underbrace{\frac{2T_{\mathrm{M}} - (T_{\mathrm{F1,M}} + T_{\mathrm{F1,R_{MI}}})}{2R_{\mathrm{M}}}}_{P_{\mathrm{col,M}}} \tag{5}$$

$$C_{\mathrm{I}}\frac{\mathrm{d}T_{\mathrm{I}}}{\mathrm{d}t} = P_{\mathrm{los,I}} - \underbrace{\frac{2T_{\mathrm{I}} - (T_{\mathrm{F1}} + T_{\mathrm{F1,M}})}{2R_{\mathrm{I}}}}_{P_{\mathrm{col,I}}} \tag{6}$$

$$C_{\mathrm{B}}\frac{\mathrm{d}T_{\mathrm{B}}}{\mathrm{d}t} = P_{\mathrm{los,B}} - \underbrace{\frac{T_{\mathrm{B}} - T_{\mathrm{F2}}}{R_{\mathrm{B}}}}_{P_{\mathrm{col,B}}}. \tag{7}$$

Here and in the following, $T_{\mathrm{F1}} = T_{\mathrm{F1,I}}$. The symbol $T_{\mathrm{M},\infty}$ denotes the temperature of the surroundings of the specific component M. This temperature can be assumed to be equal to the mean value of the inflowing and effluent cooling liquid temperature [20].

To describe the absorbed energy by the coolant fluid from both inverters the following formulation is used:

$$2\frac{T_{\mathrm{I}} - T_{\mathrm{I},\infty}}{R_{\mathrm{I}}} = \dot{m}_{\mathrm{F1}} c_{\mathrm{F}} \left(T_{\mathrm{F1,M}} - T_{\mathrm{F1}}\right), \tag{8}$$

where $\dot{m}_{\mathrm{F1}}$ describes the coolant mass flow through both inverters and c_{F} the specific heat capacity of the coolant fluid. Using $T_{\mathrm{I},\infty} = \frac{1}{2}\left(T_{\mathrm{F1}} + T_{\mathrm{F1,M}}\right)$, the following equations can be deduced to explicitly describe the temperatures of the cooling liquid entering electric machines as well as the radiator $\mathrm{R_{MI}}$:

$$T_{\mathrm{F1,M}} = \frac{T_{\mathrm{F1}}\left(\dot{m}_{\mathrm{F1}} c_{\mathrm{F}} R_{\mathrm{I}} - 1\right) + 2T_{\mathrm{I}}}{1 + \dot{m}_{\mathrm{F1}} c_{\mathrm{F}} R_{\mathrm{I}}} \tag{9}$$

$$T_{\mathrm{F1,R_{MI}}} = \frac{T_{\mathrm{F1}}\left(2\dot{m}_{\mathrm{F1}} c_{\mathrm{F}} R_{\mathrm{R_{MI}}} + 1\right) - 2T_{\mathrm{env}}}{2\dot{m}_{\mathrm{F1}} c_{\mathrm{F}} R_{\mathrm{R_{MI}}} - 1}. \tag{10}$$

We can formulate the gradients of T_{F1} and T_{F2} using the cooling power of the powertrain components P_{col} as well as the temperature differences to the environment T_{env},

$$C_{\mathrm{F1}}\frac{\mathrm{d}T_{\mathrm{F1}}}{\mathrm{d}t} = 2P_{\mathrm{col,M}} + \\ + 2P_{\mathrm{col,I}} - \\ - \frac{1}{R_{\mathrm{R_{MI}}}}\left(\frac{T_{\mathrm{F1,R_{MI}}} + T_{\mathrm{F1}}}{2} - T_{\mathrm{env}}\right) \tag{11}$$

$$C_{\mathrm{F2}}\frac{\mathrm{d}T_{\mathrm{F2}}}{\mathrm{d}t} = P_{\mathrm{col,B}} - \frac{T_{\mathrm{F2}} - T_{\mathrm{env}}}{R_{\mathrm{R_B}}}. \tag{12}$$

The thermal resistance of the motor model R_{M} can be written as a combination of two thermal resistances $R_{1/2}$ in parallel as the heat transfer acts from the air gap to both directions, the environment as well as its shaft (Fig. 4). For this model we make use of [21] that describes the stator winding temperature T_{W} as highest and most critical. $\bar{T}_{\mathrm{F1,M}} = \frac{1}{2}\left(T_{\mathrm{F1,M}} + T_{\mathrm{F1,R_{MI}}}\right)$ denotes the mean temperature of the cooling liquid through the electric machine. Therefore,

$$R_{\mathrm{M}} = \frac{R_1 R_2}{R_1 + R_2}, \tag{13}$$

with

$$R_1 = \frac{\ln\frac{r_4}{r_3}}{2\pi L k_{\mathrm{iro}}} + \frac{1}{2\pi r_4 L h_{\mathrm{f}}} \tag{14}$$

$$R_2 = \frac{\ln\frac{r_2}{r_1}}{2\pi L k_{\mathrm{iro}}} + \frac{1}{4\pi L k_{\mathrm{iro}}} + \frac{1}{2\pi r_3 L h_{\mathrm{g}}}. \tag{15}$$

Fig. 4 indicates the geometry of the electric machine. The first term in resistance R_1 takes into account conduction of the stator where L denotes its length and k_{iro} the thermal conductivity of iron. The second term describes the convective heat flux between the stator and the cooling liquid with h_{f} being the liquid's convective heat flux coefficient that can be assumed constant [22, p. 11]. Resistance R_2 consists of the thermal conductivity of the rotor and shaft [16] as well as the convection into the air gap with the respective heat flux coefficient h_{g}.

The thermal resistance of the inverter is assumed to be

$$R_{\mathrm{I}} = \frac{1}{A_{\mathrm{I}} h_{\mathrm{I}}}, \tag{16}$$

as well as for the battery [20]

$$R_{\mathrm{B}} = \frac{1}{A_{\mathrm{B}} h_{\mathrm{B}}}, \tag{17}$$

and the radiators

$$R_{\mathrm{R_{MI/B}}} = \frac{1}{A_{\mathrm{R_{MI/B}}} h_{\mathrm{R_{MI/B}}}}. \tag{18}$$

Here, A denotes the surface used for the heat exchange, h again represents heat flux coefficients.

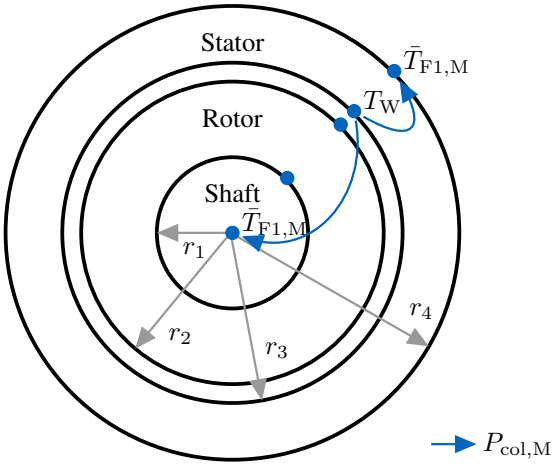


Fig. 4: Thermal resistance model of the electric machine.

C. Optimization problem

The OCP transformed into an NLP with equality and inequality constraints h_i and g_j has the form [23, p. 478], [24, p. 127]

$$\min\ l(\boldsymbol{x}) = \int_0^{S_\Sigma} \frac{\mathrm{d}t}{\mathrm{d}s}\mathrm{d}s = \int_0^{S_\Sigma} \frac{1-n\kappa}{v\cos(\xi+\beta)}\mathrm{d}s \tag{19}$$

$$s.t.\ \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}s} = f(\boldsymbol{x}(s), \boldsymbol{u}(s)) \tag{20}$$

$$h_i = 0 \tag{21}$$

$$g_j \leq 0 \tag{22}$$

with $i = 1, ..., m$ and $j = 1, ..., r$. The independent variable is s, the distance along the reference line, κ denotes the curvature profile of the race track. The objective is $l(\boldsymbol{x})$, defined as the integral over the lethargy $\frac{\mathrm{d}t}{\mathrm{d}s}$ [8] being minimized over the entire racing distance S_Σ. The lethargy can be interpreted as the time taken to travel a distance of $1\,\mathrm{m}$.
The state vector $\boldsymbol{x}(s)$ within the OCP is defined as

$$\boldsymbol{x}(s) = \Big(\underbrace{v\ \beta\ \dot{\psi}\ n\ \xi}_{\substack{\text{Driving}\\ \text{Dynamics}}}\ \underbrace{T_{\mathrm{M}}\ T_{\mathrm{I}}\ T_{\mathrm{B}}\ T_{\mathrm{F1}}\ T_{\mathrm{F2}}}_{\substack{\text{Thermo-}\\ \text{dynamics}}} \Big)^T, \tag{23}$$

consisting of the optimization variables needed to express the thermodynamics as introduced in Section II-B as well as the variables defining the driving dynamics. These variables are the velocity v on the raceline, the side slip angle β, the yaw angle ψ, the lateral distance to the reference line n, and ξ as the relative angle of the vehicle's longitudinal axis to the tangent on the reference line. For a detailed description of the driving dynamics as well as their first-order ODEs and constraints stemming from a nonlinear double track model, we refer to our previous works [5], [25], as we focus on the thermodynamical side within this paper.
The box constraints that are translated into inequality constraints for the thermodynamical state variables are

$$T_{c,\min} \leq T_c \leq T_{c,\max}, \tag{24}$$

with every component's allowed operating temperature range defining the lower $T_{c,\min}$ and upper boundaries $T_{c,\max}$.
The input vector has the form

$$\boldsymbol{u}(s) = \left(F_{\mathrm{d}}\ F_{\mathrm{b}}\ \delta\ \gamma\right)^T, \tag{25}$$

containing driving and braking force $F_{\mathrm{d}}/F_{\mathrm{b}}$, the steering angle δ, and the wheel load transfer γ as artificial control variable.

III. Results

The results were calculated using an i7-7820HQ CPU and $16\,\mathrm{GB}$ of memory. The NLP was solved with the primal-dual interior-point method IPOPT interfaced by CasADi [26] using a direct collocation transcription. The execution time of the solver for the NLP for two race laps was approximately $2.5\,\mathrm{min}$. The discretization step size varied along the race-track. In curves, a finer mesh was implemented to allow for a better description of the rapidly changing variables and their gradients leading to a step size of $\Delta s = 3\,\mathrm{m}$. On the straight parts, a coarser mesh of $\Delta s = 9\,\mathrm{m}$ was sufficient to reach high numerical precision in combination with small computation times. In total, approx. $44{\cdot}10^3$ decision variables and $50 \cdot 10^3$ constraints were present.
Two minimum race-time control strategies for a race, consisting of two laps, can be seen in Fig. 6. Two different cases were considered: In case "cold" $(-)$, the initial temperatures of all the powertrain components $T_{c,0,-}$ equaled the environment temperature T_{env}. In case "hot" $(+)$, the initial component temperatures $T_{c,0,+}$ were set to the values given in Table I.

TABLE I: Initial temperature values $T_{c,0}$ of powertrain components.

	$T_{\mathrm{M},0}$	$T_{\mathrm{I},0}$	$T_{\mathrm{B},0}$	$T_{\mathrm{F1},0}$	$T_{\mathrm{F2},0}$
			in °C		
"cold" $(-)$	30	30	30	30	30
"hot" $(+)$	100	70	48	55	40

In case "cold", none of the components c reached their maximum allowed temperature. Therefore, the maximum vehicle power of $P_\Sigma = Fv$ of $270\,\mathrm{kW}$ could be requested at all times when allowed by the driving dynamics as displayed in the last plot. The maximum physically achievable velocity of approx. $220\,\mathrm{km\,h^{-1}}$ for this vehicle on the Monteblanco race-track resulted on the straights. The battery temperature T_{B} remained far below the limit of $T_{\mathrm{B,max}} = 50\,^\circ\mathrm{C}$.
Case "hot" shows the necessity and performance of the developed control-strategy. Here, the initial component temperatures $T_{c,0,+}$ were set to a valid combination of reasonable values (Table I) that can occur during a race. The optimization's initial battery temperature $T_{\mathrm{B},0,+}$ almost equaled $T_{\mathrm{B,max}}$. The race-strategy then was adapted to the given conditions to reach $T_{\mathrm{B,max}}$ exactly when crossing the finish line at $s = 4.73\,\mathrm{km}$ to avoid a safety stop during the

race. This was achieved by augmented phases of lift and coast in comparison with the race-strategy for case "cold": As the requested power $P_{\Sigma,+}$ shows, the vehicle braked later before curves. Additionally, the requested power $P_{\Sigma,+}$ slowly decreased on the straights. When the driving resistances could not be overcome by $P_{\Sigma,+}$, the vehicle's velocity v_+ decreased till the entry of the next curve. Therefore, the breaks could be applied late. Still, acceleration phases overlapped in both strategies, even if they ended in different maximum velocities v. The requested power maxima differed in their absolute values. The influence on the vehicle speed v_+ in case "hot" is evident: Physically, maximum velocity was never reached and mean acceleration as well as deceleration occurred less aggressively, meaning the velocity's gradients were decreased.

The coolant fluid temperature $T_{\mathrm{F2},+}$ reached an equilibrium at the end of the optimization horizon since only as much heat was allowed to be released internally by the battery as the coolant fluid could dissipate. Coolant circuit F1 could be neglected in this case. Machine and inverter temperatures $T_{\mathrm{M},+}$ and $T_{\mathrm{I},+}$ did not reach their limits.

Another important point is the difference in the race paths to be driven in both cases (Fig. 5). In case "hot" velocity v_+ in curve 4 (marked) was higher. This per se had a positive influence on the lap time. Nevertheless, the higher curve speed v_+ required the path to change slightly within and immediately after this turn. Since the combined tire usage was already at the limit here, the race-path radius in case "hot" increased. By this, its curvature decreased and the higher speed v_+ could feasibly be driven. Nevertheless, the decreased acceleration after turn 4 led to a diminished increase of the powertrain temperatures.

The total race times cumulated over the two laps were $142.12\,\mathrm{s}$ in case "cold" and $149.59\,\mathrm{s}$ in case "hot".

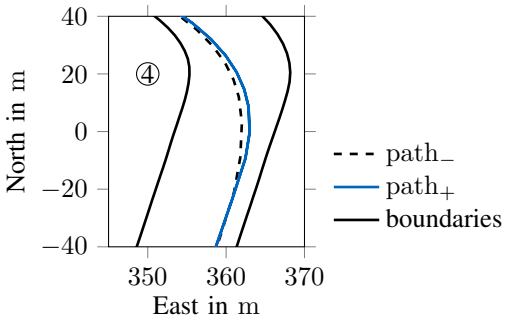


Fig. 5: Race paths for case "cold" and "hot".

IV. CONCLUSION & OUTLOOK

In the next Roborace season, the methods presented in this publication will be applied to the racecar. On the one hand, the available energy can then be used as effectively as possible. On the other hand, the powertrain components can be exploited as much as possible without loosing race time. Additionally, powertrain losses and component temperatures can then be compared with measurement data we receive when driving the proposed global race lines that were calculated considering the thermodynamical influence.

Furthermore, an additional optimization will be implemented that allows for the mentioned online re-planning of the race-strategy in the presence of disturbances. With the help of the results in this publication this simplified online optimization can be realized.

Along with these improvements the loss-models will be replaced by physically more detailed descriptions of the powertrain components.

CONTRIBUTIONS

T. H. initiated the idea of the paper and contributed significantly to the concept, modeling, and results. F. P. modeled the powertrain thermodynamics in his Master's thesis. J. B. contributed to the whole concept of the paper. M. L. provided a significant contribution to the concept of the research project. He revised the paper critically for important intellectual content. M. L. gave final approval for the publication of this version and is in agreement with all aspects of the work. As a guarantor, he accepts responsibility for the overall integrity of this paper.

ACKNOWLEDGMENT

We would like to thank the Roborace team for giving us the opportunity to work with them and for the use of their vehicles for our research project. Special thanks to Ollie Walsh for generously sharing measurement data. We would also like to thank the Bavarian Research Foundation (Bayerische Forschungsstiftung) for funding us in connection with the "rAIcing" research project.

REFERENCES

[1] J. Betz, A. Wischnewski, Heilmeier, A., Nobis, F., T. Stahl, L. Hermansdorfer, T. Herrmann, and M. Lienkamp, "A Software Architecture for the Dynamic Path Planning of an Autonomous Racecar at the Limits of Handling," in *8th IEEE International Conference on Connected Vehicles and Expo (ICCVE 2019)*. IEEE, In Press, 2019.

[2] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, T. Stahl, L. Hermansdorfer, B. Lohmann, and M. Lienkamp, "What can we learn from autonomous level-5 motorsport?" in *9th International Munich Chassis Symposium 2018*, ser. Proceedings, P. Pfeffer, Ed. Wiesbaden: Springer Fachmedien Wiesbaden, 2019, pp. 123–146.

[3] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, T. Stahl, L. Hermansdorfer, and M. Lienkamp, "A Software Architecture for an Autonomous Racecar," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, 2019, pp. 1–6.

[4] Chair of Automotive Technology, "Autonomous Driving Control Software of TUM Roborace," 2019. [Online]. Available: https://github.com/TUMFTM/veh_passenger

[5] T. Herrmann, F. Christ, J. Betz, and M. Lienkamp, "Energy Management Strategy for an Autonomous Electric Racecar using Optimal Control," in *2019 IEEE 22th International Conference on Intelligent Transportation Systems (ITSC)*. Piscataway, NJ: IEEE, 2019.

[6] J. h. Jeon, R. V. Cowlagi, S. C. Peters, S. Karaman, E. Frazzoli, P. Tsiotras, and K. Iagnemma, "Optimal Motion Planning with the Half-Car Dynamical Model for Autonomous High-Speed Driving," in *American Control Conference (ACC)*. Piscataway, NJ: IEEE, 2013, pp. 188–193.

[7] A. Carvalho, Y. Gao, A. Gray, E. Tseng, and F. Borrelli, "Predictive Control of an Autonomous Ground Vehicle using an Iterative Linearization Approach," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2013*. Piscataway, NJ: IEEE, 2013.

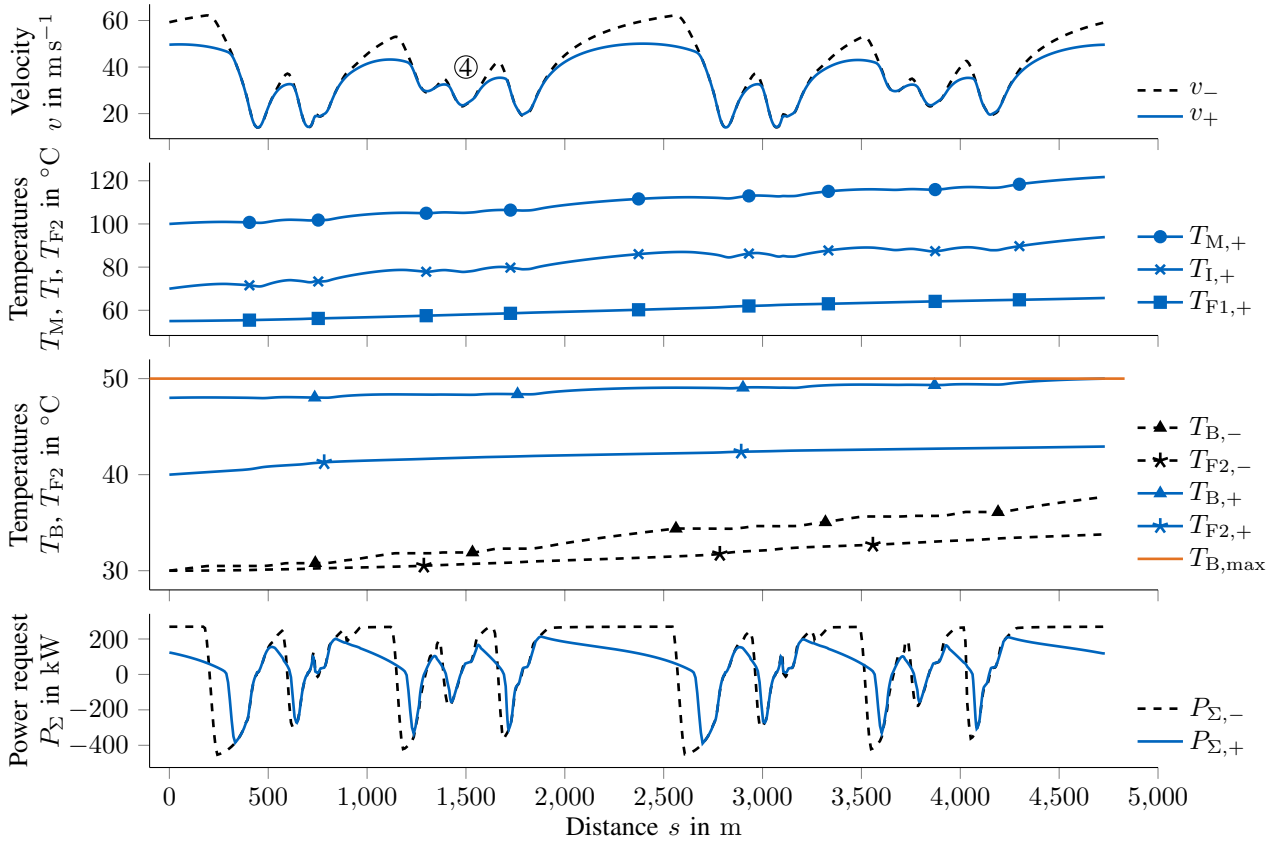


Fig. 6: Optimal race strategies for two laps on the Monteblanco race circuit showing velocity v, power request P_Σ and component temperatures T_c.

[8] S. Ebbesen, M. Salazar, P. Elbert, C. Bussi, and C. H. Onder, "Time-optimal Control Strategies for a Hybrid Electric Race Car," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 233–247, 2018.

[9] C. Kirches, S. Sager, H. G. Bock, and J. P. Schlöder, "Time-optimal control of automobile test drives with gear shifts," *Optimal Control Applications and Methods*, vol. 31, no. 2, pp. 137–153, 2010.

[10] D. J. N. Limebeer and G. Perantoni, "Optimal Control of a Formula One Car on a Three-Dimensional Track—Part 2: Optimal Control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 5, p. 051019, 2015.

[11] M. Imani Masouleh and D. J. N. Limebeer, "Optimizing the Aero-Suspension Interactions in a Formula One Car," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 3, pp. 912–927, 2016.

[12] M. Salazar, P. Elbert, S. Ebbesen, C. Bussi, and C. H. Onder, "Time-optimal Control Policy for a Hybrid Electric Race Car," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 6, pp. 1921–1934, 2017.

[13] D.-m. Wu, Y. Li, C.-q. Du, H.-t. Ding, Y. Li, X.-b. Yang, and X.-y. Lu, "Fast velocity trajectory planning and control algorithm of intelligent 4WD electric vehicle for energy saving using time-based MPC," *IET Intelligent Transport Systems*, vol. 13, no. 1, pp. 153–159, 2019.

[14] Roborace, "Roborace Ltd." 2018. [Online]. Available: https://roborace.com/

[15] J. Schuetzhold and W. Hofmann, "Analysis of the Temperature Dependence of Losses in Electrical Machines," in *IEEE Energy Conversion Congress and Exposition (ECCE), 2013*. Piscataway, NJ: IEEE, 2013, pp. 3159–3165.

[16] K. Li, S. Wang, and J. P. Sullivan, "A novel thermal network for the maximum temperature-rise of hollow cylinder," *Applied Thermal Engineering*, vol. 52, no. 1, pp. 198–208, 2013.

[17] T. Stahl, A. Wischnewski, J. Betz, and M. Lienkamp, "Multilayer Graph-Based Trajectory Planning for Race Vehicles in Dynamic Scenarios," in *2019 IEEE 22th International Conference on Intelligent Transportation Systems (ITSC)*. Piscataway, NJ: IEEE, 2019.

[18] T. Herrmann and M. Luethy, "A Real-Time Simulation Environment for Autonomous Vehicles in Highly Dynamic Driving Scenarios," 2019. [Online]. Available: https://de.mathworks.com/videos/a-real-time-simulation-environment-for-autonomous-vehicles-in-highly-dynamic-driving-scenarios-1558698299193.html

[19] A. Sciarretta and A. Vahidi, *Energy-Efficient Driving of Road Vehicles: Toward Cooperative, Connected, and Automated Mobility*, 1st ed., ser. Lecture Notes in Intelligent Transportation and Infrastructure. Cham: Springer International Publishing and Springer, 2020.

[20] T. Han, B. Khalighi, E. C. Yen, and S. Kaushik, "Li-Ion Battery Pack Thermal Management: Liquid Versus Air Cooling," *Journal of Thermal Science and Engineering Applications*, vol. 11, no. 2, p. 5115, 2019.

[21] A. M. EL-Refaie, N. C. Harris, T. M. Jahns, and K. M. Rahman, "Thermal Analysis of Multibarrier Interior PM Synchronous Machine Using Lumped Parameter Model," *IEEE Transactions on Energy Conversion*, vol. 19, no. 2, pp. 303–309, 2004.

[22] F. P. Incropera, D. P. DeWitt, T. L. Bergman, and A. S. Lavine, *Fundamentals of heat and mass transfer*, 6th ed. Hoboken, NJ: Wiley, 2007.

[23] D. P. Bertsekas, *Dynamic programming and optimal control*, 4th ed., ser. Athena scientific optimization and computation series. Belmont, Mass.: Athena Scientific, 2016, vol. 3.

[24] S. P. Boyd and L. Vandenberghe, *Convex optimization*, ser. Safari Tech Books Online. Cambridge, UK and New York: Cambridge University Press, 2004.

[25] F. Christ, A. Wischnewski, A. Heilmeier, and B. Lohmann, "Time-optimal trajectory planning for a race car considering variable tyre-road friction coefficients," *Vehicle System Dynamics*, vol. 3, no. 1, pp. 1–25, 2019.

[26] J. A. E. Andersson, J. Gillis, and G. Horn, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, 2018.

5 Energy Strategy Core Module (Global, Online) – JCAV 2022

Summary

Maximum performance by an all-electric vehicle on the race track can only be achieved using a global ES, which is capable of being dynamically recomputed in real time. Doing so, it must consider the current vehicle state, which will slightly differ compared to the predictions by the ES Core Module, due to vehicle dynamics control errors, external disturbances, or unforeseen overtaking maneuvers. In this work, we therefore present the ES Core Module, which solves an MRTP online for an optimization horizon spanning the remaining race distance (Section 3). It considers the large rates of change in the driving dynamic states of the race car, and the slowly reacting thermodynamic and energetic powertrain states. The algorithm allows to finish a race in minimum time from an energetic point of view, whilst leveraging the available battery energy in the most lap-time-efficient way. Simultaneously, temperature limitations are not exceeded and, therefore, safety shutdowns avoided. To summarize, this publication presents an online-capable algorithm which adheres to the combination of the energetic and thermodynamic constraints presented in Chapter 4. To achieve small prediction errors by the governing ODEs, we use efficient approximations of the physically exact descriptions of the powertrain component behaviors.
In this publication, we also take up the three levels of the proposed ES in Section 4.2 [3] and explain this concept with a focus on the dynamic modules, which are the ES Core Module (this chapter) and the Velocity Optimization Module (Section 6) [4]. It is important to recapitulate that the ES Core Module forwards power limitations, which summarize the energetic and thermodynamic constraints, to the velocity planner. Thus, the Velocity Optimization Module acts as an interface to the outer loop of the autonomous driving software stack (Figure 3.1) and thereby realizes the ES.

Prior to the presentation of our algorithm, a detailed literature review reveals that MLTPs, but not MRTPs, were solved by many studies before [165, 166, 169, 170]. Optimal control strategies for hybrid powertrain architectures are considered in [206–208], the powertrain behavior of all-electric race cars has been taken into account only most recently in [172, 173, 177, 178].

The underlying algorithm in the ES Core Module is formulated as a space-dependent, nonlinear OCP. It comprises the vehicle velocity, the SOC, and the component temperatures as free states together with the acceleration and brake force as control variables. The differential algebraic equation (DAE) system is completed by the lethargy, an algebraic variable denoting the time spent per traveled distance. The constraints arising from the driving dynamics are formulated for a point mass model. Additionally, based on the results in Section 4.2 [3], we entirely remove the path geometry from the optimization problem to keep the computation time low.

The mathematical description of the powertrain comprises a battery loss model [50, p. 51], including an SOC-dependent OCV and a temperature-variable internal cell resistance [231].

The VSI losses are accumulated on the basis of a semiconductor level [81, 87, 92, 93]. To also integrate the PMSM energy dissipation, their stator-iron and rotor, as well as temperature-dependent copper losses are formulated [92, 112] (Section 2.1). We mainly parameterize these models by datasheet numbers, and validate them against real-world telemetry measurement data collected during competition runs with the race car DevBot 2.0. We compare the measured and calculated battery output power, the SOC, the temperatures of the battery and the PMSM. The power losses can be translated into temperature contributions reusing the ODEs deduced in Section 4.2 [3].

We formulate the MRTP with the acados framework [150, 151]. It is transcribed to an NLP via the multiple shooting discretization technique and solved by an SQP method. As an integration scheme, we chose the implicit Runge Kutta method of the 4th order. The underlying QPs are handled by the HPIPM [157]. The SQP method was chosen to achieve real-time capability, since we solve a sequence of neighboring problems [138, p. 555]. To mitigate convergence difficulties, we divide the solution procedure into the three steps “ES Guess”, “ES Presolve” and “ES Resolve” (Figure 5.1). The ES Guess calculates a time-optimal reference speed profile for a single lap, which is only constrained by the driving dynamics. It is subsequently used as an initial guess for the step ES Presolve, which is computed before a race start to provide a first ES solution. During the race, we iteratively call the ES Resolve step, which dynamically reacts to the current vehicle state and adapts the global race strategy for a lookahead distance of the remaining race laps. The ES Resolve step achieves fast solver convergence, since previous solutions, which have been calculated several hundred meters before the current solver call, are close to the optimal solution of the current iteration.

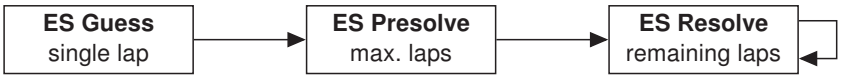


Figure 5.1: Solution procedure of the ES.

Exemplary, we show results for an MRTP consisting of 12 race laps on the Monteblanco (Spain) circuit in Figure 5.2. At the beginning of the race, the optimized strategy suggests to heat up the PMSMs as fast as possible to increase its cooling power. This is achieved by a vehicle speed, which almost equals the reference velocity profile. In the second half of the race, the vehicle keeps its velocity pattern constant on a reduced absolute level to save energy. When crossing the finish line after approx. 30 km, the SOC reaches $0\,\%$ and the PMSM its temperature maximum. A second experiment addresses the computation times of the ES Core Module. We conduct several calls of the ES Resolve step for different race lengths and circuits. Optimization horizons of up to 80 km are handled in less than 15 s of computation. We consider this computation time to be small enough for an online application, since the global strategy can be adapted several times during a single race lap, which is in accordance with the relatively slowly varying thermodynamics of the powertrain.

To validate the output of the ES Core Module, we compare its results to MLTP solutions from the literature [165], and the ones obtained in Chapter 4, which shows a small normalized root mean square error (NRMSE). In a further step, we directly solve the optimization problem of the ES Core Module using IPOPT [152]. This experiment proves that the SQP method converges to the same results calculated by the nonlinear IP method with a numerical suboptimality of less than one percent. However, the SQP approach is one order of magnitude faster.

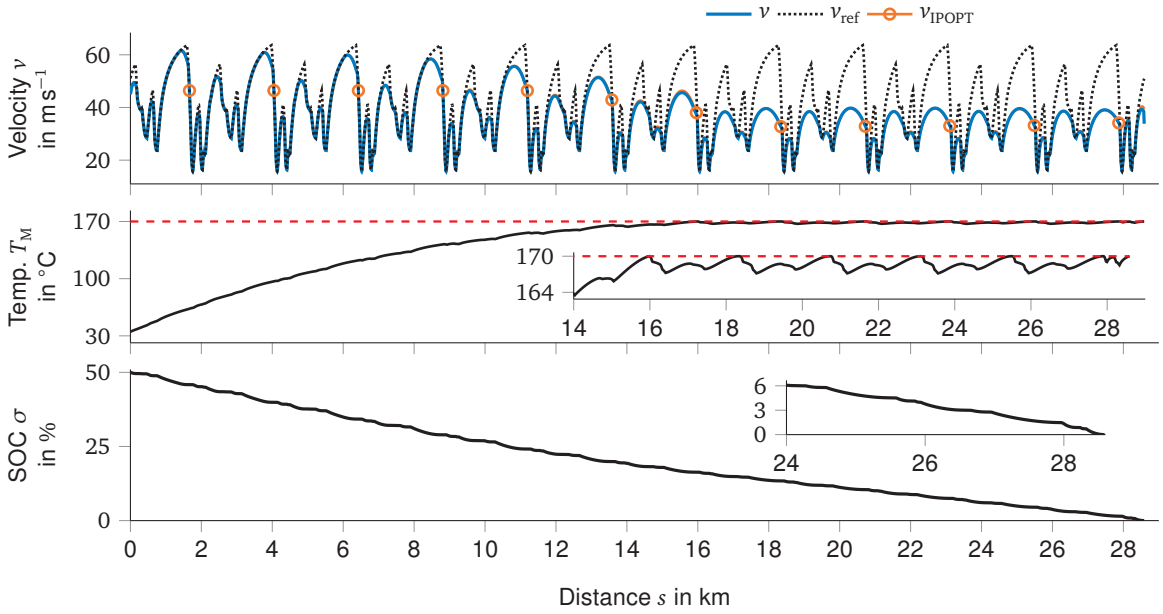


Figure 5.2: Minimum Race Time Strategy for 12 consecutive race laps on the Monteblanco (Spain) circuit, © 2022 SAE/The Authors.

Contributions

T. H. initiated the idea of the paper and contributed significantly to the concept, modeling, and results. F. S. and M. B. contributed to the powertrain modeling and the OCP in their master's theses, J. B. to the whole concept of the paper. M. L. provided a significant contribution to the concept of the research project. He revised the paper critically for important intellectual content. M. L. gave final approval for the publication of this version and is in agreement with all aspects of the work. As a guarantor, he accepts responsibility for the overall integrity of this paper.

Copyright notice

ARTICLE INFO
Article ID: 12-05-01-0005

doi:10.4271/12-05-01-0005

Optimization-Based Real-Time-Capable Energy Strategy for Autonomous Electric Race Cars

Thomas Herrmann,[1] Florian Sauerbeck,[1] Maximilian Bayerlein,[1] Johannes Betz,[2] and Markus Lienkamp[1]

[1]Technical University of Munich, Germany; School of Engineering & Design, Department of Mobility Systems Engineering, Institute of Automotive Technology, Germany
[2]University of Pennsylvania, USA; Department of Electrical & Systems Engineering, Real-Time and Embedded Systems Lab (mLab), USA

Abstract

Solving a Minimum Lap Time Problem (MLTP), under the constraints stemming from a race car's driving dynamics, can be considered to be state of the art. Nevertheless, when dealing with electric race vehicles as in Formula E or the Roborace competition, solving an MLTP is not enough to form an appropriate competition strategy: Maximum performance over the entire race can only be achieved by an optimization horizon spanning all the subsequent laps of a race. This results in a Minimum Race Time Problem (MRTP). To solve this, the thermodynamic and energetic limitations of the electric powertrain components must be taken into account, as exceeding them leads to safety shutdowns. Therefore, we present an Optimal Control Problem (OCP) to calculate an Energy Strategy (ES) for electric race cars, which contain physically detailed descriptions of its powertrain components. Leveraging a Sequential Quadratic Programming (SQP) solver, the OCP can be solved numerically in real time. This enables the ES to be recalculated during a race. As a consequence, powertrain overheating can be omitted and the limited amount of stored battery energy utilized as efficiently as possible. Simultaneously, the race can be completed in minimum time.

History

Received: 04 Feb 2021
Revised: 02 Apr 2021
Accepted: 20 Dec 2021
e-Available: 10 Jan 2022

Keywords

Real-time numerical optimization, Optimal control, Energy strategy, Minimum race time planning, Autonomous electric vehicles

Citation

Herrmann, T., Sauerbeck, F., Bayerlein, M., Betz, J. et al., "Optimization-Based Real-Time-Capable Energy Strategy for Autonomous Electric Race Cars," *SAE Int. J. of CAV* 5(1):2022, doi:10.4271/12-05-01-0005.

ISSN: 2574-0741
e-ISSN: 2574-075X

This article is part of a Special Issue on Autonomy and Connectivity at the Edge—Autonomous Racing.

Introduction

Research into real-time-capable Energy Strategies (ES) is attracting growing attention. The reasons for this development are the evolution of electric vehicles on the road and fully electric motorsport series like Formula E or Roborace [1]; see Figure 1. In motorsports applications, an ES is crucial to be able to complete a race without exceeding the thermal or energetic constraints arising from the all-electric powertrain architecture. However, a time-optimal result can only be achieved when leveraging the maximum possible power. To address both opposing objectives, this article presents an online-capable ES optimization for electric cars solving a Minimum Race Time Problem (MRTP). It takes into account the technical constraints arising from the driving dynamics, the powertrain thermodynamics, and the available amount of energy while minimizing the total racing time. We use the High-Performance Interior-Point Method (HPIPM) solver [2] interfaced by acados [3] to achieve recalculation results in short computation times and compare the results obtained to the well-known general Nonlinear Programming (NLP) solver Interior Point OPTimizer (IPOPT) [4], interfaced by CasADi [5]. By this, we validate the numerical optimality of the solutions obtained by our problem formulation in combination with HPIPM.

State of the Art

A lot of research has already been conducted in the field of global race trajectory optimization. These Minimum Lap Time Problems (MLTP) have often been formulated as an Optimal Control Problem (OCP). For reader's reference, we wish to mention a few: Limebeer and Perantoni [6], Tremlett and Limebeer [7], Christ et al. [8], and Dal Bianco et al. [9]. The implemented optimizations are based on the assumption that no thermodynamic constraints from the powertrain become active. This is valid for Internal Combustion Engine (ICE)-powered race cars or a short optimization horizon of a single race lap. Nevertheless, with the evolution of the electric racing series, the existing global trajectory optimization algorithms must be extended by the powertrain thermodynamics and be capable of solving the MLTP for multiple subsequent race laps. By this, possible lap time losses from unwanted power restrictions can be omitted.

As one of the first approaches, Liu and Fotouhi [10] use Artificial Neural Networks (ANN) and Monte Carlo Tree Search (MCTS) methods to deduce a race strategy for prerace planning and in-race recalculations in Formula E. They consider the battery temperature using a Bernardi model and its State of Charge (SOC) as optimization variables. Results are achieved in several seconds of computation time for 32 race laps. In contrast, the necessary data generation process using simulation, and training the ANN, takes days. Further extensions of the presented approach by additional physical equations require a computationally intensive retraining phase.

A more flexible approach to solving an MLTP is presented by the same authors [11], using optimal control leveraging the General Purpose OPtimal Control Software (GPOPS-II) [12] as a solver. Results are shown for different energetic and thermal constraints, which become active during their experiments in simulation. Calculation times are not given. Therefore statements about the real time capability of the proposed algorithm cannot be made.

A partial sequential Second-Order Conic Programming (SOCP) formulation is presented by Borsboom et al. [13], formulating a quasi-convex MLTP for a battery electric race car. They consider battery, electric motor, and transmission losses. For an optimization horizon of approximately 14 km, their algorithm's runtime is around one minute. Still a thermal description of the powertrain components is missing as the thermodynamics have a major influence on the race time performance [14, 15].

Salazar et al. [16] state that the control algorithm for a hybrid electric powertrain in Formula 1 strongly influences the achievable lap time. Therefore, they design a feedback control approach based on ideas from Equivalent Consumption Minimization Strategies (ECMS), which can track and adapt an offline generated energy management strategy online. By this, their algorithm reacts to external stochastic disturbances in a lap-time optimal way. Results are shown for subsequent race laps, but thermal models are not required for the problem in this publication due to the hybrid powertrain architecture.

For the same powertrain, Salazar et al. [17] derive an analytic ES for the optimal split of power delivered by the electric motors or the combustion engine. For this, they leverage Pontryagin's Minimum Principle (PMP) and nonsmooth analysis. Different operating modes for the car, being in power-limited or grip-limited sections of the track, are deduced. To further enhance this power split strategy, feedback controllers are introduced [18] by setting up a two-level Model Predictive Control (MPC) scheme to recalculate the ES in real time. The upper-level MPC consists of an OCP with a look-ahead of a single lap. This is solved using ECOS [19] within a second of calculation time. The low-level MPC is designed to track the optimal state trajectories found by the high-level controller. The Linear Programming (LP) problem in the low-level controller can be solved within several milliseconds by FORCES Pro [20]. The fuel and the battery energy consumption per race lap are considered to be tuning parameters. In contrast, when solving an MRTP with a look-ahead distance of the remaining race distance, the optimal, variable energy consumption per lap could be calculated.

Several authors set up an ES to find the optimal velocity profile for a solar race car [21, 22, 23]: Cifuentes and Pradenas [23] use algorithms like iterated local search, simulated annealing, and genetic algorithms leading to calculation times in a magnitude of several seconds for approximately 50 spatial discretization points. Merino and Duarte-Mermoud [22] propose a multistage optimization method to solve the MRTP

to account for the long planning horizons of several race days in combination with calculations on a continuous level. They include a nonlinear loss model of the Brushless Direct Current (BLDC) motor. Using a pseudo-spectral method in the resulting OCP, computation times of 1.5 min on a dual-core processor using GPOPS [24] for a prediction horizon comprising an entire race are achieved. Atmaca [21] calculates the ES taking into account the BLDC motor losses, which are assumed to be quadratically proportional to the motor's coil current, and the battery efficiency, which is assumed to be constant. The resulting NLP to find the minimum lap time velocity is solved using MATLAB's fmincon. This yields a computation time of half a minute for a circuit discretized by approximately 200 segments with a segment length of 2 m to 16 m.

Contributions

In this work we present the following contributions:

1. An OCP, which is solved by a Sequential Quadratic Programming (SQP) method [25] in real time to find an ES for electric race cars, is formulated. The optimization problem includes state-of-the-art equations and approximations, describing the vehicle's driving and detailed powertrain thermodynamics, to take the corresponding technical limitations into account. The formulation of the OCP enables a suitable compromise between a small physical error and a low computational complexity.
2. We present a methodology to solve the resulting NLP for long planning horizons, comprising all the subsequent race laps efficiently. With this method, an optimal solution to the formulated MRTP, regarding the comprehensive optimization horizon, can be found.
3. We include power loss models, which are based on physical descriptions and are mainly parametrized by datasheet values, in the OCP. These loss models span
   - A description of the electric motor's stator-iron and rotor, as well as copper losses
   - The inverter's switching and conduction losses
   - A battery model with an SOC-dependent Open-Circuit Voltage (OCV) and a temperature-dependent internal cell resistance to determine battery loss power
4. For validation purposes, we compare the presented powertrain equations to real-world measurement data, collected with the Roborace DevBot 2.0 during competition runs; see Figure 1. To further prove numerical optimality, we compare the results of our ES, which is discretized using the multiple shooting method method and solved by HPIPM, to a state-of-the-art MLTP planner [8, 14], which uses direct orthogonal collocation and IPOPT.

FIGURE 1 Roborace DevBot 2.0 on the racetrack.

Preliminaries

A continuous OCP in a general space-dependent form can be written as [3]

$$\min \int_0^S l\big(\boldsymbol{x}(s),\boldsymbol{z}(s),\boldsymbol{u}(s)\big)+ \frac{1}{2}\begin{bmatrix}\boldsymbol{s}_\mathrm{u}(s)\\ 1\end{bmatrix}^T\begin{bmatrix}\boldsymbol{Z}_\mathrm{u} & \boldsymbol{z}_\mathrm{u}\\ \boldsymbol{z}_\mathrm{u}^T & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{s}_\mathrm{u}(s)\\ 1\end{bmatrix}\mathrm{d}s+ + M\big(\boldsymbol{x}(S)\big)+\frac{1}{2}\begin{bmatrix}\boldsymbol{s}_\mathrm{u}(S)\\ 1\end{bmatrix}^T\begin{bmatrix}\boldsymbol{Z}_{\mathrm{u,f}} & \boldsymbol{z}_{\mathrm{u,f}}\\ \boldsymbol{z}_{\mathrm{u,f}}^T & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{s}_\mathrm{u}(S)\\ 1\end{bmatrix} \quad \text{Eq. (1a)}$$

$$\text{s.t.} 0 = \boldsymbol{f}\big(\dot{\boldsymbol{x}}(s),\boldsymbol{x}(s),\boldsymbol{z}(s),\boldsymbol{u}(s)\big) \quad \text{Eq. (1b)}$$

$$0 \geq \boldsymbol{g}\big(\boldsymbol{x}(s),\boldsymbol{z}(s),\boldsymbol{u}(s)\big). \quad \text{Eq. (1c)}$$

Here $\boldsymbol{x} \in \mathbb{R}^{n_x \times 1}$ represents the state vector, $\dot{\boldsymbol{x}} \in \mathbb{R}^{n_x \times 1}$ its derivative, $\boldsymbol{z} \in \mathbb{R}^{n_z \times 1}$ the algebraic variable vector, and $\boldsymbol{u} \in \mathbb{R}^{n_u \times 1}$ the collection of control inputs. The cost function includes a Lagrange term $l \in \mathbb{R}$ and a Mayer term $M \in \mathbb{R}$, for the running and the terminal cost, respectively. The system dynamics are given by a set of Differential-Algebraic Equations (DAE) in $\boldsymbol{f} \in \mathbb{R}^{n_f \times 1}$. The constraints are denoted by $\boldsymbol{g} \in \mathbb{R}^{n_g \times 1}$. The horizon length of the free-space variable s is fixed, i.e., $s \in [0, S]$. Additional slack variables $\boldsymbol{s}_\mathrm{u}(s) \in \mathbb{R}^{n_s \times 1}$ can be used to soften the upper bounds of hard constraints. Here matrices $\boldsymbol{Z}_\mathrm{u} \in \mathbb{R}^{n_s \times n_s}$ penalize the quadratic and vectors $\boldsymbol{z}_\mathrm{u} \in \mathbb{R}^{n_s \times 1}$ the linear slack variable values, as the exact penalty theory states [26].

In acados [3], OCPs are discretized using the multiple shooting method to obtain an NLP, which can be solved numerically. In embedded optimization, it is common to write the discrete objective function as a least-squares equation [3]. Therefore, the Lagrange and Mayer terms transform into Equations 2a and 2b. Their entries are additionally weighted

by the diagonal matrices $\boldsymbol{W} \in \mathbb{R}^{n_\Sigma \times n_\Sigma}$ and $\boldsymbol{W}_K \in \mathbb{R}^{n_x \times n_x}$, where $n_\Sigma = n_x + n_u + n_z$:

$$l(\boldsymbol{x}_k, \boldsymbol{z}_k, \boldsymbol{u}_k) = \left\| \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_k \\ \boldsymbol{z}_k \end{bmatrix} - \begin{bmatrix} \boldsymbol{x}_{k,\mathrm{r}} \\ \boldsymbol{u}_{k,\mathrm{r}} \\ \boldsymbol{z}_{k,\mathrm{r}} \end{bmatrix} \right\|_W^2, \qquad \text{Eq. (2a)}$$

$$M(\boldsymbol{x}_K) = \left\| \boldsymbol{x}_K - \boldsymbol{x}_{K,\mathrm{r}} \right\|_{W_K}^2, \qquad \text{Eq. (2b)}$$

where k denotes a discrete position index in s, running till K. The reference value of the respective variable is denoted by the index r.

As an error metric, we will use the Normalized Root Mean Square Error (NRMSE) throughout this article. By this, the given errors are easily interpretable and convertible to absolute values:

$$\mathrm{NRMSE} = \frac{\sqrt{\frac{\sum_i^N (\hat{y}_i - y_i)^2}{N}}}{y_{\max} - y_{\min}}. \qquad \text{Eq. (3)}$$

Here y and $\hat{y}$ are the compared signals with N data points and their corresponding maximum and minimum values indicated by the indices "max"/"min."

Powertrain Modeling

The powertrain of the race car, which is based on the Roborace DevBot 2.0, is given in Figure 2. It consists of mechanical parts like wheels ($\mathrm{W_{l/r}}$) and gears ($\mathrm{G_{l/r}}$) at the rear left (l) and right (r) and electrical components like electric motors ($\mathrm{M_{l/r}}$), inverters ($\mathrm{I_{l/r}}$), battery (B), and auxiliary devices ($\mathrm{A_x}$). The radiators $\mathrm{R_{MI}}$ and $\mathrm{R_B}$ are necessary to remove the heat generated by internal power losses inside the electrical components. Convective heat transfer is realized by two cooling liquids, whose temperatures are denoted by $T_{\mathrm{CM}}(s)$ and $T_{\mathrm{CB}}(s)$. The second subscripts at the temperature symbols denote a location in the cooling circuit, where the liquids enter a certain powertrain component.

FIGURE 2 Architecture of the race car's powertrain, compare [14].

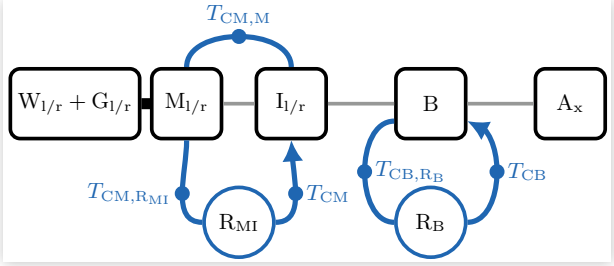
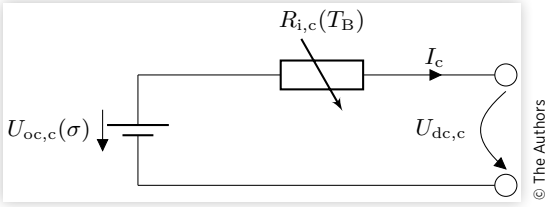


FIGURE 3 Electric equivalent circuit model of a Lithium-Ion cell with the OCV $U_{\mathrm{oc,c}}$ being dependent on the battery SOC σ and its internal resistance $R_{\mathrm{i,c}}$, which correlates with the temperature T_{B}.



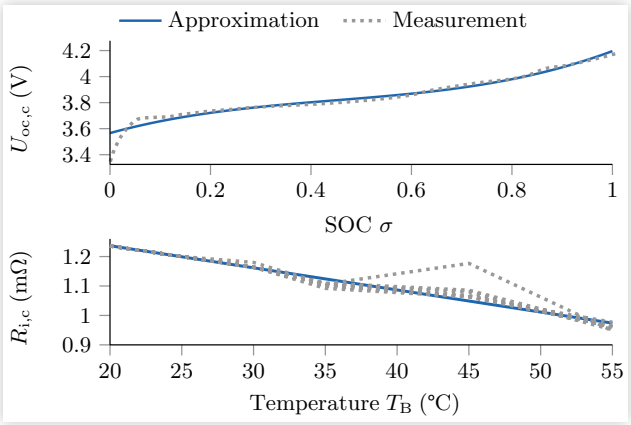
Power Loss Models

To represent the battery pack, an electric equivalent circuit for a single cell is modeled (see Figure 3) and scaled to fit an $N_{\mathrm{s}}N_{\mathrm{p}}$ (series/parallel) pack configuration. The equivalent circuit consists of an SOC-dependent voltage source $U_{\mathrm{oc,c}}(\sigma(s))$ and the cell's temperature-dependent internal resistance $R_{\mathrm{i,c}}(T_{\mathrm{B}}(s))$ [27]. To mathematically integrate these dependencies into an OCP, we approximate their datasheet measurements: For the correlation of $U_{\mathrm{oc,c}}(\sigma)$, we assume a third-degree polynomial, which results in an NRMSE of 3.15%; see Figure 4 (top). Similarly, the correlation of the internal cell resistance $R_{\mathrm{i,c}}$ and the temperature T_{B} is modeled as a linear correlation; see Figure 4 (bottom), with an NRMSE of 9.01% in the shown operating range. Multiple measurement values of the internal resistance $R_{\mathrm{i,c}}$ for the same temperature T_{B} represent different SOC levels.

The electrical cell variables scaled to pack level are

$$\begin{aligned} U_{\mathrm{oc}}(s) &= U_{\mathrm{oc,c}}(s) N_{\mathrm{s}}; \quad U_{\mathrm{dc}}(s) = U_{\mathrm{dc,c}}(s) N_{\mathrm{s}}; \\ R_{\mathrm{i}}(s) &= R_{\mathrm{i,c}}(s) \frac{N_{\mathrm{s}}}{N_{\mathrm{p}}}; \quad I_{\mathrm{B}}(s) = I_{\mathrm{c}}(s) N_{\mathrm{p}}. \end{aligned} \qquad \text{Eq. (4)}$$

FIGURE 4 Dependency of the battery cell OCV $U_{\mathrm{oc,c}}$ on its SOC σ, and $R_{\mathrm{i,c}}(T_{\mathrm{B}})$ correlation. The NRMSE for the third-order $U_{\mathrm{oc,c}}(\sigma)$ approximation is 3.15% and for the linear correlation $R_{\mathrm{i,c}}(T_{\mathrm{B}})$ 9.01%.



The battery pack terminal voltage $U_{dc}(s)$ can be written with the pack current $I_B(s)$ as

$$U_{dc}(s) = U_{oc}(\sigma(s)) - R_i(T_B(s))\, I_B(s). \quad \text{Eq. (5)}$$

To calculate the internal battery pack power $P_{i,B}(s)$, the following nonlinear equation can be applied [28]:

$$P_{i,B}(P_{o,B}(s)) = \frac{U_{oc}^2(\sigma(s))}{2R_i(T_B(s))} - U_{oc}(\sigma(s))\frac{\sqrt{U_{oc}^2(\sigma(s)) - 4P_{o,B}(s)R_i(T_B(s))}}{2R_i(T_B(s))}. \quad \text{Eq. (6)}$$

The battery output power $P_{o,B}(s)$ is the sum of the power losses of the inverters $P_{l,I}(s)$, electric motors $P_{l,M}(s)$, and the requested wheel power $P_W(s)$:

$$P_{o,B}(s) = P_W(s) + 2P_{l,I}(s) + 2P_{l,M}(s). \quad \text{Eq. (7)}$$

Through this, one obtains the battery power loss as

$$P_{l,B}(s) = P_{i,B}(s) - P_{o,B}(s). \quad \text{Eq. (8)}$$

The inverter losses are modeled on a semiconductor level and follow the expressions [29, 30, 31]

$$P_{l,I,swi}(s) = 3\frac{I_e(s)}{I_r}\frac{U_{dc}(s)}{U_r}(E_n + E_f + E_r) f_{swi}, \quad \text{Eq. (9a)}$$

$$P_{l,I,con}(s) = 3I_e(s)\underbrace{\left(\hat{U}_{ce} + \bar{U}_{ce} I_e(s)\right)}_{U_{ce}(I_e(s))}, \quad \text{Eq. (9b)}$$

where $P_{l,I,swi}(s)$ denotes the switching and $P_{l,I,con}(s)$ the conduction losses, which form the inverter losses $P_{l,I}(s)$ through summation [32, 33]. The reference voltage U_r and current I_r are datasheet parameters, together with the switching energy losses when turning on E_n and off E_f, and the reverse recovery energy E_r [29] as well as the nominal switching frequency f_{swi}. The description of the conduction losses in Equation 9b is valid for Insulated-Gate Bipolar Transistor (IGBT) inverters [33, 34]. The dependency of the collector-emitter voltage $U_{ce}(I_e(s))$ as part of the datasheet can be assumed to be linear [29] for the inverters in the race car (DevBot 2.0), where $\hat{U}_{ce}$ is the approximation's offset and $\bar{U}_{ce}$ its slope.

The power loss $P_{l,M}(s)$ in the electric motor consists of current-dependent copper $P_{l,M,cop}(s)$ [32, 35] and stator iron and rotor losses $P_{l,M,sir}(s)$ [36], which are dependent on the rotational frequency ω_M,

$$P_{l,M,cop}(s) = \frac{3}{2} I_e^2(s)\left(\alpha_c\left(T_M(s) - 20°C\right) + 1\right) R_p \quad \text{Eq. (10a)}$$

$$P_{l,M,sir}(s) = f(\omega_M, c_{1..5}). \quad \text{Eq. (10b)}$$

In Equation 10a, the temperature-dependent copper losses $P_{l,M,cop}(s)$ are calculated using the known ohmic phase resistance of the motor windings at room temperature R_p and the linear temperature coefficient of resistance of copper α_c. Apart from the copper losses $P_{l,M,cop}(s)$, the stator iron and rotor losses $P_{l,M,sir}(s)$ can only be approximated by recorded measurement data as their analytic description is highly complex and part of current research [37]. Therefore, we analyzed recorded telemetry measurement data where we subtracted the recorded mechanical powertrain power $P_W(s)$ from the recorded battery output power $P_{o,B}(s)$ and reduced the result by the calculated inverter losses $P_{l,I}(s)$, cf. Equation 7. Subtraction of the motor's calculated copper losses $P_{l,M,cop}(s)$ gives the sum of stator iron and rotor losses:

$$\tilde{P}_{l,M,sir}(s) = P_{l,M}(s) - P_{l,M,cop}(s). \quad \text{Eq. (11)}$$

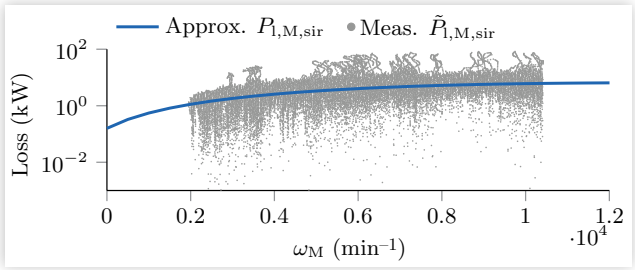
We can then fit the resulting data point cloud $\tilde{P}_{l,M,sir}(s)$ using polynomial expressions of the fourth degree $P_{l,M,sir}(s)$ with the constant parameters c_j, which are subject to identification; see Figure 5.

The effective current $I_e(s)$ through the motor-inverter combination can be calculated by the steady-state approximation [38]

$$I_e(s) = \frac{M(s)}{k_M}, \quad \text{Eq. (12)}$$

with the motor constant k_M and the motor torque $M(s)$.

FIGURE 5 Approximation and measurement of stator iron and rotor loss $P_{l,M,sir}$, plotted as the dependency of engine rotational speed ω_M. The NRMSE is 3.29%. Some negligible negative measurement values have been ignored during plotting.



Model Validation

To validate the given powertrain model equations in Section Power Loss Models, we show a comparison with real-world measurement data for a run with a duration of approximately 6.5 min with the race car DevBot 2.0 on the Monteblanco (Spain) race circuit. We assumed physically plausible values based on the literature for parameters like, e.g., the heat transfer coefficients, since they are only determinable with uncertainty.

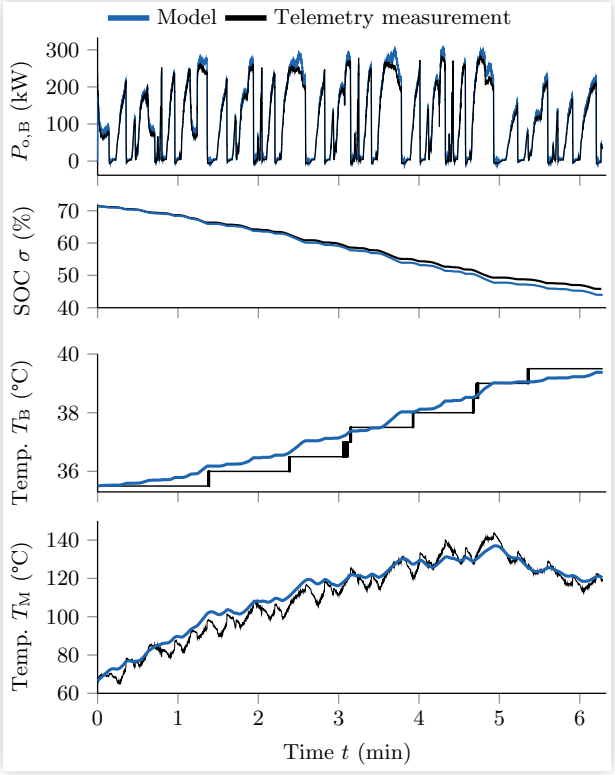
Power Losses The presented model equations were implemented in a backward simulation. Input into the simulation

was the measured motor output torque $M(t)$ and rotational speed $\omega_M(t)$, which were propagated through the powertrain model. The upper plot in Figure 6 depicts the battery output power $P_{o,B}(t)$, which was measured using its terminal voltage $U_{dc}(t)$ and current $I_B(t)$. This course was compared to the modeled battery output power $P_{o,B}(t)$ and therefore directly expresses the error in the inverter and motor loss models: The NRMSE for the battery output power $P_{o,B}(t)$ is 4.59%.

To also validate the battery loss description $P_{l,B}(t)$ in Equation 6, we compared the calculated SOC $\sigma(t)$ with the vehicle's recorded telemetry data: the deviation after the run is 1.84% of SOC; see the second plot in Figure 6.

Temperature Gradients The power losses directly influence the component temperatures. Therefore, we plot the calculated temperature curves for the most critical components in the powertrain, the battery, and the electric motor, in Figure 6 (third and fourth plots). The thermodynamic first-order Ordinary Differential Equations (ODEs), which describe the heat exchange of the powertrain components with their cooling liquids and the environment, are taken from [14] and are summarized in Equations 22a-22e. We reach an NRMSE for the thermodynamic battery model of 8.51%. In the error calculation, the temperature sensor resolution of 0.5°C plays an important role: The battery temperature $T_B(t)$ increases by 4°C during the run, but sensor measurements are only discretized by nine numeric values in total, which are compared to a continuously simulated course. The motor's temperature error is smaller at 5.38%.

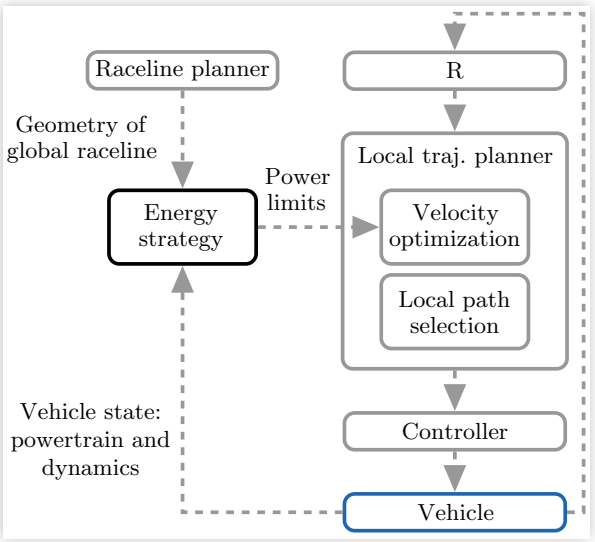
FIGURE 6 Battery output power $P_{o,B}(t)$ (first plot) and SOC $\sigma(t)$ (second plot) for a run of approximately 6.5 min with the race car DevBot 2.0 in comparison with a backward simulation through the powertrain loss models presented in Section Power Loss Models. The NRMSE for the battery output power $P_{o,B}(t)$ is 4.59%, and the SOC deviation between the vehicle's recorded telemetry data and the simulation after the entire run is 1.84%. In the thermodynamic description, NRMSEs for the battery temperature and the electric motor temperature $T_B(t)/T_M(t)$ are 8.51% and 5.38% (third and fourth plots), respectively.



Energy Strategy

First, we define the software environment the ES module is operating in: The ES module is responsible for determining the power limits for the local trajectory planner. As shown in Figure 7, it uses the powertrain state, comprising component temperatures and battery SOC, and the vehicle's dynamic state, i.e., its current position and velocity, as inputs. Additionally, the geometry of the global time-optimal raceline is necessary information for the ES module. The ES is recalculated on this path, which will most likely be the driven one during the race [14]. The ES module itself forwards optimized space-dependent power limitations to a velocity optimization module, which utilizes this information during the local trajectory planning step [39]. An update rate in the order of seconds of the ES module is sufficient in this architecture since it decouples the problem of fast trajectory planning and vehicle control [40] from the ES recalculation.

FIGURE 7 Software environment of presented ES module. Modules like "perception" or "localization" are summarized in "R" for simplicity [41].



Once an ES is available, it is valid for the remaining race distance and only needs to be updated in accordance with the slower change rate of the powertrain thermodynamics. Therefore, the necessary power limitations can independently be retrieved by the trajectory planner. With the presented software architecture, the vehicle can utilize as much power as possible, without exceeding any technical limitation, while simultaneously minimizing the entire race time. In other words, the ES module optimizes a race strategy which lets the vehicle pass the finish line with either 0% of SOC and/or a powertrain component temperature, which reached its thermodynamic limit.

Optimal Control Problem

This section defines the OCP for the ES, which must be solved in real time.

States The space-dependent state variables of the OCP, cf. Equations 1a-1c, are

$$\boldsymbol{x}(s) = \left(\upsilon\ t\ \sigma\ T_\mathrm{B}\ T_\mathrm{M}\ T_\mathrm{I}\ T_\mathrm{CM}\ T_\mathrm{CB}\right)^T \in \mathbb{R}^{n_x \times 1}, \quad \text{Eq. (13)}$$

where $\upsilon(s)$ denotes the vehicle velocity along s, $t(s)$ the travel time, $\sigma(s)$ the battery SOC, and $T_\mathrm{c}(s)$ the component temperatures. Here index c represents the battery (B), the electric motors (M), the inverters (I), the cooling liquid in the motor-inverter circuit (CM), and the battery cooling circuit (CB). The cooling liquid temperatures are simulated directly at the respective component inlets, cf. Figure 2.

As control input vector $\boldsymbol{u}$, we chose

$$\boldsymbol{u}(s) = \left(F_\mathrm{d}\ F_\mathrm{b}\right)^T \in \mathbb{R}^{n_u \times 1}, \quad \text{Eq. (14)}$$

which contains the necessary driving and braking force, $F_\mathrm{d}(s)$ and $F_\mathrm{b}(s)$, respectively.

The algebraic variable z reads

$$z(s) = \tau = \frac{1}{\upsilon} \in \mathbb{R}^{n_z \times 1}, \quad \text{Eq. (15)}$$

where $\tau(s)$ characterizes the time spent traveling one meter, called lethargy [17]. This expression can also be used to transform time-dependent DAEs, representing the dynamics of a point mass model following a fixed path, into space-dependent ones [6].

System Dynamics The system dynamics are given by a set of space-dependent DAEs Equation 1b. The time derivative reads

$$\frac{\mathrm{d}t(s)}{\mathrm{d}s} = \tau(s) = \frac{1}{\upsilon(s)}. \quad \text{Eq. (16)}$$

The space-dependent change rate of the vehicle velocity,

$$\frac{\mathrm{d}\upsilon(s)}{\mathrm{d}s} = \frac{F_\mathrm{r}(s)}{m_\mathrm{v}} \frac{\mathrm{d}t(s)}{\mathrm{d}s}, \quad \text{Eq. (17)}$$

can be written using the resulting force F_r,

$$F_\mathrm{r}(s) = F_\mathrm{d}(s) + F_\mathrm{b}(s) - F_\mathrm{c}(s) - F_\mathrm{a}(s), \quad \text{Eq. (18)}$$

which itself consists of both control inputs, the rolling resistance $F_\mathrm{c}(s)$, and the aerodynamic drag force $F_\mathrm{a}(s)$,

$$F_\mathrm{c}(s) = m_\mathrm{v} g c_\mathrm{r}, \quad \text{Eq. (19a)}$$

$$F_\mathrm{a}(s) = \frac{1}{2} \rho_\mathrm{a} c_\mathrm{w} A_\mathrm{v} \upsilon^2(s). \quad \text{Eq. (19b)}$$

Here m_v denotes the vehicle mass, g the gravitational constant, c_r the rolling resistance coefficient, ρ_a the air density, c_w the aerodynamic resistance coefficient, and A_v the vehicle front surface.

The battery SOC dynamics can be formulated as

$$\frac{\mathrm{d}\sigma(s)}{\mathrm{d}s} = -\frac{P_{\mathrm{o,B}}(s) + P_{\mathrm{l,B}}(s)}{E_\mathrm{B}} \frac{\mathrm{d}t(s)}{\mathrm{d}s}, \quad \text{Eq. (20)}$$

where E_B denotes the battery's energy capacity.

To not increase model complexity by adding the battery output current $I_\mathrm{B}(s)$ as an additional algebraic variable to the OCP, we assume

$$U_\mathrm{dc}(s) \approx U_\mathrm{oc}(\sigma(s)), \quad \text{Eq. (21)}$$

as the voltage drop over R_i in Equation 5 is small.

In the following, we summarize the temperature ODEs, describing the convective and the conductive thermodynamic power flow of the powertrain components, and refer the reader to our previous work in [14] for a more detailed derivation:

$$C_\mathrm{M} \frac{\mathrm{d}T_\mathrm{M}(s)}{\mathrm{d}t(s)} = P_{\mathrm{l,M}}(s) - \underbrace{\frac{2T_\mathrm{M}(s) - \left(T_{\mathrm{CM,M}}(s) + T_{\mathrm{CM,R_{MI}}}(s)\right)}{2R_\mathrm{M}}}_{P_{\mathrm{col,M}}(s)} \quad \text{Eq. (22a)}$$

$$C_\mathrm{I} \frac{\mathrm{d}T_\mathrm{I}(s)}{\mathrm{d}t(s)} = P_{\mathrm{l,I}}(s) - \underbrace{\frac{2T_\mathrm{I}(s) - \left(T_\mathrm{CM}(s) + T_{\mathrm{CM,M}}(s)\right)}{2R_\mathrm{I}}}_{P_{\mathrm{col,I}}(s)} \quad \text{Eq. (22b)}$$

$$C_\mathrm{B} \frac{\mathrm{d}T_\mathrm{B}(s)}{\mathrm{d}t(s)} = P_{\mathrm{l,B}}(s) - \underbrace{\frac{T_\mathrm{B}(s) - T_\mathrm{CB}(s)}{R_\mathrm{B}}}_{P_{\mathrm{col,B}}(s)} \quad \text{Eq. (22c)}$$

$$C_{CM}\frac{dT_{CM}(s)}{dt(s)} = 2P_{col,M}(s) + 2P_{col,I}(s) - \\ -\frac{1}{R_{MI}}\left(\frac{T_{CM,R_{MI}}(s)+T_{CM}(s)}{2} - T_e\right) \quad \text{Eq. (22d)}$$

$$C_{CB}\frac{dT_{CB}(s)}{dt(s)} = P_{col,B}(s) - \frac{T_{CB}(s)-T_e}{R_B}. \quad \text{Eq. (22e)}$$

The equations above are converted to the spatial domain by multiplying by the lethargy $\frac{dt(s)}{ds}$. The thermal capacities are denoted by C_c and the thermal resistances of the powertrain components by R_c, cf. Figure 2. Internal power losses are given by $P_{l,c}(s)$ and the heat exchange rates by $P_{col,c}(s)$. To describe the transfer of energy to the environment, its temperature T_e is important.

Constraints The initial values of the presented DAE system are implemented as equality constraints

$$\boldsymbol{x}(s=\tilde{s}) = \boldsymbol{x}_0. \quad \text{Eq. (23)}$$

Here $\tilde{s}$ represents the current position of the vehicle between 0 and s_f.

The state constraint on the vehicle velocity

$$\upsilon_{min} \le \upsilon(s) \le \upsilon_{max} \quad \text{Eq. (24)}$$

prevents backward movement through a small, positive numeric constant υ_{min} and includes a limited maximum velocity υ_{max}.

Furthermore the component and cooling fluid temperatures need to be constrained according to their respective operating ranges,

$$T_{c,min} \le T_c(s) \le T_{c,max}. \quad \text{Eq. (25)}$$

The SOC is limited by

$$0 \le \sigma(s) \le 1. \quad \text{Eq. (26)}$$

The input box constraints,

$$0 \le F_d(s) \le F_{d,max}, \quad \text{Eq. (27a)}$$

$$F_{b,max} \le F_b(s) \le 0, \quad \text{Eq. (27b)}$$

are necessary to limit the driving and braking force according to the vehicle's actuator limits. The bilinear power expression $P(s)$ is limited by the maximum engine power P_{max},

$$0 \le P(s) = F_d(s)\upsilon(s) \le P_{max}. \quad \text{Eq. (28)}$$

With the help of the complementarity constraint

$$F_d(s)F_b(s) = 0, \quad \text{Eq. (29)}$$

simultaneous activation of the brakes while accelerating is avoided. We only use this constraint to calculate a reference velocity profile $\upsilon_{ref}(s)$, which would be time-optimal if no thermodynamic constraints became active. When calculating an optimal ES instead, the driving force $F_d(s)$ is inherently limited by the thermodynamic constraints Equation 25 and the maximum available amount of energy Equation 26, which become active when the solver converges to an optimal solution $\boldsymbol{x}^*(s)$. Thus they make the explicit formulation of Equation 29 redundant, as unnecessary power losses are omitted by nature. At the same time, the computation speed of the ES increases enormously.

The vehicle dynamics are limited by

$$\left\| \left(\frac{F_d(s)+F_b(s)}{\mu F_N(s)}, \frac{F_y(s)}{\mu F_N(s)} \right) \right\|_1 \le 1 + s_u(s) \in \mathbb{R}^{n_s \times 1}, \quad \text{Eq. (30)}$$

where the normal force

$$F_N(s) = F_{N,s} + F_{N,d}(s) \quad \text{Eq. (31)}$$

consists of a static and a dynamic contribution, which can be expressed as

$$F_{N,s} = m_v g, \quad \text{Eq. (32a)}$$

$$F_{N,d}(s) = \frac{1}{4}\left(c_{l,f} + c_{l,r}\right)\upsilon^2(s). \quad \text{Eq. (32b)}$$

Here $c_{l,f}$ and $c_{l,r}$ denote the lift coefficients at the front and the rear axle. The lateral force is

$$F_y(s) = m_v \kappa(s) \upsilon^2(s) \quad \text{Eq. (33)}$$

with $\kappa(s)$ describing the raceline curvature and μ the road friction parameter. In Equation 30, we use the vector $\boldsymbol{s}_u(s)$ containing the slack variables, as we rewrite the l_1-norm to the following set of four constraints [39]

$$\pm\frac{F_d(s)+F_b(s)}{\mu F_N(s)} \pm \frac{F_y(s)}{\mu F_N(s)} \le 1 + s_{u,i}(s), i \in \{1,..,4\}. \quad \text{Eq. (34)}$$

By softening constraint Equation 30, using slack variables $\boldsymbol{s}_u(s)$, the optimization problem stays feasible if noisy and filtered velocity measurement values υ_0 are inserted as the initial condition in Equation 23.

Objective Function The objective of the ES optimization algorithm is given by Equation 1a. The matrices, which weigh

the entries in the Lagrange term l (Equation 2a) and Mayer term M (Equation 2b), are

$$W = \mathrm{diag}(Q,R,P), \quad W_K = Q, \qquad \text{Eq. (35)}$$

where

$$Q = \mathrm{diag}\left(\left[0\,0\,1e^{-2}\,1e^{-5}\,1e^{-5}\,1e^{-5}\,1e^{-5}\,1e^{-5}\right]\right) \in \mathbb{R}^{n_x \times n_x}, \qquad \text{Eq. (36a)}$$

$$R = \mathrm{diag}\left(\left[1e^{-2}\,1e^{-3}\right]\right) \in \mathbb{R}^{n_u \times n_u}, \qquad \text{Eq. (36b)}$$

$$P = 1 \in \mathbb{R}^{n_z \times n_z}. \qquad \text{Eq. (36c)}$$

The residuals are calculated using the reference values

$$\boldsymbol{x}_{k,\mathrm{r}} = \boldsymbol{x}_{K,\mathrm{r}} = \\ = \left[0\,0\,\sigma_\mathrm{r}\;T_{\mathrm{B},0}\;T_{\mathrm{M},0}\;T_{\mathrm{I},0}\;T_{\mathrm{CM},0}\;T_{\mathrm{CB},0}\right]^T, \qquad \text{Eq. (37a)}$$

$$\boldsymbol{u}_{k,\mathrm{r}} = \left[F_{\mathrm{d,r}}\;F_{\mathrm{b,r}}\right]^T, \qquad \text{Eq. (37b)}$$

$$z_{k,\mathrm{r}} = 0. \qquad \text{Eq. (37c)}$$

The weight on the algebraic variable $z(s) = \tau(s)$ and its reference value of 0 s m^{-1} in Equation 37c were chosen to enforce a time-optimal solution: The additional weights on the battery SOC, the temperature, and the control reference tracking errors act as a regularization contribution and do not have a significant influence on time-optimality [3]. The reference values were chosen to be the environmental conditions for the temperatures $T_{\mathrm{c},0}$ and to be constants, which equal about half the maximum of the actuator limits in Equations 27a and 27b, for $F_{\mathrm{d,r}}$ and $F_{\mathrm{b,r}}$. The parameter σ_r is set to a small numeric constant value, which expresses 0% SOC, when calculating a reference speed profile v_{ref}, and to 100% when optimizing an ES strategy.

The penalties in the objective on the slack variables $\boldsymbol{s}_\mathrm{u}(s)$ in Equation 30 are

$$\boldsymbol{Z}_\mathrm{u} = 10 \cdot \boldsymbol{I}_\mathrm{n} \in \mathbb{R}^{n_s \times n_s}, \qquad \text{Eq. (38a)}$$

$$\boldsymbol{z}_\mathrm{u} = 1000 \cdot \boldsymbol{I}_\mathrm{e} \in \mathbb{R}^{n_s \times 1}, \qquad \text{Eq. (38b)}$$

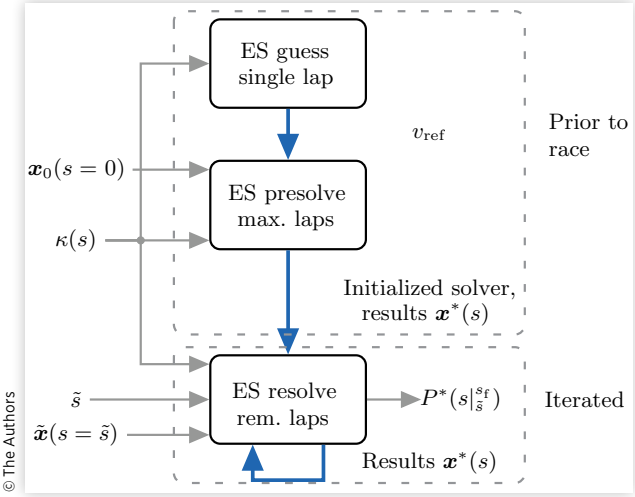
where $\boldsymbol{I}_\mathrm{n}$ denotes the identity matrix and $\boldsymbol{I}_\mathrm{e}$ an all-ones vector. No slack variables on the final coordinate $\boldsymbol{s}_\mathrm{u}(S)$ are present.

Real-Time-Capable Concept

To meet the real-time requirements in the ES module, we divide the optimization of the presented OCP into three steps; see Figure 8.

ES Guess An initial guess is crucial for an SQP optimization method. Therefore, we calculate an initial guess for the velocity profile $v_{\mathrm{ref}}(s)$, which we call the "driving dynamics reference": We solve an MLTP for a fixed path $\kappa(s)$ by solving the nonlinear OCP in Section Optimal Control Problem for two race laps. By doing so, the second lap is a flying one, which has continuous initial and final velocity values. After solving the MLTP, we can repeat the reference velocity profile $v_{\mathrm{ref}}(s)$ for the provided number of race laps and, afterward, start the ES solver therewith to optimize an MRTP. Optimizing the ES for two race laps is possible by including constraint Equation 29 in the optimization problem to form a unique solution: Since no thermodynamic constraints will become active during the two laps, an infinite number of possible combinations of the control inputs $\boldsymbol{u}$ would otherwise exist.

FIGURE 8 ES concept consisting of the stages "Guess," "Presolve," and iterative "Resolve."



ES Presolve In this step, the ES module is initialized, i.e., an optimal ES for the entire race horizon till the final coordinate s_f is calculated. Here the initial battery SOC and the initial component temperatures in $\boldsymbol{x}(s = 0)$ are set in accordance with the environment temperature T_e. To accelerate the convergence in this step, we provide an initial guess for the optimal velocity profile, which equals half the values of the $v_{\mathrm{ref}}(s)$ profile. Thereby the initial guess takes into account that the optimal velocity profile will definitely have smaller values than $v_{\mathrm{ref}}(s)$. Otherwise, no ES would be necessary. The reference velocity profile is used as an initial guess since it fulfills the modeled driving dynamic constraints and could only violate the thermodynamic and energetic ones. In combination with the calculation of the problem's exact Hessian in acados [3], no convergence issues have been detected in extensive simulations on multiple racetracks (Berlin [Germany], Hong Kong [China], Modena [Italy], Monteblanco [Spain], Paris [France], Upper Heyford [UK]).

The purpose of this step is to ensure the calculation of a valid ES and to initialize the numerical solver prior to the race start. Therefore, it is sufficient in this step to solve the OCP presented in Section Optimal Control Problem offline.

ES Resolve During the race, the ES is permanently recalculated with current measurement input from the powertrain: Based on the remaining race distance from the current

position $\tilde{s}$ to the race finish s_f, the ES solver is recompiled to account for the decreasing optimization horizon length. Afterwards the ES can be updated considering the measured states $\tilde{\boldsymbol{x}}(\tilde{s})$. In every iteration, the last optimal solution $\boldsymbol{x}^*(s)$ of the ES is used as a first guess to reduce computation times. The optimal power usage of the vehicle $P^*\left(s\big|_{\tilde{s}}^{s_f}\right)$ for the remaining race distance between $\tilde{s}$ and s_f is then forwarded to our trajectory planner, which considers this input in the velocity planning phase [39], cf. Figure 7.

Results

The experiments in this section have been conducted on a laptop running Ubuntu 20.04 equipped with an Intel i7-7820HQ CPU. The racetrack discretization was variably spaced with step size lengths ranging from 2 m in the corners to 14 m on the straights. By this, dynamic stages with high gradients are resolved accurately while the computation time can be kept low. We implemented the problem using the SQP method in acados [3], where the Quadratic Programming (QP) problems were solved by HPIPM [2] internally. For this, the implicit Runge-Kutta method of the fourth order was chosen as an integration scheme. The discretization technique was multiple shooting. All the experiments were conducted several times to account for operating-system-related varying computation times. They turned out to be constant without significant variations.

MRTP

We show an optimized MRTP for 12 laps on the Monteblanco (Spain) racetrack. The solver took approximately 10.5 s for the initial calculation of the ES, cf. Section ES Presolve, depicted in Figure 9. A total number of around 54,000 optimization variables from $\boldsymbol{x}(s)$, $\boldsymbol{u}(s)$, $\boldsymbol{s}_u(s)$, and $z(s)$ on 300 discretization points per lap arises. The number of states, control, and nonlinear constraints (see Section Constraints) sums up to approximately 50,000 in the NLP.

In Figure 9 it can be seen that the optimized velocity profile $v(s)$ and the corresponding power $P_W(s)$ are almost at a maximum at the beginning of the race [compared to the unconstrained velocity profile $v_{ref}(s)$] when the powertrain components are cold. Because of this, the powertrain is heated up as fast as possible to reach the electric motor temperature limit $T_{M,max}$ after approximately 16 km of traveled distance. After reaching this temperature limit, the velocity profile keeps $v(s)$ in a lap-wise repeating pattern in order to not exceed $T_{M,max}$; see magnification in the sixth diagram in Figure 9. This behavior can be considered time-optimal for the following argumentation: In the shown scenario, consisting of 12 race laps, the initial battery SOC $\sigma(s)$ of 50% provides a sufficient amount of energy, which is less crucial to complete the race than exceeding the motor temperature limit. It is therefore beneficial to increase the motor temperature $T_M(s)$ to the highest possible absolute value to maximize the motor's cooling power $P_{col,M}(s)$, cf. Equation 22a. By doing this, the acceptable motor power losses $P_{l,M}(s)$ are maximized, which ultimately leads to a minimization of the race time $t(s_f)$, i.e., the summed lethargy $z(s)$.

The constant velocity pattern in the second half of the race is not only beneficial for keeping the motor temperature $T_M(s)$ within its boundary but also for keeping the absolute power losses $P_l(s) = P_{i,B}(s) - P_W(s)$ in the powertrain to a minimum: Increased driving velocity or driving power leads to overproportional losses $P_l(s)$, cf. Equations 6, 9b, and 10a. In contrast, decreasing the vehicle velocity $v(s)$ reduces power losses $P_l(s)$ only proportionally less. Additionally, a linear increase of the average velocity for a race lap in combination with a linear decrease of the same magnitude of the average driving velocity on a subsequent lap leads to a diminished average race speed over time for both laps. Due to this, a uniform driving pattern supports both the minimization of power losses $P_l(s)$ and race time $t(s_f)$.

In Figure 9, we also depicted battery $P_{l,B}(s)$, motor $P_{l,M}(s)$, and inverter losses $P_{l,I}(s)$. Even if the driven average velocity per lap decreases slightly between 0 km and 8 km, motor losses $P_{l,M}(s)$ increase. Here the rising motor temperature $T_M(s)$ leads to the increasing copper losses $P_{l,cop}(s)$, cf. Equation 10a. Stator iron and rotor losses $P_{l,sir}(s)$ vary with the rotational frequency $\omega_M(s)$ as modeled in Equation 10b. Similarly, battery losses $P_{l,B}$ decrease with the dropping internal resistance $R_i(s)$, but increase in total, due to the decreasing voltage $U_{oc}(s)$ and therefore increasing current $I_B(s)$. The lower velocity $v(s)$ at the end of the race (16 km to 28 km) leads to a falling battery temperature $T_B(s)$, which, in turn, raises the internal resistance $R_i(s)$ again. The inverter's peak switching losses $P_{l,I,swi}(s)$ decrease slightly by approximately 0.2 kW from $s = 0$ km to 12 km due to the decreasing DC voltage $U_{dc}(s)$, cf. Equation 9a. The effect on the peak conduction losses $P_{con}(s)$ over the same race distance due to less requested motor torque $M(s)$, and therefore $I_e(s)$, is minor at 0.1 kW, cf. Equation 9b.

The optimal vehicle behavior over the last few meters of the shown scenario is also interesting. Here the imminent crossing of the finish line lets the algorithm break with the previous velocity pattern and, instead, maximize the requested vehicle power $P(s)$. By doing this, the battery SOC $\sigma(s)$ decreases to 0% of the remaining energy and the motor temperature $T_M(s)$ simultaneously reaches $T_{M,max}$ at the end of the race.

Real Time Capability

To demonstrate the real time capability of the presented ES concept, we show its calculation times in Figure 10.

For the Monteblanco (Spain) and the Modena (Italy) racetrack, we show the calculation times Δt_{cal} achieved to initialize (ES Presolve) and to subsequently recalculate (ES Resolve) the ES, cf. Section Real-Time-Capable Concept. To re-optimize,

FIGURE 9 MRTP solved for 12 laps on the Monteblanco (Spain) racetrack. Shown are optimal and reference velocity $v(s)$ and $v_{ref}(s)$, corresponding power $P(s)$ including total losses $P_l(s)$, detailed battery power losses $P_{l,B}(s)$, and motor losses $P_{l,M}(s)$ split into copper, and stator iron and rotor losses. Additionally, the battery and motor temperatures $T_{B/M}(s)$ are indicated together with the internal battery resistance $R_i(T_B(s))$, the DC voltage $U_{dc}(\sigma(s))$, and the SOC $\sigma(s)$.

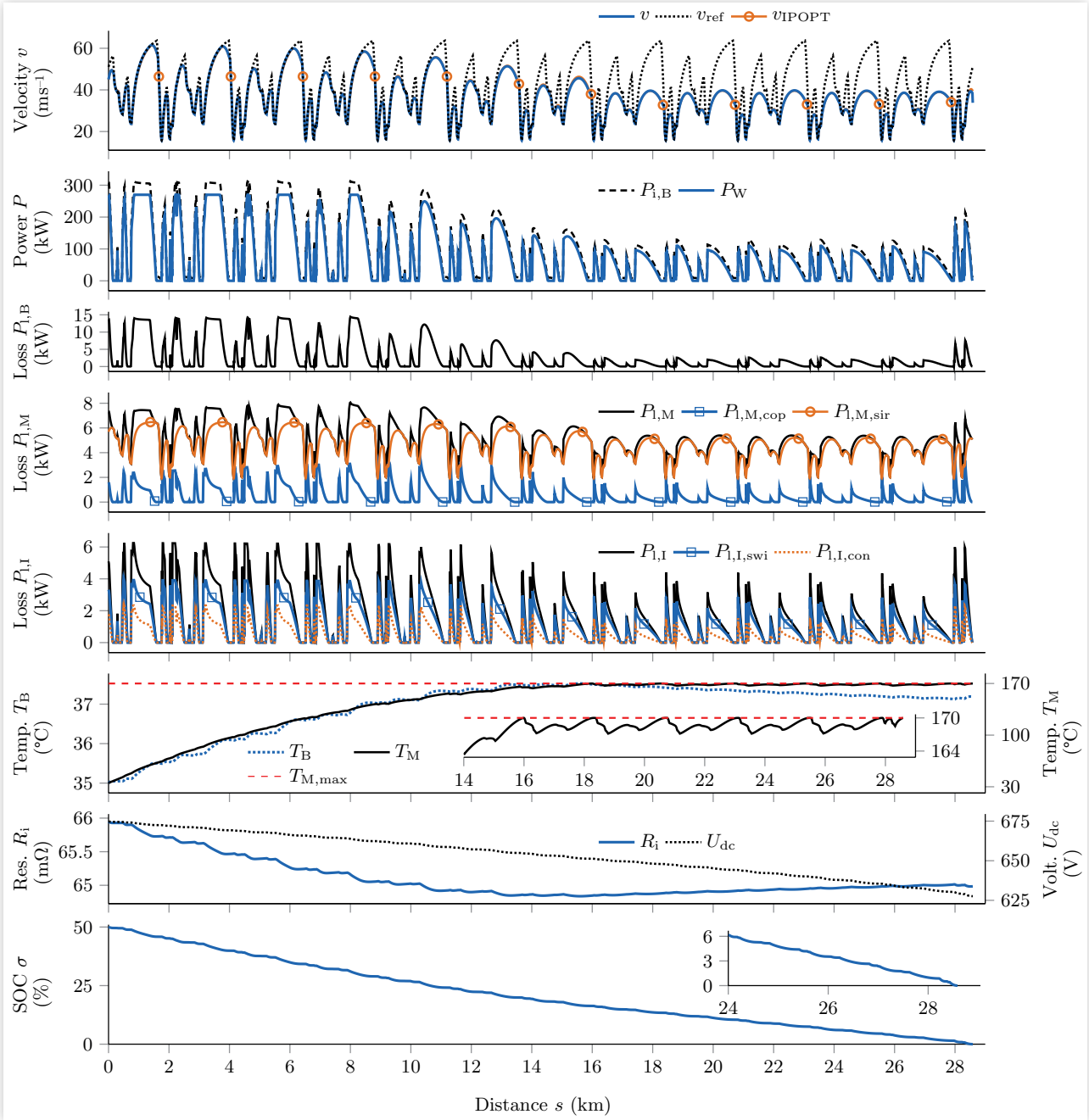
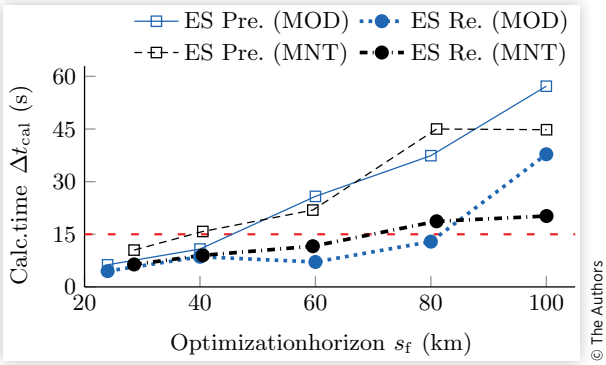


FIGURE 10 Computation times of the ES on the Monteblanco (MNT, Spain) and Modena (MOD, Italy) racetracks. The real time limit is marked in red.



we insert a simulated measurement value at $\tilde{s}$, with the difference that the sensor resolution (0.5°C) is added to the initially calculated battery temperature $T_B(s)$. The discretization point $\tilde{s}$ was chosen randomly after the first meters of the race start to showcase the ES recalculation on a long remaining optimization horizon till s_f. By doing this, the solver has to recalculate the ES and converge from an infeasible starting point $\boldsymbol{x}_0$ as the ODEs in the environment of the discretization point $\tilde{s}$ are not fulfilled anymore. The calculation times Δt_{cal}, given for the ES Resolve-step, comprise the solver recompilation and its runtime to account for the entire computation procedure in the online scenario. The starting conditions for the ES initialization are 35°C and 100% of the SOC.

On the Modena track, we obtain the following results: In the 24 km and 40 km run, the motor temperature $T_{M,max}$ is permanently exploited. Here the recalculation times are almost equal to the ones of the ES initialization phase. With longer optimization horizons (60 km, 80 km, and 100 km), the constraining variable shifts from the motor temperature $T_M(s)$ to the battery SOC $\sigma(s)$, which limits the velocity $v(s)$ in the optimal solution. In these scenarios, recalculation times are smaller compared to the ES initialization.

On the Monteblanco track, the results are similar qualitatively: The recalculation times are smaller than for the initialization. In the experiments comprising optimization horizons of 29 km, 40 km, and 60 km, the motor temperature $T_M(s)$ permanently represents an active constraint and limits full-speed operation. On the 100 km run, the battery SOC $\sigma(s)$ is the crucial variable. The calculation times Δt_{cal} for both ES concept steps with a horizon of 80 km are interesting: In this case, both SOC $\sigma(s)$ and motor temperature $T_M(s)$, reach their corresponding lower and upper boundaries, respectively, when crossing the finish line. Because of this, the computation time Δt_{cal} for the ES initialization resulted in a slightly higher value compared to Δt_{cal} for $s_f = 100$ km.

From Figure 10 we can conclude that for optimization horizons up to 70 km, corresponding to 35 and 29 race laps on the Modena and the Monteblanco racetracks, respectively, the recalculation times Δt_{cal} remained below 15 s. Compared to the lap times achieved during maximum speed operation leveraging $v_{ref}(s)$, the ES can be recalculated several times during a single race lap. Therefore we consider the online update rate to be sufficient for a competition run.

Validation and Benchmark

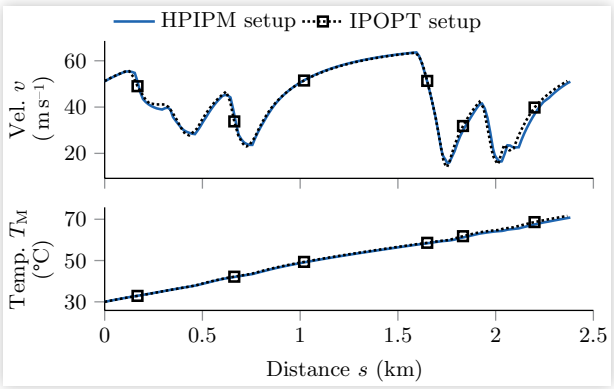
In the following, we will validate the numerical optimality of the proposed ES algorithm and justify the solver selection.

Optimality To prove the optimality of the proposed ES algorithm (Section Energy Strategy), we compare our results to the algorithms presented in [8, 14] for the Monteblanco (Spain) race circuit: The authors discretize an MLTP using direct orthogonal collocation and solve the resulting NLP with the numerical solver IPOPT [4]. The MLTP algorithm in [8] has been proven to produce accurate results in real-world autonomous racing applications.

When comparing the results produced by our ES algorithm for one race lap, i.e., when solving an MLTP for a fixed path (ES Guess in Figure 8), we achieve the results shown in Figure 11. The NRMSE for the optimized velocity profile $v(s)$ is 3.11% in comparison to the solution produced by the IPOPT setup. The error mainly stems from different velocity extrema, caused by the model descriptions: In the ES algorithm, we use a point mass model (Section Optimal Control Problem), whereas the MLTP formulation in [8] comprises a Nonlinear Double Track Model (NDTM) including a detailed tire behavior description.

Due to the good correlation in the optimal velocities $v(s)$, the temperatures T_c calculated by HPIPM and IPOPT also match accurately; see Figure 11 (bottom). The difference in the motor temperature $T_M(s)$ at the end of the race lap is 0.78 °C, which correlates to a normalized error for the entire temperature increase of 1.19%. This behavior is feasible as the IPOPT velocity profile is slightly more aggressive regarding accelerations and the velocity maxima, as the selected discretization points by both algorithms do not match exactly. In turn, exponentially higher power losses $P_l(s)$ are produced by the scenario solved by the IPOPT setup from [8, 14].

FIGURE 11 Solution of the IPOPT setup [8, 14] in comparison with the ES problem formulation solved by HPIPM.



Solver Comparison In this paragraph, we justify the choice of the HPIPM solver. Therefore we compare the MRTP's optimal velocity profile $v(s)$, calculated in the ES Initialization step for 12 race laps from Section MRTP, to a solution $v_{\mathrm{IPOPT}}(s)$ calculated by the solver IPOPT for the same OCP, cf. Section Optimal Control Problem. IPOPT is a well-known solver for offline applications, which is often used for benchmarking solution quality [3, 39]. Note that the IPOPT solver is also initialized with the reference velocity profile $v_{\mathrm{ref}}(s)$.

Both velocity profiles—$v(s)$ and $v_{\mathrm{IPOPT}}(s)$—match closely with a small numerical error of 0.95%; see Figure 9 (first plot). The calculation times differ by the absolute values of 10.5 s to calculate $v(s)$ and 2.4 min to optimize $v_{\mathrm{IPOPT}}(s)$. In this scenario, HPIPM is approximately 13.5 times faster, and the solution quality is nearly the same.

Summary of the Results

In the following, we wish to summarize the results in this publication. We presented

- And solved an MRTP comprising 12 subsequent race laps on the Monteblanco (Spain) racetrack in an initialization calculation time of approximately 10.5 s. When the vehicle follows this strategy, it resolves the conflict of achieving a minimum race time while simultaneously adhering to its technical constraints.
- An analysis of the computation times of our ES algorithm. Race distances of approximately 70 km to 80 km can be re-optimized in less than our target calculation time of 15 s.
- A qualitative argumentation explaining the plausibility of the ES results and a quantitative comparison to a state-of-the-art MLTP optimization. The benchmark showed an error of 3.11% for the velocity and 1.19% for the machine temperature profile.
- A justification of the selected numerical solver (HPIPM). We compared its solving time to IPOPT, where we identified a speedup of the factor 13.5 and a suboptimality of 0.95%.

Conclusion

To achieve maximum performance in an all-electric vehicle race, we presented an ES which solves an MRTP in real time. It takes into account the thermodynamic limitations of the powertrain components and can, therefore, avoid safety shutdowns during a race.

Furthermore, the limited amount of stored battery energy is utilized as efficiently as possible. By this, a race can be finished without exceeding technical limitations from the powertrain in a minimum time. The necessary powertrain models for the battery, electric motors, and inverters are based on physical descriptions, which are mainly parameterized using their datasheet values and a data-driven approximation for the stator iron losses. We also compared the powertrain models to real-world measurement data.

Optimization horizons with a length of up to 100 km can be solved in less than a minute, while simultaneously discretizing the racetrack finely enough to accurately describe the fast-changing driving dynamics. We validated the output of our ES algorithm by a comparison with the results obtained with a state-of-the-art MLTP planner, where we achieved good coherence. We also justified the solver selection, HPIPM, for solution quality and computational speed.

In future research, we wish to entirely validate the results of the proposed ES algorithm, when applied to an electric race car, which follows the optimized ES strategy. Further, we will investigate the mathematical structure of the resulting optimization problem in greater detail to speed up its computation. The source code of the ES algorithm is available on an open-source basis on our institute's GitHub page [42].

Contributions and Acknowledgment

T. H. initiated the idea of the article and contributed significantly to the concept, modeling, and results. F. S. and M. B. contributed to the powertrain modeling and the OCP in their master's theses, J. B. to the whole concept of the article. M. L. provided a significant contribution to the concept of the research project. He revised the article critically for important intellectual content. M. L. gave final approval for the publication of this version and is in agreement with all aspects of the work. As a guarantor, he accepts responsibility for the overall integrity of this article.

We thank our colleagues Alexander Koch, Francesco Passigato, Andreas Schimpe, Alexander Wischnewski, and Frank Diermeyer for the valuable discussions about the content of this article. We would like to thank the Roborace team for giving us the opportunity to work with them and for the use of their vehicles for our research project. We would also like to thank the Bavarian Research Foundation (Bayerische Forschungsstiftung) for funding us in connection with the "rAIcing" research project. This work was also conducted using the basic research fund of the Institute of Automotive Technology at the Technical University of Munich.

Contact Information

Thomas Herrmann
thomas.herrmann@tum.de

Acronyms

ANN - Artificial Neural Network
BLDC - Brushless Direct Current
DAE - Differential-Algebraic Equation
ECMS - Equivalent Consumption Minimization Strategy
ES - Energy Strategy

HPIPM - High-Performance Interior-Point Method
ICE - Internal Combustion Engine
IGBT - Insulated-Gate Bipolar Transistor
IPOPT - Interior Point OPTimizer
LP - Linear Programming
MCTS - Monte Carlo Tree Search
MLTP - Minimum Lap Time Problem
MPC - Model Predictive Control
MRTP - Minimum Race Time Problem
NDTM - Nonlinear Double Track Model
NLP - Nonlinear Programming
NRMSE - Normalized Root Mean Square Error
OCP - Optimal Control Problem
OCV - Open-Circuit Voltage
ODE - Ordinary Differential Equation
PMP - Pontryagin's Minimum Principle
QP - Quadratic Programming
SOC - State of Charge
SOCP - Second-Order Conic Programming
SQP - Sequential Quadratic Programming

References

1. Roborace, "About Roborace," https://roborace.com/, accessed January 8, 2022.
2. Frison, G. and Diehl, M., "HPIPM: A High-Performance Quadratic Programming Framework for Model Predictive Control," *IFAC-PapersOnLine* 53, no. 2 (2020): 6563-6569.
3. Verschueren, R., Frison, G., Kouzoupis, D., van Duijkeren, N. et al., "Acados: A Modular Open-Source Framework for Fast Embedded Optimal Control," arXiv preprint arXiv:1910.13753, 2019.
4. Wächter, A. and Biegler, L.T., "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming* 106, no. 1 (2006): 25-57, https://doi.org/10.1007/s10107-004-0559-y.
5. Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B. et al., "CasADi: A Software Framework for Nonlinear Optimization and Optimal Control," *Mathematical Programming Computation* 11, no. 1 (2019): 1-36, https://doi.org/10.1007/s12532-018-0139-4.
6. Limebeer, D.J.N. and Perantoni, G., "Optimal Control of a Formula One Car on a Three-Dimensional Track—Part 2: Optimal Control," *Journal of Dynamic Systems, Measurement, and Control* 137, no. 5 (2015): 051019, https://doi.org/10.1115/1.4029466.
7. Tremlett, A.J. and Limebeer, D.J.N., "Optimal Tyre Usage for a Formula One Car," *Vehicle System Dynamics* 54, no. 10 (2016): 1448-1473, https://doi.org/10.1080/00423114.2016.1213861.
8. Christ, F., Wischnewski, A., Heilmeier, A., and Lohmann, B., "Time-Optimal Trajectory Planning for a Race Car Considering Variable Tyre-Road Friction Coefficients," *Vehicle System Dynamics* 3, no. 1 (2019): 1-25, https://doi.org/10.1080/00423114.2019.1704804.
9. Dal Bianco, N., Lot, R., and Gadola, M., "Minimum Time Optimal Control Simulation of a GP2 Race Car," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 232, no. 9 (2018): 1180-1195, https://doi.org/10.1177/0954407017728158.
10. Liu, X. and Fotouhi, A., "Formula-E Race Strategy Development Using Artificial Neural Networks and Monte Carlo Tree Search," *Neural Computing and Applications* 60, no. 4 (2020): 1516, https://doi.org/10.1007/s00521-020-04871-1.
11. Liu, X., Fotouhi, A., and Auger, D.J., "Optimal Energy Management for Formula-E Cars with Regulatory Limits and Thermal Constraints," *Applied Energy* 279 (2020): 115805, https://doi.org/10.1016/j.apenergy.2020.115805.
12. Patterson, M.A. and Rao, A.V., "GPOPS-II," *ACM Transactions on Mathematical Software* 41, no. 1 (2014): 1-37, https://doi.org/10.1145/2558904.
13. Borsboom, O., Fahdzyana, C.A., Salazar, M., and Hofman, T., "Time-Optimal Control Strategies for Electric Race Cars with Different Transmission Technologies," in *2020 IEEE Vehicle Power and Propulsion Conference (VPPC)*, Gijon, Spain, 2020, IEEE.
14. Herrmann, T., Passigato, F., Betz, J., and Lienkamp, M., "Minimum Race-Time Control-Strategy for an Autonomous Electric Racecar," in *2020 IEEE Intelligent Transportation Systems Conference (ITSC)*, Rhodes, Greece, 2020, IEEE, https://doi.org/10.1109/ITSC45102.2020.9294681.
15. Locatello, A., Konda, M., Borsboom, O., Hofmann, T. et al., "Time-Optimal Control of Electric Race Cars under Thermal Constraints," in *2021 European Control Conference (ECC)*, Rotterdam, the Netherlands, 2021, IEEE.
16. Salazar, M., Balerna, C., Chisari, E., Bussi, C. et al., "Equivalent Lap Time Minimization Strategies for a Hybrid Electric Race Car," in *2018 IEEE Conference on Decision and Control (CDC)*, Miami, FL, 2018, 6125-6131, IEEE, https://doi.org/10.1109/CDC.2018.8618724.
17. Salazar, M., Elbert, P., Ebbesen, S., Bussi, C. et al., "Time-Optimal Control Policy for a Hybrid Electric Race Car," *IEEE Transactions on Control Systems Technology* 25, no. 6 (2017): 1921-1934, https://doi.org/10.1109/TCST.2016.2642830.
18. Salazar, M., Balerna, C., Elbert, P., Grando, F.P. et al., "Real-Time Control Algorithms for a Hybrid Electric Race Car Using a Two-Level Model Predictive Control Scheme," *IEEE Transactions on Vehicular Technology* 66, no. 12 (2017): 10911-10922, https://doi.org/10.1109/TVT.2017.2729623.
19. Domahidi, A., Chu, E., and Boyd, S., "ECOS: An SOCP Solver for Embedded Systems," in *2013 European Control Conference (ECC)*, Zurich, Switzerland, 2013, 3071-3076, IEEE, https://doi.org/10.23919/ECC.2013.6669541.
20. Zanelli, A., Domahidi, A., Jerez, J., and Morari, M., "FORCES NLP: An Efficient Implementation of Interior-Point Methods for Multistage Nonlinear Nonconvex

Programs," *International Journal of Control* 93, no. 1 (2020): 13-29, https://doi.org/10.1080/00207179.2017.1316017.

21. Atmaca, E., "Energy Management of Solar Car in Circuit Race," *Turkish Journal of Electrical Engineering & Computer Sciences* 23 (2015): 1142-1158, https://doi.org/10.3906/elk-1212-37.
22. Guerrero Merino, E. and Duarte-Mermoud, M.A., "Online Energy Management for a Solar Car Using Pseudospectral Methods for Optimal Control," *Optimal Control Applications and Methods* 37, no. 3 (2016): 537-555, https://doi.org/10.1002/oca.2210.
23. Cifuentes, D. and Pradenas, L., "Optimization of the Velocity Profile of a Solar Car Used in the Atacama Desert," in *Hybrid Metaheuristics*, Blesa Aguilera, M.J., Blum, C., Gambini Santos, H., Pinacho-Davidson, P., and Godoy del Campo, J. (Eds). Vol. 11299 of Lecture Notes in Computer Science (Cham: Springer International Publishing, 2019), 164-171, https://doi.org/10.1007/978-3-030-05983-5_12.
24. Rao, A.V., Benson, D.A., Darby, C., Patterson, M.A. et al., "Algorithm 902," *ACM Transactions on Mathematical Software* 37, no. 2 (2010): 1-39, https://doi.org/10.1145/1731022.1731032.
25. Boggs, P.T. and Tolle, J.W., "Sequential Quadratic Programming," *Acta Numerica* 4 (1995): 1-51, https://doi.org/10.1017/S0962492900002518.
26. Kerrigan, E.C., "Soft Constraints and Exact Penalty Functions in Model Predictive Control," in *Proceedings of the United Kingdom Automatic Control Council 2000*, Cambridge, UK, IFAC (Eds), 2000.
27. Wildfeuer, L., Wassiliadis, N., Reiter, C., Baumann, M. et al., "Experimental Characterization of Li-Ion Battery Resistance at the Cell, Module and Pack Level," in *2019 Fourteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, 2019, 1-12, IEEE, https://doi.org/10.1109/EVER.2019.8813578.
28. Sciarretta, A. and Vahidi, A., *Energy-Efficient Driving of Road Vehicles: Toward Cooperative, Connected, and Automated Mobility. Lecture Notes in Intelligent Transportation and Infrastructure*, 1st ed. (Cham: Springer International Publishing and Springer, 2020), https://doi.org/10.1007/978-3-030-24127-8.
29. Semikron, A.G., "Datasheet SKiM459GD12E4," 2016, https://www.semikron.com/products/product-classes/igbt-modules/detail/skim459gd12e4-23930010.html, accessed January 8, 2022.
30. Mazgaj, W., Rozegnał, B., and Szular, Z., "Switching Losses in Three-Phase Voltage Source Inverters," *Technical Transactions* , no. 2-E (13) (2015 (112)): 47-60, https://doi.org/10.4467/2353737XCT.15.087.3919, accessed January 8, 2022.
31. Graovac, D., Puerschel, M., and Kiep, A., "Application Note: MOSFET Power Losses Calculation Using the Data-Sheet Parameters," 2006, https://application-notes.digchip.com/070/70-41484.pdf, accessed January 8, 2022.
32. Xie, W., Wang, X., Wang, F., Xu, W. et al., "Dynamic Loss Minimization of Finite Control Set-Model Predictive Torque Control for Electric Drive System," *IEEE Transactions on Power Electronics* 31, no. 1 (2016): 849-860, https://doi.org/10.1109/TPEL.2015.2410427.
33. Blaabjerg, F., Jaeger, U., and Munk-Nielsen, S., "Power Losses in PWM-VSI Inverter Using NPT or PT IGBT Devices," *IEEE Transactions on Power Electronics* 10, no. 3 (1995): 358-367, https://doi.org/10.1109/63.388002.
34. International Rectifier, "Application Note: Application Characterization of IGBTs," https://www.chtechnology.com/media/documentlibrary/documents/a/n/an-990_application_characterization_of_igbt_s.pdf, accessed January 8, 2022.
35. Rollbuehler, C., Peukert, S., Fritz, D., Heyd, J.-F. et al., "Investigations on the Experimental Identification of AC-Copper Losses in Permanent Magnet Synchronous Machines Using a Motor Sub-Assembly," in *IECON 2019—45th Annual Conference of the IEEE Industrial Electronics Society*, Lisbon, Portugal, 2019, 1150-1156, IEEE, https://doi.org/10.1109/IECON.2019.8926835.
36. Melkebeek, J.A., *Electrical Machines and Drives* (Cham: Springer International Publishing, 2018), https://doi.org/10.1007/978-3-319-72730-1.
37. Chiodetto, N., Bianchi, N., and Alberti, L., "Improved Analytical Estimation of Rotor Losses in High-Speed Surface-Mounted PM Synchronous Machines," *IEEE Transactions on Industry Applications* 53, no. 4 (2017): 3548-3556, https://doi.org/10.1109/TIA.2017.2693178.
38. Mohan, N. and Raju, S., *Analysis and Control of Electric Drives: Simulations and Laboratory Implementation* (Hoboken, NJ: John Wiley & Sons, Inc., 2021)
39. Herrmann, T., Wischnewski, A., Hermansdorfer, L., Betz, J. et al., "Real-Time Adaptive Velocity Optimization for Autonomous Electric Cars at the Limits of Handling," *IEEE Transactions on Intelligent Vehicles* 6, no. 4, (2021): 665-677, https://doi.org/10.1109/TIV.2020.3047858.
40. Heilmeier, A., Wischnewski, A., Hermansdorfer, L., Betz, J. et al., "Minimum Curvature Trajectory Planning and Control for an Autonomous Race Car," *Vehicle System Dynamics* 25, no. 8 (2019): 1-31, https://doi.org/10.1080/00423114.2019.1631455.
41. Betz, J., Wischnewski, A., Heilmeier, A., Nobis, F. et al., "A Software Architecture for the Dynamic Path Planning of an Autonomous Racecar at the Limits of Handling," in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, Graz, Austria, 2019, 1-8, IEEE, https://doi.org/10.1109/ICCVE45908.2019.8965238.
42. Institute of Automotive Technology, "embes - Embedded Energy Strategy," https://github.com/TUMFTM/embes, accessed January 8, 2022.

6 Velocity Optimization (Local, Online) – TIV 2021

Summary

In order to implement the global ES on the road, we present an optimization-based velocity planner. This real-time software module acts as the interface to an autonomous driving software stack: Together with a graph-based path planner [196], the proposed velocity optimization algorithm forms the trajectory planning module in our Roborace software stack [12, 25, 224]. The velocity module handles external information about the environment like the tire-road friction potential, and the power limitations, which are computed by the ES Core Module presented in Chapter 5 [9]. This approach allows to use the limited amount of available energy as lap-time-efficiently as possible, and to transfer the global and dynamically adapting ES to the road. The final local trajectory is forwarded to the vehicle dynamics controller [167] (Figures 3.1 and 6.1).

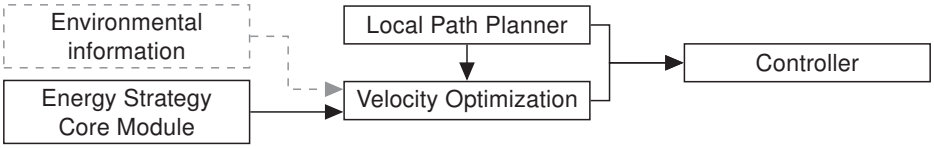


Figure 6.1: In- and outputs of the Velocity Optimization module. Environmental information is, e.g., the tire-road friction potential.

We split the literature dealing with the online replanning of trajectories into the three fields of “separated/two-step trajectory planning”, where path and velocity are calculated independently [187, 191, 193, 201], “combined trajectory planning”, where the free variables defining a trajectory are jointly optimized [24, 194, 197, 198], and “MPC approaches”, which also consider the current vehicle state and feasible terminal sets [181, 183, 200, 227, 232]. Our two-step approach allows to be applied in real-time-critical applications, while simultaneously considering the ES input, spatially variable acceleration limits [233], and the selected path as external parameters. The algorithmic combination with the graph-based path planner handles non-convex scenarios at a high update rate being sufficient for autonomous racing [196].

To solve the problem of velocity planning, we present a nonlinear model predictive planner (NMPP), mathematically formulated as a tailored multi-parametric sequential quadratic program (mpSQP) [234, p. 353]. It iteratively solves local convex approximations [227] of the original problem. The underlying point mass vehicle dynamics model delivers a small number of optimization variables. We also integrate slack variables into the problem formulation to allow the algorithm for violations of the combined acceleration constraints. These slacks are crucial to preserve recursive feasibility, since the friction estimation module and the local path planner deliver slightly numerically varying inputs, possibly leading to infeasibility while driving. We implement the objective function as a sum of the squared l^2-norms of the desired vehicle velocity and its jerk, together with the l^2- and l^1-norms of the slack variables to keep their values small

and maintain the original problem's optimum. Through this formulation, we obtain a constant Hessian matrix of the objective function with a tunable condition number, which supports the real-time capability of the algorithm.

To show the real-time performance of the developed algorithm, we run the full software stack on a HIL-simulator including the automotive-grade ARM-based vehicle ECU NVIDIA Drive PX2 [235, p. 115], [236]. On this device, the mpSQP permanently computes both, an emergency and a performance speed profile. First, we validate the most important part of the algorithm, the design of the objective function. A run including an autonomous start and stop scenario shows that the slack variables are not exploited for lap time gains, and that the jerk term contributes by smoothing numerical oscillations.

To demonstrate the ability of the proposed algorithm to transfer the global ES to the road, we conduct an experiment where the space-dependent power limits are received externally (Figure 6.2). We calculate a global static ES beforehand to isolate the interaction between local and global planner without inference from a dynamic ES replanning. By this, the mpSQP adapts the local velocities to meet the variable power constraints. A drift between the globally admissible energy expenditure and the local consumption after one race lap originates from different vehicle dynamics assumptions, which stem from a nonlinear double-track model in the global ES, and a point mass model in the velocity planning algorithm.

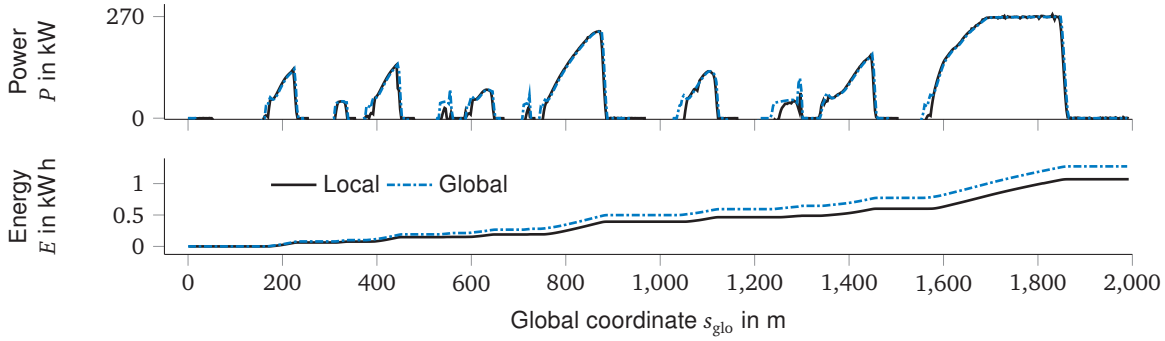


Figure 6.2: Global ES locally implemented, © 2020 IEEE.

The multi-parametric nature of the proposed SQP algorithm also allows to handle other environmental information like, e.g., the spatially variable acceleration limits (Figure 6.1). A second simulation reveals that drops of up to $50\,\%$ in the friction potential within several meters between the tires and the road can algorithmically be handled, while fully leveraging the maximum available acceleration potential, ultimately leading to the minimum achievable lap time.

A third experiment summarizes the computation time of the developed SQP algorithm. Mean runtimes between $6\,\text{ms}$ to $7\,\text{ms}$ are achieved on an Intel i7-7820HQ CPU, and $32\,\text{ms}$ to $34\,\text{ms}$ on the target hardware with an ARM A57 CPU. To justify our QP solver selection, we benchmarked three iterative solvers, where each of them implements a different mathematical solving strategy (Section 2.2). The selected solvers are the ADMM solver OSQP [156], the active set solver qpOASES [159], and the primal-dual nonlinear IP method IPOPT [152]. It turned out that the solution quality using OSQP was higher compared to qpOASES. Even the calculation time was smaller by orders of magnitude. To validate the effort we put into tailoring the proposed mpSQP, we directly solve the original nonlinear velocity optimization problem by IPOPT. To this end, we implement the optimization problem using the CasADi modeling language [149]. Here, the IPOPT solver produces velocity profiles which are slightly better in quality compared to the mpSQP solved by OSQP in an SQP method. However, the calculation time of the nonlinear

problem solved by IPOPT was approximately five times higher on average. Additionally, the peak calculation times of IPOPT exclude this solver for the usage in our real-time setup.

Contributions

T. H. initiated the idea of the paper and contributed significantly to the concept, modeling, implementation and results. A. W. contributed to the design and feasibility analysis of the optimization problem. L. H. contributed essentially to the integration of variable acceleration limits. J. B. contributed to the whole concept of the paper. M. L. provided a significant contribution to the concept of the research project. He revised the paper critically for important intellectual content. M. L. gave final approval for the publication of this version and is in agreement with all aspects of the work. As a guarantor, he accepts responsibility for the overall integrity of this paper.

Copyright notice

Real-Time Adaptive Velocity Optimization for Autonomous Electric Cars at the Limits of Handling

Thomas Herrmann, Alexander Wischnewski, Leonhard Hermansdorfer, Johannes Betz, Markus Lienkamp

***Abstract*—With the evolution of self-driving cars, autonomous racing series like Roborace and the Indy Autonomous Challenge are rapidly attracting growing attention. Researchers participating in these competitions hope to subsequently transfer their developed functionality to passenger vehicles, in order to improve self-driving technology for reasons of safety, and due to environmental and social benefits. The race track has the advantage of being a safe environment where challenging situations for the algorithms are permanently created. To achieve minimum lap times on the race track, it is important to gather and process information about external influences including, e.g., the position of other cars and the friction potential between the road and the tires. Furthermore, the predicted behavior of the ego-car's propulsion system is crucial for leveraging the available energy as efficiently as possible. In this paper, we therefore present an optimization-based velocity planner, mathematically formulated as a multi-parametric Sequential Quadratic Problem (mpSQP). This planner can handle a spatially and temporally varying friction coefficient, and transfer a race Energy Strategy (ES) to the road. It further handles the velocity-profile-generation task for performance and emergency trajectories in real time on the vehicle's Electronic Control Unit (ECU).**

***Index Terms*—Real-Time Numerical Optimization, Optimal Control, Velocity Planning, Trajectory Planning, Autonomous Electric Vehicles, Energy Strategy, Variable Friction Potential**

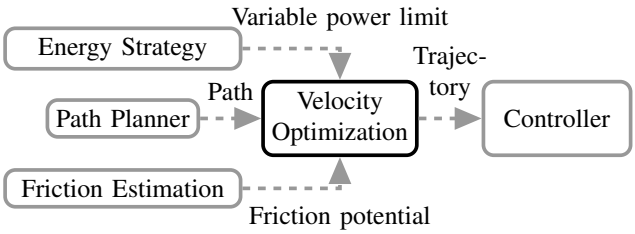


Fig. 1. Software environment of presented velocity optimization module.

Fig. 2. TUM Roborace DevBot 2.0 on the race track.

I. Introduction

THE Technical University of Munich has been participating in the Roborace competition since 2018. Many parts of our software stack are already available on an open source basis [1] including the code of the algorithm in this work [2]. This paper explains an optimization-based Nonlinear Model Predictive Planner (NLMPP), mathematically formulated as a multi-parametric Sequential Quadratic Problem (mpSQP) [3], to calculate the velocity profiles during a race. The velocity planner inputs are the offered race paths ("performance" and "emergency"), stemming from our graph-based path-planning framework [4], see Fig. 1. The presented velocity optimization in combination with the path framework span our local trajectory planner that will be used within the competition. The trajectory planner's output is forwarded to the underlying vehicle controller [5], [6], transforming the target trajectory into actuator commands for the race car, called "DevBot 2.0", see Fig. 2. A huge motivation behind setting up an optimization-based velocity planner was to be able to handle information about the locally and temporally varying friction potential on the race track [7], and utilize the information provided by the race Energy Strategy (ES) [8] as a vehicle's velocity profile has a significant influence on it's energy consumption and power losses [9]. The friction potential estimation and the calculation of the ES are handled by separate modules in our software stack. Their outputs, the friction potential and variable power limits, are then considered in the presented velocity optimization algorithm.

To achieve real-time-capable calculation times, we build local approximations of the nonlinear velocity-planning problem, resulting in convex multi-parametric Quadratic Problems (mpQP) [10] that can be solved iteratively using a Sequential Quadratic Problem (SQP) method. We evaluated different open-source Quadratic Problem (QP) solvers and compared their solution qualities and calculation times to a direct solution of the Nonlinear Problem (NLP). We chose the Operator Splitting Quadratic Problem (OSQP) [11] solver as it outperformed its competitors on a standard x86-64 platform as well as on the DevBot's automotive-grade Electronic Control Unit (ECU), the ARM-based NVIDIA Drive PX2.

T. Herrmann (corresponding author), L. Hermansdorfer, J. Betz and M. Lienkamp are with the Institute of Automotive Technology, Department of Mechanical Engineering, Technical University of Munich, Garching (Munich), 85748 Germany e-mail: thomas.herrmann@tum.de.

A. Wischnewski is with the Institute of Automatic Control, Department of Mechanical Engineering, Technical University of Munich, Garching (Munich), 85748 Germany

Manuscript received May 30, 2020; revised September 25, 2020.

A. State of the art

The field of trajectory planning of vehicles at the limits of handling is attracting growing attention in research. The scenarios where the car is required to be operated at the limits of its driving dynamics will become more and important as we see the spread of cars equipped with self-driving functionality, and even fully autonomous vehicles. Through this, complex scenarios with self-driving vehicles on the road will occur more frequently. Research is also being carried out on the race track where these challenging scenarios can deliberately be created in a safe environment [12].

In the field of global trajectory optimization for race tracks, different mathematical concepts are applied. In the work of Ebbesen et al. [13] a Second Order Conic Problem (SOCP) formulation is used to calculate the optimal power distribution within the hybrid powertrain of a Formula One race car leading to globally time-optimal velocity profiles for a given path. For the same racing format, Limebeer and Perantoni [14] took into account the 3D geometry of the race track within their formulation of an Optimal Control Problem (OCP) to solve a Minimum Lap Time Problem (MTLP). In a similar approach, Tremlett and Limebeer [15] consider the thermodynamic effects of the tires. Christ et al. [16] consider spatially variable but temporally fixed friction coefficients along the race track to calculate time-optimal global race trajectories for a sophisticated Nonlinear Double Track Model (NDTM). They show a significant influence of the variable friction coefficients on the achievable lap time when considered during the trajectory optimization process. A minimum-curvature QP formulation, calculating the quasi-time-optimal trajectory for an autonomous race car on the basis of an occupancy grid map, is given by Heilmeier et al. [5]. Their advantage in comparison to [16] is the computation time, but the resulting trajectories are suboptimal in terms of lap time. Also, Dal Bianco et al. [17] formulate an OCP to find the minimum lap time for a GP2 car and include a detailed multibody vehicle dynamics model with 14 degrees of freedom. However, none of these approaches are intended to work in real-time on a vehicle ECU, but to deliver detailed and close-to-reality results for lap time or for the sensitivity analysis of vehicle setup parameters.

A further necessary and important part in a software stack for autonomous driving is the online re-planning of trajectories to avoid static and dynamic obstacles at high speeds. The literature can be structured into the three fields:

- "separated/two-step trajectory planning", where the velocity calculation is a subsequent process of the path planner [18]–[20].
- "combined trajectory planning", optimizing both path and velocity at the same time [21]–[23].
- "Model Predictive Control (MPC) approaches" taking into account the current vehicle state [10], [24]–[27].

In the following, we evaluate the literature according to the implemented features regarding

- spatially and temporally varying friction coefficients,
- powertrain behavior,
- applicability at the limits of vehicle dynamics through fast computation times.

In the spline-based approach of Mercy et al. [23] trajectories for robots operating at low velocities are optimized. The calculation times of the general NLP-solver Interior Point OPTimizer (IPOPT) [28] range up to several hundred milliseconds, which is too long for race car applications. Another general NLP-formulation is done by Svensson et al. [29]. The latter describe a planning approach for safety trajectories of automated vehicles, which they validate experimentally in simulations for maximum velocities of $30\,\mathrm{km\,h^{-1}}$, leveraging the general nonlinear optimal control toolkit ACADO [30].

Huang et al. [18] describe a two-step approach: first determining the path across discretized available space and then calculating a sufficient velocity. Also, Meng et al. [19] leverage a decoupled approach using a quadratic formulation for the speed profile optimization, reaching real-time capable calculation times below $0.1\,\mathrm{s}$ in this step. Nevertheless, both publications deal with low vehicle speeds in simulations of max. $60\,\mathrm{km\,h^{-1}}$. Furthermore, Zhang et al. [20] implement a two-step algorithm where they use MTSOS [31] for the speed profile generation within several milliseconds for path lengths of up to $100\,\mathrm{m}$. The speed-profile optimization framework MTSOS developed by Lipp and Boyd [31] works for fixed paths leveraging a change of variables. As in the aforementioned publications, they consider a static friction coefficient and neglect the maximum available power of the car. The same is true for the MPC algorithm by Carvalho et al. [24]. They plan trajectories considering the driving dynamics of a bicycle model, neglecting physical constraints stemming from the powertrain, like maximum available torque or power. This is a major drawback for our application, as the DevBot 2.0 is often operating at the power limit of its electric machines.

Subosits and Gerdes [21] formulate a Quadratically Constrained Quadratic Problem (QCQP) replanning path and velocity of a race car at spatially fixed points on the track to avoid static obstacles. They consider a constant friction potential and the maximum available vehicle power. However, the obstacles need to be known in advance before the journey commences, and must be placed at a decent distance from the replanning points to allow the algorithm to find a feasible passing trajectory, given the physical constraints. In order to reach fast calculation times, Alrifaee et al. [25] use a sequential linearization technique for real-time-capable trajectory optimization. They consider the friction maxima, with included velocity dependency that they determine beforehand. This dependency is assumed to be globally constant, thus neglecting the true track conditions during driving. Their experimental results stem from simulations with peak computation times of several hundred milliseconds on a desktop PC.

Considering variable friction on the road is attracting more attention, as it is an emergency-relevant feature for passenger cars and a performance-critical topic for race cars. Therefore, Svensson et al. [22] describe an adaptive trajectory-planning and optimization approach. They pre-sample trajectory primitives to avoid local optima in subsequent SQPs stemming mainly from avoidance maneuvers to the suboptimal side of an obstacle. The vehicle adaptively reacts to a varying friction potential on the road at speeds of up to $100\,\mathrm{km\,h^{-1}}$. The resulting problem is solved using simulations in MATLAB,

so no information about the calculation speed on embedded hardware is given.

Stahl et al. [4] describe a two-step, multi-layered graph-based path planner. This approach allows for functionalities such as following other vehicles and overtaking maneuvers, also in non-convex scenarios at a high update rate. We use this path planner to generate the inputs for the velocity-optimization algorithm.

B. Contributions

In this paper we contribute to the state of the art in the field of real-time-capable trajectory planning with the following content.

(1) We formulate a tailored mpSQP algorithm capable of adaptive velocity planning for race cars operating at the limits of handling, and at velocities above $200\,\mathrm{km\,h^{-1}}$. The planner computes velocity profiles for various paths using the path planner [4] in real time on the target hardware, an NVIDIA Drive PX2 [32] being an ECU already proven for autonomous driving. The adaptivity refers to the multi-parametric input to the planner, depending on the vehicle's environment. The quadratic subproblems within the mpSQP are handled using the OSQP [11] solver. Its primal and dual infeasibility detection for convex problems [33] was integrated to flag up (as fast as possible) offered paths which could not feasibly be driven [4].

(2) With the formulation of an mpSQP optimization algorithm, it is possible to integrate our race ES, described in our previous works [8], [34]. The necessary variable power parameters are forwarded to the velocity planner and considered as a hard constraint, see Fig. 1. In the case of electric race cars, such an ES is vital in order not to overstress the powertrain thermodynamically.

(3) We further allow the friction coefficient on the race track to vary spatially as well as temporally [7]. Therefore, global limits of the allowed longitudinal and lateral acceleration of the vehicle are omitted. This improves the achievable lap time significantly as the tires are locally exploited to their maximum. Via the temporal variation of the friction limits, we take into account varying grip due to, e.g., warming tires or changing weather conditions.

(4) To boost the solver selection for similar projects dealing with trajectory optimization within the community, we compare the efficiency of different solver types regarding calculation speed and solution quality. Therefore, we solve the quadratic subproblems in our mpSQP using a first-order Alternating Direction Method of Multipliers (ADMM) implemented in the OSQP-framework. Its results are compared to the active-set solver QP Online Active SEt Strategy (qpOASES) [35]. We contrast both SQPs with a direct solution of the nonlinear velocity optimization problem with the open-source, second-order interior point solver IPOPT [28], interfaced by the symbolic framework CasADi [36].

Section II introduces the mathematical background of an SQP method to solve an NLP. In the following Section III, the nonlinear equations of our velocity planner are introduced. We explain their efficient incorporation within an mpSQP and explain details about the recursive feasibility of our optimization problem. The Results section shows the realization of the ES and the handling of variable friction by our velocity-optimization algorithm. Furthermore, we contrast different solvers in terms of their runtime and solution quality.

II. Preliminaries

In this section, the mathematical background to an SQP optimization method to solve local approximations of an NLP with objective function $J(\boldsymbol{o})$, $h_b(\boldsymbol{o})$ and $g_c(\boldsymbol{o})$, denoting equality and inequality constraints of scalar quantity b and c, and optimization variables $\boldsymbol{o}$ is introduced.

The standard form of a Nonlinear Optimal Control Problem (NOCP) is given by [36], [37]:

$$\min \; J(\underbrace{x(s), u(s)}_{\boldsymbol{o}}) \tag{1}$$

$$\text{s.t. } \frac{\mathrm{d}x(s)}{\mathrm{d}s} = f(x(s), u(s)) \tag{2}$$

$$h_b(\boldsymbol{o}) = 0 \tag{3}$$

$$g_c(\boldsymbol{o}) \leq 0. \tag{4}$$

The independent space variable s describes the distance along the vehicle's path in our problem. The function $f(x(s), u(s))$ specifies the derivatives of the state variable $x(s)$ as a function of the state $x(s)$ and the control input $u(s)$.

The standard form of a QP is expressed as [38]

$$\begin{aligned} \min \quad & \frac{1}{2}\boldsymbol{z}_{\mathrm{qp}}^T \boldsymbol{P} \boldsymbol{z}_{\mathrm{qp}} + \boldsymbol{q}^T \boldsymbol{z}_{\mathrm{qp}} \\ \text{s.t.} \quad & \boldsymbol{l} \leq \boldsymbol{A}\boldsymbol{z}_{\mathrm{qp}} \leq \boldsymbol{u}, \end{aligned} \tag{5}$$

where $\boldsymbol{z}_{\mathrm{qp}}$ is the optimization vector, matrix $\boldsymbol{P}$ is the Hessian matrix of the discretized objective $J(\boldsymbol{o}^k)$ and the vector $\boldsymbol{q}^T$ equals the Jacobian of the discretized objective $\nabla J(\boldsymbol{o}^k)$ with iterate k. Matrix $\boldsymbol{A}$ contains the linearized versions of the constraints h_b and g_c in the optimization problem. Their upper and lower bounds are summarized in both vectors, $\boldsymbol{l}$ and $\boldsymbol{u}$.

In an SQP method, the linearization point $\boldsymbol{o}^k$ is updated after every QP iteration k using [39]

$$\boldsymbol{o}^{k+1} = \boldsymbol{o}^k + \alpha \boldsymbol{z}_{\mathrm{qp}} \tag{6}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}_{\mathrm{qp}}^k \tag{7}$$

$$\boldsymbol{z}_{\mathrm{qp}} = \boldsymbol{o} - \boldsymbol{o}^k. \tag{8}$$

In the quadratic subproblem, a solution for $\boldsymbol{z}_{\mathrm{qp}}$ is computed. We chose to initialize the Lagrange multiplier vector $\boldsymbol{\lambda}^{k+1}$ using the previous QP solution as stated in the local SQP algorithm in [40].

On the one hand, the steplength parameter α must be calculated in order to perform a large step in the direction of the optimum $\boldsymbol{o}^*$ for fast convergence. On the other, α must be small enough to not skip or oscillate around $\boldsymbol{o}^*$. It is therefore necessary to define a suitable merit function, taking into account the minimization of the objective function as well as the adherence of the constraints [37], [39]. As it is hard to find such a merit function, we use the SQP Root Mean Square

Error (RMSE) $\bar{\varepsilon}_{\mathrm{SQP}}$ as well as the SQP infinity norm error $\hat{\varepsilon}_{\mathrm{SQP}}$ to determine whether a stepsize α is suitable or not,

$$\bar{\varepsilon}_{\mathrm{SQP}} = \frac{1}{K} \left\| \boldsymbol{o}^{k+1} - \boldsymbol{o}^{k} \right\|_2 \leq \bar{\varepsilon}_{\mathrm{SQP,tol}} \tag{9}$$

$$\hat{\varepsilon}_{\mathrm{SQP}} = \left\| \boldsymbol{o}^{k+1} - \boldsymbol{o}^{k} \right\|_\infty \leq \hat{\varepsilon}_{\mathrm{SQP,tol}}, \tag{10}$$

where K denotes the number of elements in $\boldsymbol{o}^k$. In case one of the two errors $\varepsilon_{\mathrm{SQP}}$ increases when applying (6), the counting variable γ is increased and, therefore, α is reduced until the tolerance criteria values $\bar{\varepsilon}_{\mathrm{SQP,tol}}$ and $\hat{\varepsilon}_{\mathrm{SQP,tol}}$ are met:

$$\alpha = \beta^{\gamma}. \tag{11}$$

The parameter $\beta \in]0, 1[$ is to be tuned problem-dependent as the Armijo rule states [37] with $\gamma \in [0; 1; 2; ...]$.

To bring the objective $J(\boldsymbol{o})$ and the necessary nonlinear constraints $h_b(\boldsymbol{o})$ and $g_c(\boldsymbol{o})$ into the mathematical form of a QP (5), they are discretized and approximated quadratically or linearly, respectively, using Taylor series expansions in the form

$$\begin{aligned} J(\boldsymbol{o}) \approx & \frac{1}{2}(\boldsymbol{o} - \boldsymbol{o}^k)^T \boldsymbol{P}(\boldsymbol{o}^k)(\boldsymbol{o} - \boldsymbol{o}^k) + \\ & \nabla J(\boldsymbol{o}^k)(\boldsymbol{o} - \boldsymbol{o}^k) + J(\boldsymbol{o}^k) \end{aligned} \tag{12}$$

and

$$g_c(\boldsymbol{o}) \approx \nabla g(\boldsymbol{o}^k)(\boldsymbol{o} - \boldsymbol{o}^k) + g(\boldsymbol{o}^k). \tag{13}$$

III. Optimization-Based Velocity Planner

This section describes the implemented point mass model, the used objective function, and the constraints necessary to optimize the velocity on the available paths. The point mass model was chosen, as it is commonly used to describe the driving dynamics in the automotive context. Due to its simplicity, it delivers a small number of optimization variables and constraints. Therefore, quick solver runtimes can be achieved. It still delivers quite accurate results for the task of pure velocity optimization [13].

The concept of the optimization-based algorithm is to plan velocities with inputs from other software modules, cf. Section I. We do not deal with sensor noise in the planner but in the vehicle dynamics controller [5], which receives the trajectory input. The trajectory planning module, consisting of a path planner [4] and the presented velocity optimization algorithm, always keeps the first discretization points of a new trajectory constant with the solution from a previous planning step. After matching the current vehicle position to the closest coordinate in the previously planned trajectory, a new plan starting from this position is made within the remaining part of a new trajectory. Through this, two control loops can be omitted, and prevent from unnecessary inferences in the planning and the control module.

A. Nonlinear problem

This subsection presents the nonlinear velocity optimization problem, structured into its system dynamics, equality and inequality constraints as well as its objective function.

1) System dynamics: Let us first introduce the system dynamics of the point mass model for our physical vehicle state $v(s)$. Newton's second law for a point mass m_{v} states

$$m_{\mathrm{v}} v(s) \frac{\mathrm{d}v(s)}{\mathrm{d}s} = m_{\mathrm{v}} a_{\mathrm{x}}(s). \tag{14}$$

With the derivative of the kinetic energy,

$$\frac{\mathrm{d}E_{\mathrm{kin}}(s)}{\mathrm{d}s} = \frac{1}{2} m_{\mathrm{v}} \frac{\mathrm{d}v^2(s)}{\mathrm{d}s} = m_{\mathrm{v}} v(s) \frac{\mathrm{d}v(s)}{\mathrm{d}s}, \tag{15}$$

the system dynamics are given by the longitudinal acceleration

$$a_{\mathrm{x}}(s) = \frac{1}{m_{\mathrm{v}}} \frac{\mathrm{d}E_{\mathrm{kin}}(s)}{\mathrm{d}s}. \tag{16}$$

The force $F_{\mathrm{x,p}}(s)$ applied by the powertrain to move the point mass model can be calculated by

$$F_{\mathrm{x,p}}(s) = m_{\mathrm{v}} a_{\mathrm{x}}(s) + c_{\mathrm{r}} v^2(s) \tag{17}$$

where c_{r} is the product of the air density ρ_{a}, the air resistance coefficient c_{w} and the vehicle's frontal area A_{v},

$$c_{\mathrm{r}} = \frac{1}{2} \rho_{\mathrm{a}} c_{\mathrm{w}} A_{\mathrm{v}}. \tag{18}$$

2) Equality and inequality constraints: To improve numerical stability and avoid backward movement, the velocity $v(s)$ is constrained,

$$0 \leq v(s) \leq v_{\max}(s). \tag{19}$$

To ensure that the optimization remains feasible in combination with a moving horizon, the terminal constraint

$$v(s_{\mathrm{f}}) \leq v_{\mathrm{end}} \tag{20}$$

on the last coordinate point s_{f} within the optimization horizon is leveraged. Here, v_{end} denotes the minimal velocity the vehicle can take in the case of maximum specified track curvature $\kappa_{\max}$ at the vehicle's technically maximum possible lateral acceleration $a_{\mathrm{y,max}}$. Therefore,

$$v_{\mathrm{end}} = \sqrt{\frac{a_{\mathrm{y,max}}}{\kappa_{\max}}}. \tag{21}$$

At the beginning of the optimization horizon, the velocity and acceleration must equal the vehicle's target states of the currently executed plan v_{ini} and $a_{\mathrm{x,ini}}$,

$$\begin{aligned} v(s_{\mathrm{s}}) &= v_{\mathrm{ini}}, \\ a_{\mathrm{x,ini}} - \delta_{\mathrm{a}} \leq a_{\mathrm{x}}(s_{\mathrm{s}}) &\leq a_{\mathrm{x,ini}} + \delta_{\mathrm{a}}, \end{aligned} \tag{22}$$

where s_{s} denotes the first coordinate within the moving optimization horizon and δ_{a} a small tolerance to account for numerical imprecision.

As the vehicle's maximum braking as well as driving forces are technically limited, the resulting constraints are

$$F_{\min} \leq F_{\mathrm{x,p}}(s) \leq F_{\max}. \tag{23}$$

The negative force constraint $F_{\min}$ does not affect the optimization-problem feasibility, as the DevBot's braking actuators can produce more negative force than the tires can transform.

The electric machine's output power $P(s)$ is computed using

$$P(s) = F_{\mathrm{x,p}}(s) v(s), \tag{24}$$

limited by the available maximum

$$P(s) \leq P_{\max}(s). \tag{25}$$

We highlight that $P_{\max}(s)$ is a space-dependent parameter in contrast to the constant maximum force $F_{\max}$. By this, the given race ES based on our previous works [8], [34] is realized.

To further integrate the tire physics, we interpret the friction potential as a combined, diamond-shaped acceleration limit for the vehicle [7] given by the inequality

$$\left\|(\hat{a}_{\mathrm{x}}(s), \hat{a}_{\mathrm{y}}(s))\right\|_1 \leq 1 + \epsilon(s) \tag{26}$$

where $||\cdot||_1$ denotes the l^1-norm. Furthermore, the normalized longitudinal $\hat{a}_{\mathrm{x}}(s)$ as well as the lateral tire utilizations $\hat{a}_{\mathrm{y}}(s)$ are given by

$$\hat{a}_{\mathrm{x}}(s) = \frac{F_{\mathrm{x,p}}(s)}{m_{\mathrm{v}}} \frac{1}{\bar{a}_{\mathrm{x}}(s)}, \quad \hat{a}_{\mathrm{y}}(s) = \frac{a_{\mathrm{y}}(s)}{\bar{a}_{\mathrm{y}}(s)}. \tag{27}$$

Here, we use $\bar{a}_{\mathrm{x/y}}(s)$ to indicate a variable, space-dependent acceleration potential in both longitudinal and lateral direction, which is to be leveraged [7]. The lateral acceleration $a_{\mathrm{y}}(s)$ reads [41]

$$a_{\mathrm{y}}(s) = \kappa(s) v^2(s) \tag{28}$$

accounting for the target path geometry by the variable road curvature parameter $\kappa(s)$.

In (26) the slack variable $\epsilon(s)$ ensures the recursive feasibility of the optimization problem: details are given in Subsection III-D. We constrain the slack variable $\epsilon(s)$ by

$$0 \leq \epsilon(s) \leq \epsilon_{\max} \tag{29}$$

to prohibit negative values and additionally keep the physical tire exploitation within a specified maximum.

Similarly to (21), the longitudinal and lateral acceleration limits at the end of the optimization horizon $\bar{a}_{\mathrm{x}}(s_{\mathrm{f}})$ and $\bar{a}_{\mathrm{y}}(s_{\mathrm{f}})$ must be set to the lowest physically possible acceleration limits $\bar{a}_{\mathrm{x/y,min}}$ for the current track conditions,

$$\bar{a}_{\mathrm{x/y}}(s_{\mathrm{f}}) \leq \bar{a}_{\mathrm{x/y,min}}. \tag{30}$$

3) Continuous objective function: With the help of the introduced symbols and equations we can now formulate the objective function $J(x(s))$ to minimize the traveling time along the given path:

$$J(x(s)) = \int_0^{s_{\mathrm{f}}} \frac{1}{v(s)} \mathrm{d}s + \frac{\rho_{\mathrm{j}}}{s_{\mathrm{f}}} \int_0^{s_{\mathrm{f}}} \left(\frac{\mathrm{d}^2 v(s)}{\mathrm{d}^2 t} \right)^2 \mathrm{d}s + \frac{\rho_{\epsilon,\mathrm{l}}}{s_{\mathrm{f}}} \int_0^{s_{\mathrm{f}}} \epsilon(s) \mathrm{d}s + \frac{\rho_{\epsilon,\mathrm{q}}}{s_{\mathrm{f}}} \int_0^{s_{\mathrm{f}}} \epsilon^2(s) \mathrm{d}s. \tag{31}$$

We chose the optimization variables $\boldsymbol{o}$ to be the state velocity $v(s)$ as well as the slacks $\epsilon(s)$. The control input to the vehicle $u(s) = F_{\mathrm{x,p}}(s)$ doesn't occur explicitly in the objective function but can be recalculated from the state trajectory $v(s)$, cf. (17).

Minimizing the term $\frac{1}{v(s)}$ is equivalent to the minimization of the lethargy $\frac{\mathrm{d}t}{\mathrm{d}s}$, which can be interpreted as the time necessary to drive a unit distance [13]. To weight the different terms, the penalty parameters ρ are used. These include a jerk penalty ρ_{j}, a slack weight $\rho_{\epsilon,\mathrm{l}}$ on their integral and a penalty $\rho_{\epsilon,\mathrm{q}}$ on the integral of the squared slack values. The linear penalty term on the slack variable $\epsilon(s)$ is necessary to achieve an exact penalty maintaining the original problem's optimum $[v^*(s)\ \epsilon^*(s)]$ if feasible [42]. Similar to a regularization term, the integral of the squared slacks $\epsilon(s)$ is additionally added to improve numerical stability and the smoothness of the results.

B. Multi-parametric Sequential Quadratic Problem

This chapter gives details about the implementation of the NLP given in Subsection III-A as an mpSQP in order to efficiently solve local approximations of the velocity planning problem. We describe how to approximate the nonlinear objective function $J(x(s))$ (31) to achieve a constant and tuneable Hessian matrix within our tailored mpSQP algorithm. Furthermore, we present a method to reduce the number of slack variables $\epsilon(s)$ and the slack constraints (29) therefore necessary.

Our optimization vector $\boldsymbol{z} = \boldsymbol{o}^k$ in a discrete formulation transforms into

$$\boldsymbol{z} = \Big[\underbrace{v_1(s_1) \dots v_{M-1}(s_{M-1})}_{\boldsymbol{v}} \quad \underbrace{\epsilon_0(s_0) \dots \epsilon_{N-1}(s_{N-1})}_{\boldsymbol{\epsilon}} \Big]^T \in \mathbb{R}^{K \times 1} \tag{32}$$

where $K = M - 1 + N$ where M denotes the number of discrete velocity points v_m and N the number of discrete slack variables ϵ_n used in the tire inequality constraints within one optimization horizon. We drop the dependency of $\boldsymbol{z}$ on s_m in the following for the sake of readability. The velocity variable v_0 is removed from the vector $\boldsymbol{z}$ as it is a fixed parameter equaling the velocity planned in a previous SQP $l-1$ for the current position.

To reduce the problem size, we apply one slack variable ϵ_n to multiple consecutive discrete velocity points v_m. This is done uniformly and leads to:

$$\Big[\underbrace{v_1 \ \dots \ v_{\tilde{N}}}_{\epsilon_0} \quad \underbrace{v_{\tilde{N}+1} \ \dots \ v_{2\tilde{N}}}_{\epsilon_1} \quad \underbrace{v_{2\tilde{N}+1} \ \dots \ v_{M-1}}_{\dots} \Big] \tag{33}$$

Here, $\tilde{N}$ is a problem-specific parameter setting a trade off between the number of optimization variables and therefore the calculation speed and accuracy in the solution.

From domain knowledge we know that the objective function can be approximated in the form

$$J(\boldsymbol{z}) \approx \underbrace{\left\|\boldsymbol{v} - \boldsymbol{v}_{\max}\right\|_2^2}_{J_{\mathrm{v}}} + \underbrace{\rho_{\mathrm{j}} \left\|\Delta \boldsymbol{v}\right\|_2^2}_{J_{\mathrm{j}}} + \underbrace{\rho_{\epsilon,\mathrm{l}} \left\|\zeta \boldsymbol{\epsilon}\right\|_1^2}_{J_{\epsilon,\mathrm{l}}} + \underbrace{\rho_{\epsilon,\mathrm{q}} \left\|\zeta \boldsymbol{\epsilon}\right\|_2^2}_{J_{\epsilon,\mathrm{q}}}. \tag{34}$$

The slack variables are transformed via the constant factor ζ; how this is selected is discussed at the end of this section. By using the l^2-norm of the vector difference of $\boldsymbol{v}$ and $\boldsymbol{v}_{\max}$, the solution tends to minimize the travel time along the path. Still, this formulation in combination with (19) makes the car keep a specified maximum velocity $v_{\max}(s)$ dependent on the current position s to react, e.g., to other cars. To control the

vehicle's jerk behavior, we add the Tikhonov regularization term $\rho_\mathrm{j} \left\| \Delta \boldsymbol{v} \right\|_2^2$ [38] that approximates the second derivative of $\boldsymbol{v}$. The tridiagonal Toeplitz matrix $\Delta \in \mathbb{R}^{M-3 \times M-1}$ contains the diagonal elements $\begin{pmatrix} 1 & -2 & 1 \end{pmatrix}$ [38]. By the l^1-norm within $J_{\epsilon,\mathrm{l}}$, the summation of the absolute values of the slack variable vector entries in $\boldsymbol{\epsilon}$ is achieved. To improve the numerical conditioning of the problem, their l^2-norm is added additionally by $J_{\epsilon,\mathrm{q}}$.

For the specific choice of cost function in (12), the Hessian matrix $\boldsymbol{P} \in \mathbb{R}^{K \times K}$ does not depend on $\boldsymbol{z}$. The condition number σ_H of the Hessian $\boldsymbol{P}$ is tuned to be as close to 1 as possible via the penalties ρ_j and $\rho_{\epsilon,\mathrm{l}}$ as well as ζ denoting the unit conversion factor of the tire slack variable values in $\boldsymbol{\epsilon}$ to SI units:

$$\boldsymbol{P} = \left[\begin{array}{ccc|cc} \ddots & \ddots & 0 & & \\ \ddots & c_j \rho_\mathrm{j} & \ddots & 0 & \\ 0 & \ddots & \ddots & & \\ \hline & 0 & & 2\rho_{\epsilon,\mathrm{q}}\zeta^2 & 0 \\ & & & 0 & \ddots \end{array}\right] \tag{35}$$

The function $c_j \rho_\mathrm{j}$ represents different constant entries j that are linearly dependent on ρ_j. This upper left part of $\boldsymbol{P}$ is a bisymmetric matrix with constant entries on its main diagonal, as well as on its first and second ones.

By using the approach of multi-parametric programming, we can vary several problem parameters online in the SQP (Section II) without changing the problem size. These parameters include the

- spatial discretization length Δs_m.
- curvature of the local path $\kappa(s_m)$ [4].
- maximum allowed velocity $v_\mathrm{max}(s_m)$.
- power limitations $P_\mathrm{max}(s_m)$ stemming from a global race strategy taking energy limitations into account [8], [34].
- longitudinal and lateral acceleration limits $\bar{a}_\mathrm{x}(s_m)$, $\bar{a}_\mathrm{y}(s_m)$ [7].

C. Variable acceleration limits

To fully utilize the maximum possible tire forces, a time- and location-dependent map of the race track, containing the maximum possible accelerations, is generated. The acceleration limits can be interpreted as vehicle-related friction coefficients, cf. Subsection III-A. The 1D map along the global coordinate s_glo with variable discretization step length stores the individual acceleration limitations $\Sigma_{\bar{a}}(s_\mathrm{glo})$ in longitudinal and lateral directions. The acceleration limits are used in the selected local path as the parameters $\bar{a}_{\mathrm{x/y}}(s)$ (27). It is important to know that while the vehicle proceeds, the target path is updated constantly, i.e., the global coordinates s_glo selected as the local path s vary permanently in subsequent velocity optimizations. The path planning guarantees to reuse the first few global coordinates s_glo from a previous timestep t^0 as the starting coordinates s_m of the subsequently chosen path at t^1. Still, the path coordinates s_m at the end of the planning horizon are not guaranteed to precisely match all of the previously used global coordinates s_glo since the path might change. Therefore, the requested coordinates in the acceleration map do also vary slightly, but are matched to the same local coordinate indices in s_m within the local path in subsequent timesteps. This leads to differences in the acceleration limits $\bar{a}_{\mathrm{x/y}}(s_m)$ in subsequent planning iterations for identical local path indices m and therefore probably to infeasible problems in terms of optimization, see Section III-D.

The nature of the discretization problem is illustrated in Fig. 3. The given example provides stored acceleration limits $\Sigma_{\bar{a}}(s_\mathrm{glo})$ in $10\,\mathrm{m}$ steps. The planning horizon ranges from $s_\mathrm{glo} = 0\,\mathrm{m}$ to $300\,\mathrm{m}$. Therefore, we show a snippet of the end of the planning horizon ($s_\mathrm{glo} = 200\,\mathrm{m}$ to $270\,\mathrm{m}$) as the discretization issues are clearly visible here. At timestep t^0, the planning algorithm requests the stored acceleration limits $\Sigma_{\bar{a}}(s_\mathrm{glo})$ every $5.5\,\mathrm{m}$, starting at $s_\mathrm{glo} = 0\,\mathrm{m}$. Within the depicted path snippet, the subsequent iteration at t^1 starts at a shift of $2.0\,\mathrm{m}$ with the same stepsize.

The simple approach of directly obtaining the local acceleration limits from the stored values $\Sigma_{\bar{a}}(s_\mathrm{glo})$ by applying zero-order hold comes with drawbacks. A slight shift in the global coordinate selection can lead to situations where the acceleration limits $\bar{a}_{\mathrm{x/y}}(s_m)$ differ between subsequent timesteps (t^0, t^1) in the local path. If the subsequent acceleration limits $\bar{a}_{\mathrm{x/y}}(s_m)$ at t^1 are smaller, they can lead to infeasibility. In Fig. 3, the gray areas highlight the situations where the obtained acceleration limits $\bar{a}_{\mathrm{x/y}}(s_m)$ at t^1 are smaller compared to t^0 for identical local path indices m.

To mitigate the discretization effects, we propose an interpolation scheme leading to the values $\tilde{\Sigma}_{\bar{a}}(s_\mathrm{glo})$. It applies linear interpolation between the stored acceleration limits $\Sigma_{\bar{a}}(s_\mathrm{glo})$ but acts cautiously in the sense that it always underestimates the actual values $\Sigma_{\bar{a}}(s_\mathrm{glo})$. This can be seen from $s_\mathrm{glo} = 200\,\mathrm{m}$ to $210\,\mathrm{m}$, where the value is kept constant instead of interpolating between 11 and $12\,\mathrm{m\,s^{-2}}$, and from $s_\mathrm{glo} = 230\,\mathrm{m}$ to $240\,\mathrm{m}$ where the algorithm adapts to the decreasing values although the stored value is $13\,\mathrm{m\,s^{-2}}$. Then, zero-order hold is applied to this conservatively interpolated line. The limits obtained at t^1 often lie above $\bar{a}_{\mathrm{x/y}}(s_m)$ at t^0, which allow higher accelerations $a_{\mathrm{x/y}}(s_m)$ than expected at t^0 and thus compensating for overestimated areas (gray areas).

The acceleration limits $\Sigma_{\bar{a}}(s_\mathrm{glo})$ are constantly updated by an estimation algorithm [7] and are therefore also considered time-variant. During the update process, it must be guaranteed that the update does not lead to an infeasible vehicle state for the velocity-planning algorithm, e.g., when the vehicle is approaching a turn already utilizing full tire forces under braking, and the acceleration limits are suddenly decreased in front of the vehicle. Therefore, the updates only take place outside the planning horizon of the algorithm.

The difference between subsequently obtained acceleration limits $\bar{a}_{\mathrm{x/y}}(s_m)$ at identical local path coordinates s_m can be controlled via the maximum change between the stored values $\Sigma_{\bar{a}}(s_\mathrm{glo})$. The slope of the interpolated values $\tilde{\Sigma}_{\bar{a}}(s_\mathrm{glo})$ can be used to calculate the maximum error when applying a particular step size Δs_m in path planning.

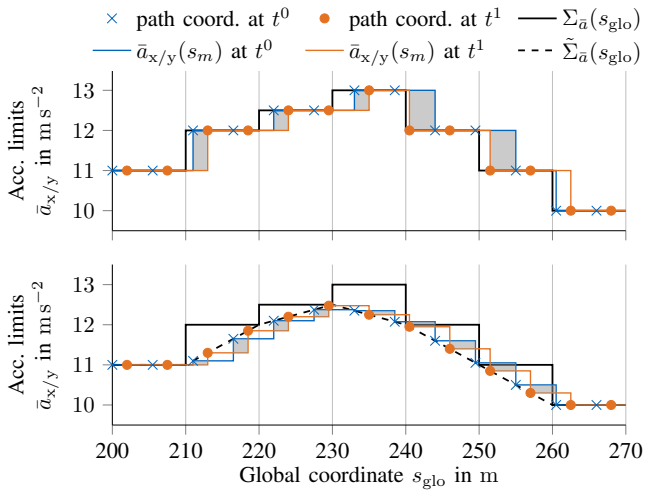


Fig. 3. Diagram of acceleration limits for two subsequently planned paths; with pure readout values $\Sigma_{\bar{a}}(s_{\mathrm{glo}})$ (top) and with the proposed interpolation scheme $\tilde{\Sigma}_{\bar{a}}(s_{\mathrm{glo}})$ (bottom). Gray areas show where the subsequently planned path receives decreased maximum acceleration limits $\bar{a}_{\mathrm{x/y}}(s_m)$ due to tolerances in the spatial discretization.

D. Recursive feasibility

The minimum-time optimization problem tends to produce solutions with many active constraints, as it maximizes the tire utilization. It follows from this property, that ensuring recursive feasibility is a highly relevant aspect for the application of such an algorithm, and should be achieved via the terminal constraints (21) and (30) by making a worst-case assumption about the curvature $\kappa(s_{\mathrm{f}})$ and acceleration limits $\bar{a}_{\mathrm{x/y}}(s_{\mathrm{f}})$ at the end of the optimization horizon. This property holds as long as the optimization problem is shifted by an integer multiple of the discretization Δs_m while the relations of the local path coordinates s_m with the curvature $\kappa(s_m)$ and the acceleration limits $\bar{a}_{\mathrm{x}}(s_m)$ and $\bar{a}_{\mathrm{y}}(s_m)$ remain constant. This cannot be ensured since the path planner [4] might slightly vary the target path due to an obstacle entering the planning horizon, or due to discretization effects. This leads to a deviation in the curvature profile $\kappa(s_m)$ and deviations in the admissible accelerations $\bar{a}_x(s_m)$ and $\bar{a}_y(s_m)$ since the local coordinate s_m might refer to a different point in global coordinates s_{glo} now, see Fig. 3.

To mitigate this deficiency, we introduce slack variables ϵ based on the exact penalty function approach [42]. This strategy ensures that the hard-constrained solution is recovered if it is feasible, and therefore the solution is not altered by addition of the slacks unless it is mandatory. The nature of the combined acceleration constraint (26) allows for a straightforward interpretation of the slack variables as a violation of ϵ in $\%$. Together with the upper bound on the slack variables in (29), we can therefore state that the optimization problem is always feasible as long as the maximum required violation is limited to $\epsilon_{\max}$. In case no solution is found within the specified tolerance band, a dedicated failure-handling strategy is employed within the trajectory planning framework. We wish to point out that a suitable scaling of the slack variables is crucial to achieve sufficiently tight tolerances $\varepsilon_{\mathrm{QP,tol}}$ when using a numerical QP solver. We therefore employ a variable transformation with $\epsilon = \zeta\epsilon_n$ and optimize over ϵ_n instead. Realistic values for the maximum slack variable $\epsilon_{\max}$ were found to be around $3\,\%$ in extensive simulations on different race tracks (Berlin (Germany), Hong Kong (China), Indianapolis Motor Speedway (USA), Las Vegas Motor Speedway (USA), Millbrook (UK), Modena (Italy), Monteblanco (Spain), Paris (France), Upper Heyford (UK), Zalazone (Hungary)) and obstacle scenarios. We consider this to be an acceptable tolerance level and believe it will be difficult to achieve significantly tighter guarantees in the face of the scenario complexity we tackle in [4].

TABLE I
EMERGENCY- AND PERFORMANCE-SQP PARAMETRIZATION.

Parameter	Unit	Value	
		Performance	Emergency
M	-	115	50
N	-	12	5
δ_{a}	$\mathrm{m\,s^{-2}}$	0.1	inactive
$\epsilon_{\max}$	$\%$	3.0	3.0
ρ_{j}	-	$3e^2$	0.0
$\rho_{\epsilon,\mathrm{l}}$	$\mathrm{m^2\,s^{-2}}$	$1e^5$	$5e^4$
$\rho_{\epsilon,\mathrm{q}}$	$\mathrm{m^2\,s^{-2}}$	$1e^4$	$1e^3$
$n_{\mathrm{SQP,max}}$	-	20	20
$\Delta t_{\max}$	ms	300	100
β	-	0.5	0.5
$\bar{\varepsilon}_{\mathrm{SQP,tol}}$	-	$1e^0$	$1.5e^0$
$\hat{\varepsilon}_{\mathrm{SQP,tol}}$	-	$1e^0$	$1.5e^0$
$\varepsilon_{\mathrm{QP,tol}}$	-	$1e^{-2}$	$1e^{-2}$

IV. Results

In this section, the results achieved with the presented velocity mpSQP will be presented. We conducted the experiments on our Hardware-in-the-Loop simulator, which consists of a Speedgoat Performance real-time target machine, where validated physics models of the real race car in combination with realistic sensor noise are implemented. An additional NVIDIA Drive PX2 receives this sensor feedback and calculates the local trajectories. A Speedgoat Mobile real-time target machine transforms this trajectory input into low level vehicle commands to close the loop to the physics simulation. Therefore, we used the DevBot 2.0 data: $m_{\mathrm{v}} = 1160\,\mathrm{kg}$, $P_{\max} = 270\,\mathrm{kW}$, $F_{\max} = 7.1\,\mathrm{kN}$, $F_{\min} = -20\,\mathrm{kN}$, $c_{\mathrm{r}} = 0.85\,\mathrm{kg\,m^{-1}}$. The results in this section have been produced with the velocity planner parametrizations given in Table I.

We show results for two types of offered paths: performance and emergency. The emergency path is identical to the performance one, except for a coarser spatial discretization, and the fact that the velocity planner tries to stop as soon as possible on the emergency line. The optimization of the emergency line requires therefore fewer variables $M + N$. This formulation reduces the necessary calculation time of the emergency line that must be updated more frequently for safety reasons.

A. Objective function design

To explain the chosen values of the penalty weights ρ, we show the values of the single objective function terms in $J(\boldsymbol{z})$

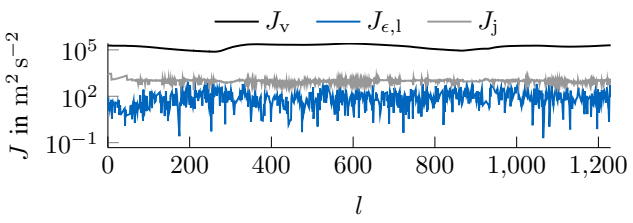


Fig. 4. Cost terms of the objective function J being minimized within the performance velocity profile. J_v shows the most significant influence on the solution due to its value range compared to the other objective terms J_j, $J_{\epsilon,\mathrm{l}}$ and $J_{\epsilon,\mathrm{q}}$ (not displayed as it equals almost 0). Symbol l denotes the number of the optimized velocity profiles.

being minimized during the calculation of the performance velocity profile in Fig. 4. The symbol l denotes the number of the optimized velocity profiles during the driven lap (including the race start) as well as the vehicle's stopping scenario. It can clearly be seen that the velocity term J_v has the highest relative impact on the optimum solution. Its value range is at least two orders of magnitude higher than the slack penalty terms $J_{\epsilon,\mathrm{l}}$, $J_{\epsilon,\mathrm{q}}$ (not displayed) and the jerk penalty J_j. The penalty weight $\rho_{\epsilon,\mathrm{l}}$ on the linear slack term was chosen to increase the value of $J_{\epsilon,\mathrm{l}}$ to be one order of magnitude higher than J_v if ϵ_max was fully exploited on all the slack variables ϵ_n. Therefore, $J_{\epsilon,\mathrm{l}}$ prevents the solver from permanent usage of tire slack ϵ for further lap time gains. As $\rho_{\epsilon,\mathrm{q}}$ is applied to the squared values ϵ_n, it is sufficient to keep the magnitude of $\rho_{\epsilon,\mathrm{q}}$ one order smaller than $\rho_{\epsilon,\mathrm{l}}$. To provide a smooth velocity profile, we set the jerk penalty ρ_j to increase the value of J_j to be higher than the slack penalty terms during normal operation. By this, effects on a possible lap time loss stay as small as possible, whilst a smoothing effect in the range of numerical oscillations on the velocity and acceleration profile is still visible.

We further integrated a calculation time limit Δt_max for the velocity optimization and a maximum SQP iteration number $n_\mathrm{SQP,max}$. In case of reached limits, the algorithm would return the last suboptimal but driveable solution. The SQP never reached these limits during our experiments, and they can be considered as safety limitations. Instead, the SQP-algorithm always terminated due to the reached tolerance criteria $\bar{\varepsilon}_\mathrm{SQP,tol}$ and $\hat{\varepsilon}_\mathrm{SQP,tol}$, cf. (9) and (10).

B. Energy Strategy

As stated in Subsection III-B, the presented mpSQP is able to implement our global race ES [8], [34]. The ES is pre-computed offline and re-calculated online due to disturbances, unforeseen events during the race and model uncertainties. Through this, we account for the limited amount of stored battery energy and the thermodynamic limitations of the electric powertrain. Therefore, the ES delivers the maximum permissible power $P_\mathrm{max}(s_\mathrm{glo})$ in order to reach the minimum race time, see Fig. 1. It takes the following effects into account:

- the vehicle dynamics in the form of an NDTM including a nonlinear tire model;
- the electric behavior of battery, power inverters and electric machines, i.e., the power losses of these components during operation;
- the thermodynamics within the powertrain transforming power losses into temperature contribution.

Fig. 5 depicts the output of the ES computed offline (top). We varied the amount of energy available for one race lap by the three values $100\,\%$ ($E_\mathrm{glo,100}(s_\mathrm{glo})$), $80\,\%$ ($E_\mathrm{glo,80}(s_\mathrm{glo})$) and $60\,\%$ ($E_\mathrm{glo,60}(s_\mathrm{glo})$). The optimal power usage $P_\mathrm{glo}(s_\mathrm{glo})$ belonging to these energy values $E_\mathrm{glo}(s_\mathrm{glo})$ is depicted in the first diagram. The positive values in $P_\mathrm{glo}(s_\mathrm{glo})$ become the parametric input of the power constraint within the velocity optimization (25). By this formulation, we only restrict the vehicle's acceleration power but leave the braking force unaffected. This experiment consists of one race lap with constant maximum acceleration values $\bar{a}_\mathrm{x/y}$ on the Modena (Italy) race circuit.

The power usage $P_\mathrm{loc,80}(s_\mathrm{glo})$ locally planned by the mpSQP is shown in the middle plot of Fig. 5. The positive power values in $P_\mathrm{loc,80}(s_\mathrm{glo})$ remained below the maximum power request allowed by the global strategy $P_\mathrm{glo,80}(s_\mathrm{glo})$. Differences between the globally optimal power usage $P_\mathrm{glo}(s_\mathrm{glo})$ and the locally transformed power $P_\mathrm{loc}(s_\mathrm{glo})$ stem from different model equations in both - offline ES and online mpSQP - optimization algorithms. The velocity planner's point mass is more limited in its combined acceleration potential due to the diamond-shaped acceleration constraint (26) compared to the vehicle dynamics model (NDTM) in the ES. The NDTM overshoots the dynamical capability of the car slightly in edge cases due to parameter-tuning difficulties. Furthermore, the effect of longitudinal wheel-load transfer is considered within the NDTM. For these reasons, the point mass model accelerates less but meets the maximum admissible power $P_\mathrm{glo}(s_\mathrm{glo})$ on the straights. The accumulated error in this experiment can be expressed by the energy demand E_loc resulting from

$$E_\mathrm{loc} = \int F_\mathrm{x,p}(s_\mathrm{glo})\mathrm{d}s_\mathrm{glo}. \tag{36}$$

In total, an energy amount of $E_\mathrm{glo} = 1.27\,\mathrm{kW\,h}$ was allowed whereas $E_\mathrm{loc} = 1.07\,\mathrm{kW\,h}$ was used, implying $15.8\,\%$ drift. With the help of a re-calculation strategy adjusting the ES during the race, this error can be significantly reduced. This feature was switched off during this experiment to isolate the working principle of the race ES, i.e., the interaction between global and local planners.

The effect of the ES on the vehicle speed $\boldsymbol{v}_\mathrm{P80}$ is depicted in Fig. 6. The left part (A) of both plots consists of a scenario where the race car is accelerating with the maximum available machine power of $P_\mathrm{max} = 270\,\mathrm{kW}$ and a subsequent straight where the ES forces the vehicle to coast ($P_\mathrm{glo,80}(s_m) = P_\mathrm{loc,80}(s_m) = 0\,\mathrm{kW}$). Acceleration without any power restriction on this straight would have resulted in the velocity curve $\boldsymbol{v}_\mathrm{P100}$ meaning a slightly higher top speed by approx. $4\,\%$ or $8\,\mathrm{km\,h^{-1}}$. The second part (B) of the shown planning horizon ensures recursive feasibility. Here, we force the velocity variable $v(s_N)$ to reach v_end (20).

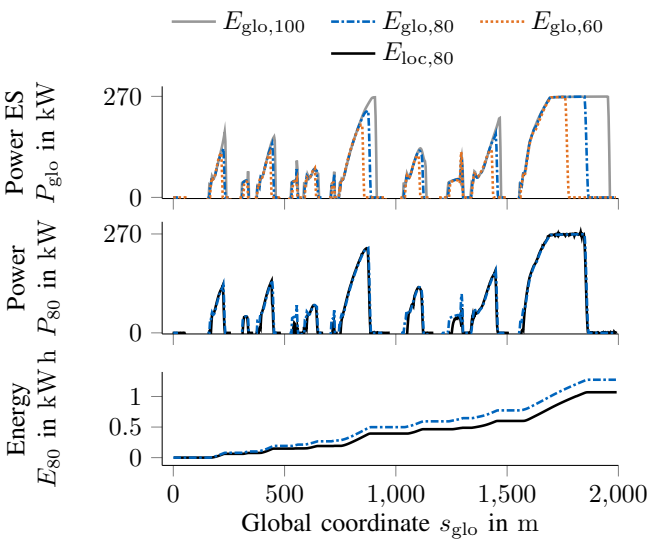


Fig. 5. Top: Optimal power usage $P_{\mathrm{glo}}(s_{\mathrm{glo}})$ from the ES that should be requested whilst driving on the Modena (Italy) race track. Mid: Power planned locally $P_{\mathrm{loc},80}(s_{\mathrm{glo}})$ compared to $P_{\mathrm{glo},80}(s_{\mathrm{glo}})$ resulting in an absolute drift in the energy demand between $E_{\mathrm{glo},80}(s_{\mathrm{glo}})$ and $E_{\mathrm{loc},80}(s_{\mathrm{glo}})$ of approx. 15.8 % within one race lap (bottom).

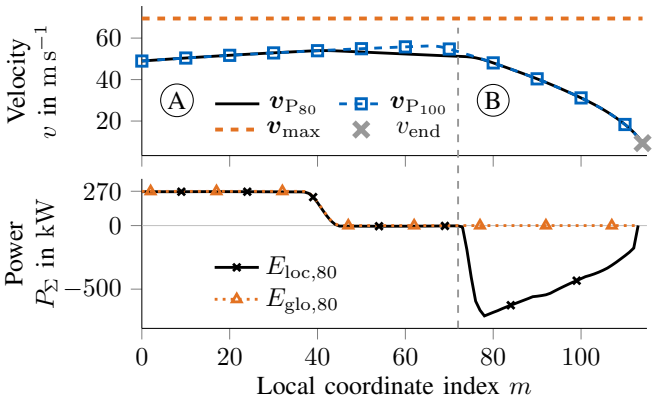


Fig. 6. The effect of the ES on one solution of the velocity-planning problem on the vehicle speed $\boldsymbol{v}_{\mathrm{P80}}$. In part (A), the race car is accelerating with the maximum available machine power of $P_{\Sigma} = 270\,\mathrm{kW}$ and is forced to coast ($P_{\mathrm{glo}}(s_m) = P_{\mathrm{loc}}(s_m) = 0\,\mathrm{kW}$) thereafter. Acceleration without any power restriction is denoted by $\boldsymbol{v}_{\mathrm{P100}}$. Part (B) of the planning horizon ensures recursive feasibility.

C. Variable acceleration limits

Fig. 7 shows a locally variable acceleration map to determine the values of $\bar{a}_{\mathrm{y}}(s)$ along the driven path. We conducted three experiments with a constant acceleration potential of $\bar{a}_{\mathrm{x}}$ = 12.5 m s^{-2} and varied $\bar{a}_{\mathrm{y}}$ in three different ways:

- $\bar{a}^{+}_{\mathrm{y,cst}}$ has a constant value of 12.5 m s^{-2} along the entire track, describing a high friction potential.
- $\bar{a}^{-}_{\mathrm{y,cst}}$ has a constant value of 6.5 m s^{-2} along the entire track, describing a low friction potential.
- $\bar{a}_{\mathrm{y,var}}(s)$ has a variable value in the range of 6.5 m s^{-2} to 12.5 m s^{-2} with the steepest gradients between s_{glo} = 1520 - 1550 m and s_{glo} = 1670 - 1700 m, as shown in Fig. 7.

The results of these experiments can be seen in Fig. 8. We

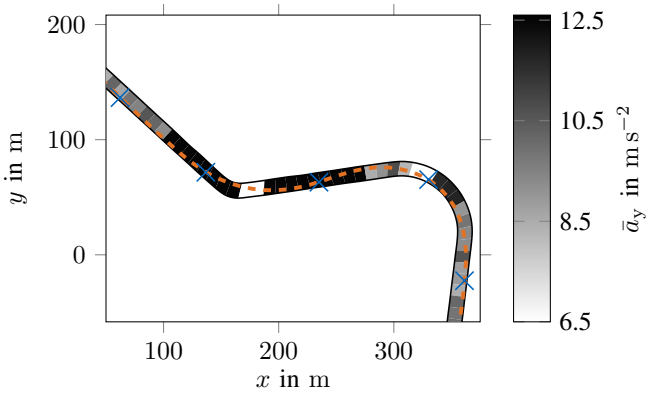


Fig. 7. Race track map displaying the spatially variable acceleration potential $\Sigma_{\bar{a}}(s_{\mathrm{glo}})$ including the driven path and markers starting from the global s-coordinate of 1400 m at 100 m gaps.

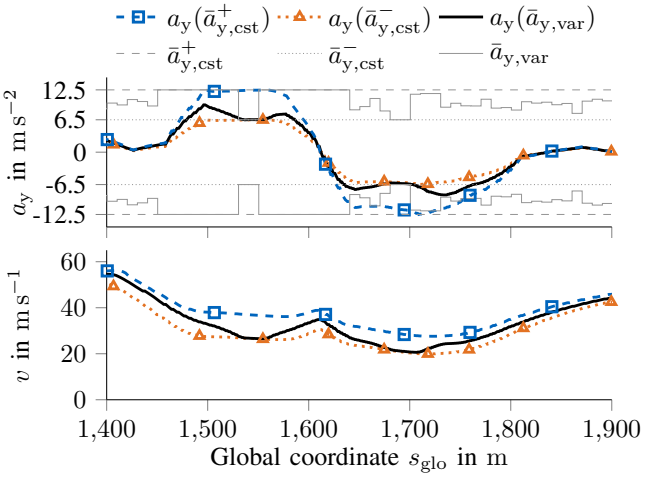


Fig. 8. Comparison of velocity $v(s_{\mathrm{glo}})$ and lateral acceleration $a_{\mathrm{y}}(s_{\mathrm{glo}})$ profiles resulting from spatially and temporally varying acceleration coefficients $\bar{a}_{\mathrm{y}}(s_{\mathrm{glo}})$.

depicted the acceleration potentials $\bar{a}_{\mathrm{y}}$ of the three scenarios in combination with the planned lateral acceleration $a_{\mathrm{y}}(s_{\mathrm{glo}})$ (top) and the vehicle velocity $v(s_{\mathrm{glo}})$ (bottom). In both scenarios with constant $\bar{a}_{\mathrm{y,cst}}$ values, the planner leverages the entire admissible lateral potentials which can be seen between s_{glo} = 1500 - 1600 m and s_{glo} = 1650 - 1700 m. The results achieved using the variable $\bar{a}_{\mathrm{y,var}}(s)$ are interesting: The planned lateral acceleration $a_{\mathrm{y}}(s)$ stayed within the boundaries of the low- and high-friction experiments. The planner handles the drop of $\bar{a}_{\mathrm{y,var}}(s)$ of 50 % at s_{glo} = 1520 m by reducing $a_{\mathrm{y}}(s)$ in advance, while still fully leveraging $\bar{a}_{\mathrm{y,var}}(s)$. This results in a vehicle velocity of $v(s_{\mathrm{glo}})$ in the low-friction scenario. The same is true for s_{glo} = 1670 m. Also during the rest of the shown experiment, the oscillating values of $\bar{a}_{\mathrm{y,var}}(s)$ can be handled by the mpSQP algorithm. This behavior allows us to fully exploit the dynamical limits of the race car on a track with variable acceleration potential.

A further indicator that the velocity planner utilizes the full acceleration potential is shown in Fig. 9. Here, we depicted the

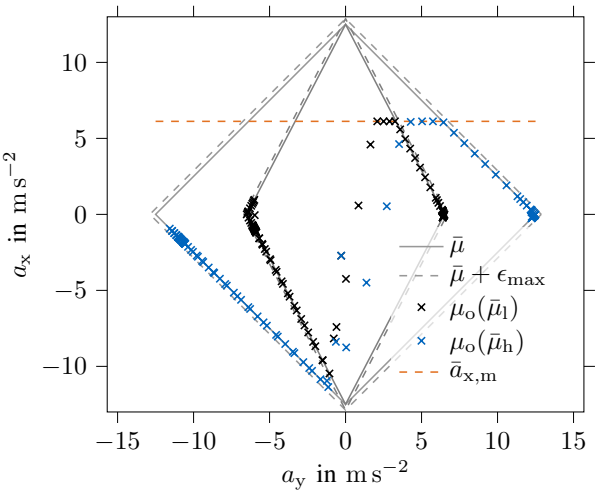


Fig. 9. Acceleration limit model including operating points at low- and high-friction experiments ($\bar{\mu}_\mathrm{l}$ and $\bar{\mu}_\mathrm{h}$) for a planning horizon ranging from the global coordinate of s_glo = 1500 m to 1800 m.

vehicle's optimized operating points $\mu_\mathrm{o}(s)$ regarding combined acceleration $a_\mathrm{x}(s)$ and $a_\mathrm{y}(s)$ within the planning horizon of s_glo = 1500 - 1800 m. Both solid diamond shapes express the given constant acceleration limits of the high-friction scenario, $\bar{\mu}_\mathrm{h} = \bar{a}_\mathrm{x}$ & $\bar{a}^{+}_\mathrm{y,cst}$ = 12.5 & 12.5 $\mathrm{m\,s^{-2}}$, and the low-friction scenario $\bar{\mu}_\mathrm{l} = \bar{a}_\mathrm{x}$ & $\bar{a}^{-}_\mathrm{y,cst}$ = 12.5 & 6.5 $\mathrm{m\,s^{-2}}$. The horizontal red dashed line indicates the maximum available electric machine acceleration $\bar{a}_\mathrm{x,m} = \frac{F_\mathrm{max}}{m_\mathrm{v}}$. From this diagram we see that the vehicle operates at the limits of the given acceleration constraints in both scenarios. This means the mpSQP leverages the maximum acceleration potential in combination with fully available cornering potential.

D. Solver comparison

The number of variables in the performance profile within one QP for the performance trajectory is 126, including 810 constraints. From the problem formulation (34), a small number of non-zero entries in the matrices $\boldsymbol{A}$ and $\boldsymbol{P}$ of 2295 in total arises with constant entries in the problem's Hessian $\boldsymbol{P}$.

Fig. 10 contains a comparison of the ADMM solver OSQP [11] and the active set solver qpOASES [35] as QP solvers for an mpSQP method as proposed in this paper. Furthermore, we solve the original NLP using the interior point solver IPOPT [28] interfaced via CasADi [36] to obtain a measure of solution quality for the proposed mpSQP. Note that IPOPT is widely used to benchmark the solution quality even if it is not specifically designed for embedded optimization. We chose a scenario where the vehicle is heading towards a narrow right-hand turn at a high velocity of almost 200 $\mathrm{km\,h^{-1}}$. The optimization horizon spans this turn including the consecutive straight where positive acceleration occurs. Therefore, the solvers have to deal with high gradients for the longitudinal force $F_\mathrm{x}(s_m)$ and lateral acceleration $a_\mathrm{y}(s_m)$ stemming from curve entry and exit. The velocity plot shows that the optimal solution $\boldsymbol{z}^*$ almost equals the initial guess $\boldsymbol{z}_\mathrm{ini}$ for both the

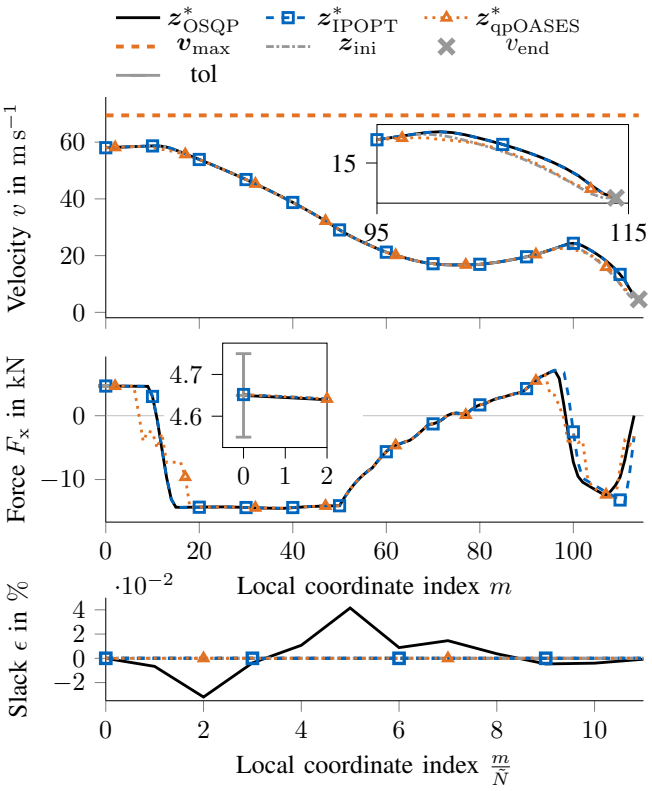


Fig. 10. Comparison of the solution of the mpSQP (internal QPs solved by the ADMM solver OSQP [11]) with the general NLP interior point solver IPOPT [28] interfaced by CasADi [36] and the active set solver qpOASES [35]. The chosen scenario includes the vehicle heading towards a narrow right-hand bend at a high velocity of almost 200 $\mathrm{km\,h^{-1}}$. The optimization horizon spans the curve, and includes the subsequent straight.

velocity vector $\boldsymbol{v}$ and the slack values $\boldsymbol{\epsilon}$. This behavior is expected, as the previous SQP solution $\boldsymbol{z}^{l-1}(s_m)$ is shifted by the traveled distance and used as initialization $\boldsymbol{z}_\mathrm{ini}$. The optimization outputs $\boldsymbol{z}^*_\mathrm{OSQP}$ and $\boldsymbol{z}^*_\mathrm{IPOPT}$ overlap except at the end of the planning horizon where IPOPT initially allows more positive longitudinal force $F_\mathrm{x}(s_m)$, resulting in more aggressive braking to fulfill the hard constraint v_end. qpOASES's solution oscillates in the force $F_\mathrm{x}(s_m)$ at the steep gradients. All the algorithms keep the initially given longitudinal force $F_\mathrm{x}(s_\mathrm{s})$ within the specified tolerance of ± 0.1 kN, with OSQP matching the exact value (see magnified section in the second plot). The slack values $\boldsymbol{\epsilon}$ are close to zero. However, the OSQP-solution shows small numerical oscillations within a negligible range of approx. $\pm 0.04\,\%$. Nevertheless this behavior is typical for an ADMM algorithm, and is thus noteworthy.

Apart from the solution qualities, we further analyzed the velocity optimization runtimes. The scenario consisted of two race laps, including a race start and coming to a standstill after the second lap on the Monteblanco (Spain) race circuit with a variable acceleration potential along the circuit, cf. Subsection IV-C. The histograms in Fig. 11 display the calculation times Δt_sol for the number of calls C to optimize a speed profile. C_P and C_E denote the calls for the performance and the emergency lines, respectively. Their mean values are given

TABLE II
SOLVER MEAN RUNTIMES.

CPU	Intel i7-7820HQ			A57 ARM
Prob. formulation	mpSQP	NLP	mpSQP	mpSQP
Solver	OSQP	IPOPT	qpOASES	OSQP
Performance in ms	6.20	40.4	164	32.4
Emergency in ms	6.95	29.8	26.1	34.2

by $\Delta\tilde{t}_{\mathrm{sol}}$. We wish to point out that the algorithm runtimes shown refer to an entire SQP optimization process for a speed profile consisting of the solution of several QPs in the case of the used solvers OSQP or qpOASES, which have been warm-started. The optimization runtimes on the specific CPUs are also summarized in Table II where the ARM A57 is the NVIDIA Drive PX2 CPU. As we selected OSQP for our application on the target hardware, we do not show additional solver times of IPOPT or qpOASES for the ARM A57 CPU.

Our mpSQP in combination with the QP solver OSQP reaches nearly equal mean runtimes of 6 ms to 7 ms for both velocity profiles on an Intel i7-7820HQ CPU and 32 ms to 34 ms on an A57 ARM CPU. An amount of 2 to 5 SQP iterations for the performance line was sufficient to reach the defined tolerances $\varepsilon_{\mathrm{SQP}}$. To optimize the emergency speed profile, a higher amount of SQP iterations in the range of 5 - 10 was necessary. Therefore, the computational effort for the iterative linearizations increased on this profile. In contrast, it was possible to solve the single QPs in less calculation time, as less than half the number of optimization variables $M+N$ in comparison to the performance profile are present. With maximum computation times of 16.9 ms (73.3 ms A57 ARM) on the performance profile and 15.0 ms (77.1 ms A57 ARM) for the emergency line, we managed to stay far below our predefined process-timeouts, see Table I.

The same optimization problem was formulated with the CasADi-language [36] as a general NLP and passed to the interior point solver IPOPT. The IPOPT mean solver runtimes are at least approximately five times higher, with maxima of around 0.1 s on the emergency line being too long for vehicle operations at velocities beyond 200 km h^{-1}. On the emergency profile, the active set solver qpOASES beats IPOPT slightly in terms of its mean runtime $\Delta\tilde{t}_{\mathrm{sol}}$ but consumes four times the IPOPT computation time to generate the performance speed profile, see Table II. This behavior is rational, as a higher number of optimization variables and active constraints increases the calculation speed of an active set solver significantly. Nevertheless, maximum computation times for the mpSQP of around 0.5 s on the performance line exclude the QP solver qpOASES for this type of application.

V. CONCLUSION

In this paper we presented a tailored mpSQP algorithm capable of adaptive velocity planning in real time for race cars operating at the limits of handling, and velocities above 200 km h^{-1}. The planner can deal with performance and emergency velocity profiles. Furthermore, the optimization handles multi-parametric input, e.g., from the friction estimation module or the race ES. We also specified the boundaries

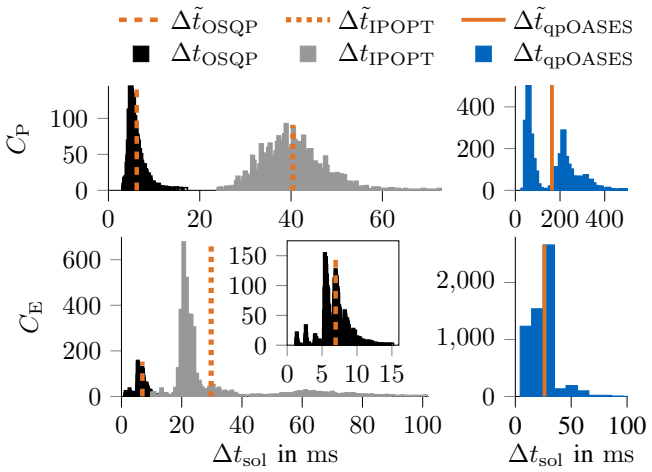


Fig. 11. Solver runtimes for the mpSQP with the local QPs solved by OSQP [11] or by the active set solver qpOASES [35], compared to the problem solved by the general NLP formulation passed to the interior point solver IPOPT interfaced by CasADi [36].

of maximum variation within these parameters to keep the problem feasible. Additionally, we compared different solvers applied to our problem formulation to compare calculation times as well as the solution qualities. Here, our mpSQP in combination with the ADMM solver OSQP outperformed the active set strategy qpOASES and the general NLP solver IPOPT in terms of calculation time, but reached nearly the same solution quality as IPOPT. This indicates that the first order ADMM in OSQP shows its strength for the minimum-time optimization problem as it handles the large set of active constraints well.

In future work we will apply the presented algorithm in autonomous races and implement a tailored trajectory optimization module based on the presented results and techniques for comparison.

CONTRIBUTIONS & ACKNOWLEDGMENTS

T. H. initiated the idea of the paper and contributed significantly to the concept, modeling, implementation and results. A. W. contributed to the design and feasibility analysis of the optimization problem. L. H. contributed essentially to the integration of variable acceleration limits. J. B. contributed to the whole concept of the paper. M. L. provided a significant contribution to the concept of the research project. He revised the paper critically for important intellectual content. M. L. gave final approval for the publication of this version and is in agreement with all aspects of the work. As a guarantor, he accepts responsibility for the overall integrity of this paper.

We would like to thank the Roborace team for giving us the opportunity to work with them and for the use of their vehicles for our research project. We would also like to thank the Bavarian Research Foundation (Bayerische Forschungsstiftung) for funding us in connection with the “rAIcing” research project. This work was also conducted with basic research fund of the Institute of Automotive Technology from the Technical University of Munich.

REFERENCES

[1] Chair of Automotive Technology, "Autonomous Driving Control Software of TUM Roborace," 2020. [Online]. Available: https://github.com/TUMFTM

[2] ——, "Python Package: Velocity Optimization," 2020. [Online]. Available: https://pypi.org/project/velocity-optimization/

[3] M. Morari, F. Allgöwer, L. Magni, D. M. Raimondo, and M. Thoma, *Nonlinear Model Predictive Control: Towards New Challenging Applications*, ser. Lecture Notes in Control and Information Sciences. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, vol. 384.

[4] T. Stahl, A. Wischnewski, J. Betz, and M. Lienkamp, "Multilayer Graph-Based Trajectory Planning for Race Vehicles in Dynamic Scenarios," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, pp. 3149–3154.

[5] A. Heilmeier, A. Wischnewski, L. Hermansdorfer, J. Betz, M. Lienkamp, and B. Lohmann, "Minimum curvature trajectory planning and control for an autonomous race car," *Vehicle System Dynamics*, vol. 25, no. 8, pp. 1–31, 2019.

[6] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, L. Hermansdorfer, T. Stahl, T. Herrmann, and M. Lienkamp, "A Software Architecture for the Dynamic Path Planning of an Autonomous Racecar at the Limits of Handling," in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*. IEEE, 2019, pp. 1–8.

[7] L. Hermansdorfer, J. Betz, and M. Lienkamp, "A Concept for Estimation and Prediction of the Tire-Road Friction Potential for an Autonomous Racecar," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, pp. 1490–1495.

[8] T. Herrmann, F. Passigato, J. Betz, and M. Lienkamp, "Minimum Race-Time Control-Strategy for an Autonomous Electric Racecar," in *2020 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2020 (In Press).

[9] E. Ozatay, U. Ozguner, and D. Filev, "Velocity profile optimization of on road vehicles: Pontryagin's Maximum Principle based approach," *Control Engineering Practice*, vol. 61, pp. 244–254, 2017.

[10] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2015.

[11] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 5, no. 1, p. 42, 2020.

[12] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, T. Stahl, L. Hermansdorfer, B. Lohmann, and M. Lienkamp, "What can we learn from autonomous level-5 motorsport?" in *9th International Munich Chassis Symposium 2018*, ser. Proceedings, P. Pfeffer, Ed. Wiesbaden: Springer Fachmedien Wiesbaden, 2019, pp. 123–146.

[13] S. Ebbesen, M. Salazar, P. Elbert, C. Bussi, and C. H. Onder, "Time-optimal Control Strategies for a Hybrid Electric Race Car," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 233–247, 2018.

[14] D. J. N. Limebeer and G. Perantoni, "Optimal Control of a Formula One Car on a Three-Dimensional Track—Part 2: Optimal Control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 5, p. 051019, 2015.

[15] A. J. Tremlett and D. J. N. Limebeer, "Optimal tyre usage for a Formula One car," *Vehicle System Dynamics*, vol. 54, no. 10, pp. 1448–1473, 2016.

[16] F. Christ, A. Wischnewski, A. Heilmeier, and B. Lohmann, "Time-optimal trajectory planning for a race car considering variable tyre-road friction coefficients," *Vehicle System Dynamics*, vol. 3, no. 1, pp. 1–25, 2019.

[17] N. Dal Bianco, R. Lot, and M. Gadola, "Minimum time optimal control simulation of a GP2 race car," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 232, no. 9, pp. 1180–1195, 2018.

[18] Y. Huang, H. Ding, Y. Zhang, H. Wang, D. Cao, N. Xu, and C. Hu, "A Motion Planning and Tracking Framework for Autonomous Vehicles Based on Artificial Potential Field Elaborated Resistance Network Approach," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 2, pp. 1376–1386, 2020.

[19] Y. Meng, Y. Wu, Q. Gu, and L. Liu, "A Decoupled Trajectory Planning Framework Based on the Integration of Lattice Searching and Convex Optimization," *IEEE Access*, vol. 7, pp. 130 530–130 551, 2019.

[20] Y. Zhang, H. Chen, S. L. Waslander, J. Gong, G. Xiong, T. Yang, and K. Liu, "Hybrid Trajectory Planning for Autonomous Driving in Highly Constrained Environments," *IEEE Access*, vol. 6, pp. 32 800–32 819, 2018.

[21] J. K. Subosits and J. C. Gerdes, "From the Racetrack to the Road: Real-Time Trajectory Replanning for Autonomous Driving," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 309–320, 2019.

[22] L. Svensson, M. Bujarbaruah, N. R. Kapania, and M. Torngren, "Adaptive Trajectory Planning and optimization at Limits of Handling," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 3942–3948.

[23] T. Mercy, R. van Parys, and G. Pipeleers, "Spline-Based Motion Planning for Autonomous Guided Vehicles in a Dynamic Environment," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 6, pp. 2182–2189, 2018.

[24] A. Carvalho, Y. Gao, A. Gray, E. Tseng, and F. Borrelli, "Predictive Control of an Autonomous Ground Vehicle using an Iterative Linearization Approach," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2013*. Piscataway, NJ: IEEE, 2013.

[25] B. Alrifaee and J. Maczijewski, "Real-time Trajectory Optimization for Autonomous Vehicle Racing using Sequential Linearization," in *2018 IEEE Intelligent Transportation Systems Conference*. Piscataway, NJ: IEEE, 2018, pp. 476–483.

[26] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1433–1440.

[27] E. Alcalá, V. Puig, J. Quevedo, and U. Rosolia, "Autonomous racing using Linear Parameter Varying-Model Predictive Control (LPV-MPC)," *Control Engineering Practice*, vol. 95, p. 104270, 2020.

[28] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[29] L. Svensson, L. Masson, N. Mohan, E. Ward, A. P. Brenden, L. Feng, and M. Torngren, "Safe Stop Trajectory Planning for Highly Automated Vehicles: An Optimal Control Problem Formulation," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 517–522.

[30] B. Houska, H. J. Ferreau, and M. Diehl, "ACADO toolkit-An open-source framework for automatic control and dynamic optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.

[31] T. Lipp and S. Boyd, "Minimum-time speed optimisation over a fixed path," *International Journal of Control*, vol. 87, no. 6, pp. 1297–1311, 2014.

[32] T. Denton, *Automated Driving and Driver Assistance Systems*, first edition ed. Abingdon, Oxon and New York, NY: Routledge, 2020.

[33] G. Banjac, P. Goulart, B. Stellato, and S. Boyd, "Infeasibility Detection in the Alternating Direction Method of Multipliers for Convex Optimization," *Journal of Optimization Theory and Applications*, vol. 183, no. 2, pp. 490–519, 2019.

[34] T. Herrmann, F. Christ, J. Betz, and M. Lienkamp, "Energy Management Strategy for an Autonomous Electric Racecar using Optimal Control," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, pp. 720–725.

[35] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: a parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[36] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[37] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, third edition ed., ser. International Series in Operations Research & Management Science. New York, NY and Berlin and Heidelberg: Springer, 2008, vol. 116.

[38] S. P. Boyd and L. Vandenberghe, *Convex optimization*, ser. Safari Tech Books Online. Cambridge, UK and New York: Cambridge University Press, 2004.

[39] P. T. Boggs and J. W. Tolle, "Sequential Quadratic Programming," *Acta Numerica*, vol. 4, pp. 1–51, 1995.

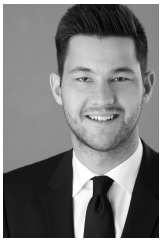
[40] J. Nocedal and S. J. Wright, *Numerical Optimization*, second edition ed., ser. Springer Series in Operations Research and Financial Engineering. New York, NY: Springer Science+Business Media LLC, 2006.

[41] F. Braghin, F. Cheli, S. Melzi, and E. Sabbioni, "Race driver model," *Computers & Structures*, vol. 86, no. 13-14, pp. 1503–1516, 2008.

[42] E. C. Kerrigan, "Soft Constraints and Exact Penalty Functions in Model Predictive Ccontrol," in *Proceedings of the United Kingdom Automatic Control Council 2000*, IFAC, Ed., 2000.

Thomas Herrmann was awarded a B.Sc. and an M.Sc. in Mechanical Engineering by the Technical University of Munich (TUM), Germany, in 2016 and 2018, respectively. He is currently pursuing his doctoral studies at the Institute of Automotive Technology at TUM where he is working as a Research Associate. His current research interests include optimal control in the field of trajectory planning for autonomous vehicles and the efficient incorporation of the electric powertrain behavior within these optimization problems.

Alexander Wischnewski was awarded a B.Eng. in Mechatronics by the DHBW Stuttgart in 2015, and an M.Sc. in Electrical Engineering and Information Technology by the University of Duisburg-Essen, Germany, in 2017. He is currently pursuing his doctoral studies at the Chair of Automatic Control at the Department of Mechanical Engineering at Technical University of Munich (TUM), Germany. His research interests lie at the intersection of control engineering and machine learning, with a strong focus on autonomous driving.



Leonhard Hermansdorfer was awarded a B.Sc. and an M.Sc. in Mechanical Engineering by the Technical University of Munich (TUM), Germany, in 2015 and 2018, respectively. He is currently pursuing his doctoral studies at the Institute of Automotive Technology at TUM where he is working as a Research Associate. His current research interest is the identification of a vehicle's maximum transmittable tire forces, with a strong focus on model-less approaches based on machine learning.

Johannes Betz is currently a postdoctoral researcher at the Technical University of Munich (TUM). He graduated in 2013 with an M.Sc. in Automotive Technology from the University of Bayreuth, Germany. He graduated with a doctoral degree from TUM about the topic of fleet disposition for electric vehicles. Since then, he continues his research in the field of autonomous driving. His research topics include the dynamic trajectory and behavioral planning for autonomous vehicles at the handling limits, as well as prediction of traffic participant behavior in non-deterministic environments.

Markus Lienkamp studied Mechanical Engineering at TU Darmstadt, Germany, and Cornell University, USA, and received a Ph.D. degree from TU Darmstadt in 1995. He was with Volkswagen as part of an International Trainee Program and took part in a joint venture between Ford and Volkswagen in Portugal. Returning to Germany, he led the Brake Testing Department, VW Commercial Vehicle Development Section, Wolfsburg. He later became the Head of the "Electronics and Vehicle" Research Department, Volkswagen AG's Group Research Division. His main priorities were advanced driver assistance systems and vehicle concepts for electromobility. He has been Head and Professor of the Institute of Automotive Technology, Technical University of Munich (TUM), Germany, since November 2009. He is conducting research in the area of electromobility, autonomous driving and mobility.

7 Summary and Discussion

This chapter summarizes the content and results of the thesis, discusses its strengths and limitations, and introduces directions for future research.

7.1 Summary

The objective of this thesis was to develop an online-capable ES, which operates a race vehicle at the performance limit to achieve a minimum race time, leveraging the available amount of energy as efficiently as possible.

We conducted offline studies in Chapter 4 to isolate the most important effects of the energetic and thermodynamic constraints on the time-optimal global race trajectories. The formulation of an NLP allowed to precisely model the driving dynamics and the thermodynamic powertrain behavior. In particular, the powertrain model comprises descriptions of the battery, VSIs, electric motors, and cooling fluids. In this process, the effects on the time-optimal velocity profiles and the corresponding race paths were jointly studied. Moreover, we were able to formulate and solve an MRTP for multiple subsequent race laps considering long-term effects of the SOC and the powertrain component temperatures.

To dynamically recompute the global ES during a race for optimization horizons spanning the remaining race distance, we introduced the ES Core Module (Chapter 5). To achieve real-time capability, the computations in the ES Core Module are split into the three steps of “ES Guess”, “ES Presolve”, and “ES Resolve”, leveraging an SQP method. A comparison of real-world measurement data with our powertrain models, which were mainly parametrized by datasheet numbers, validate the underlying ODE system.

As a final necessary element to implement the global ES on the road, we presented an optimization-based velocity planner in Chapter 6. It handles external information like the power limitations provided by the ES Core Module. The velocity optimization algorithm forms the local trajectory planner together with a graph-based path planner in our autonomous software stack. Therefore, it must meticulously meet real-time requirements to operate a vehicle at the handling limits. To do so, we implemented the velocity computation algorithm as an mpSQP. Since the solver selection is crucial, we thoroughly benchmarked three open-source packages for the velocity planning problem, which implement a different iterative strategy each. Thereby, the solver unifying the best compromise between solution accuracy and computation speed on the embedded vehicle ECU could be selected.

To summarize, in this thesis we presented a software architecture to realize an ES, which is seamlessly integrable into an existing autonomous driving software stack. This is achieved by the interconnection of two software loops, which operate at a higher and a lower update frequency, via the power limitation interface. The results demonstrated that the planned global ES

adheres to the technical constraints stemming from the electric powertrain while still achieving a minimum lap time compared to state-of-the-art and real-world-validated global trajectory planning algorithms. At the same time, the locally requested vehicle power never exceeds the admissible global limitations. Several runtime analyses prove the real-time capability of the proposed ES architecture in combination with the autonomous driving software stack. Still, the governing equations describing the powertrain predict an accurate system behavior compared to measurement data. The mathematical nonlinear formulation of the ES algorithms allows for future extensions by, e.g., additional physical constraints.

7.2 Discussion and Outlook

Algorithmic Setup

In the ES Core Module we selected an IP method to solve the underlying QPs with high accuracy. According to our solver comparison in [4], mathematical strategies like ADMM can provide a competitive alternative to IP methods regarding the compromise between the resulting solution accuracy and the computation time. According to the literature (Section 2.2), ADMMs handle large-scale optimization problems well. Therefore, they would also depict a noteworthy research direction when applied to the ES problem.

In the current implementation of the ES Core Module, the optimization problem does not comprise the driving path as a free variable. But we assume that the vehicle will mainly follow the ideal race line, which is generated by solving an MLTP for a single race lap. Future research should therefore deal with the following two topics: First, one could incorporate the driving path into the optimization problem, since energetic and thermodynamic constraints do affect the geometry of the optimal race path, even if the velocity profile has the major impact on the energy consumption (Chapter 4). Second, the OCP in the ES Core Module could be reformulated in a convex way. In our work [22], we have demonstrated the convex formulation of an OCP to compute maximum-distance race strategies for optimization horizons spanning a compulsory endurance race time. Here, we have included a maximum velocity profile to incorporate the limitations of the driving dynamics, and additional constraints on the permissible amount of energy consumption per stint. To formulate the OCP, the physical governing equations were replaced by convex approximations. By doing so, approximation errors in the model descriptions are introduced but the global optimality of the obtained solution can be guaranteed.

Opponent Vehicles

As stated in the introduction in Chapter 1, the ES architecture and its formulation aim to solve the ES problem from a self-perspective. Therefore, the possible influence of opponent vehicles on the energy consumption is neglected, i.e., we considered a traffic-free race track in the planning phase. In the case of interactions with opponent vehicles, they are considered as disturbances in the proposed ES architecture, which reacts by computationally fast reoptimizations. Through this, the current powertrain state, which might differ from the previous prediction of the ES, is taken into account when adapting the ES to the remaining race distance. However, when integrating the behavior of opponent vehicles already in the planning phase, benefits through slipstream effects, e.g., could actively be leveraged. Similarly, necessary amounts of energy and “positive temperature delta” for overtaking maneuvers could be reserved in advance.

Closed-loopd Study

The physical powertrain descriptions in the ES Core Module are based on models with static parameters. These imply difficulties reacting to long-term effects like a varying internal battery resistance, which can occur during the course of a race season. Furthermore, systematic modeling discrepancies, which can stem from parameters like heat transfer coefficients that are only determinable with uncertainty from vehicle telemetry measurement data, cannot be handled by static parameters. To show the influence of a systematic modeling discrepancy, we set up a closed-loop simulation environment with the proposed ES architecture running in parallel to the autonomous driving software stack [237] (Figure 3). The vehicle powertrain is simulated based on the ODE system presented in Chapter 5 but with $7\,\%$ higher component power losses, which are additionally added to the vehicle dynamics control errors compared to the planned trajectories. The results in Figure 7.1 compare the initially planned ES to the iteratively adapted, i.e., the currently active ES trajectories. In the first ES prediction, the motor temperature would have limited the vehicle performance the most. Since the power losses during the course of the race are higher compared to the initial prediction, the SOC drops significantly faster. Through permanently replanning the ES trajectories the vehicle power gets limited such that the motor temperature does not reach its permitted limit but decreases during the course of the last race laps. Moreover, the difference between the initial SOC prediction and the measured powertrain SOC stops increasing after half the total race distance, which stresses that the ES replannings are able to react to disturbances or parameter uncertainties and to adapt the trajectories accordingly.

The results show that the ES Core Module is able to adapt the optimal driving behavior since it takes the current powertrain state into account during replanning. Nevertheless, the vehicle SOC will reach $0\,\%$ slightly before the finish line as can be seen around the distance of approx. $28\,\mathrm{km}$ in Figure 7.1. Since the iterative adaptions of the ES are based on constant model parameters, the actual powertrain behavior permanently differs from the ES predictions in this experiment. This can also be seen when comparing the last available ES prediction and the low SOC region, ultimately leading to a prematurely empty battery. Therefore, parameter estimation algorithms should be included into the ES Core Module to retune the model parameters online. In combination with the high update frequency of the ES Core Module, dynamic model parameters can be handled and iteratively replanning the ES with retuned model parameters will lead to a high prediction accuracy.

In future work, the output of the ES Core Module and the simulated close-loop behavior of the proposed ES architecture should also be validated in a prototype vehicle.

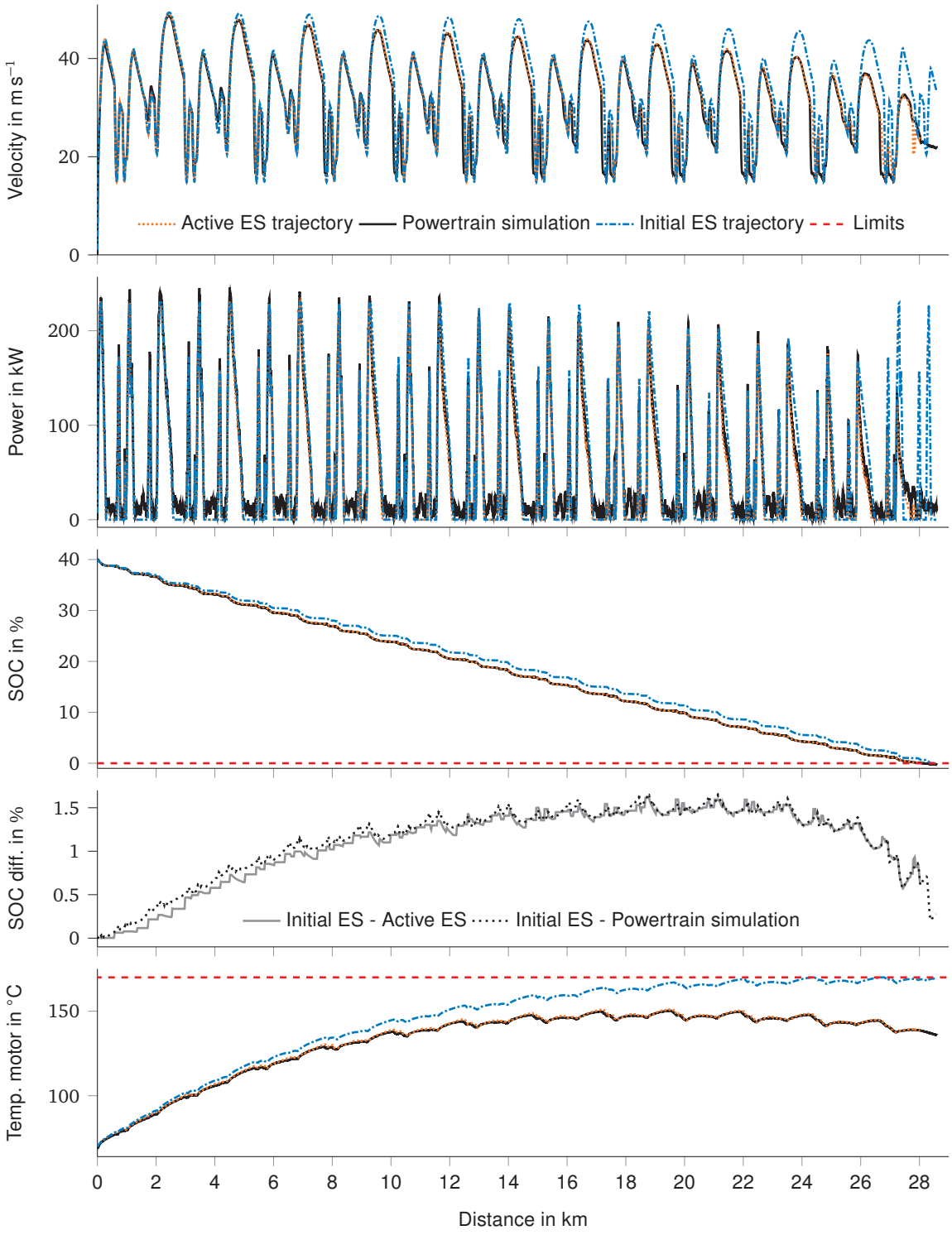


Figure 7.1: Closed-loop simulation of the proposed ES architecture in parallel to the autonomous driving software stack, and a powertrain simulation, which is based on [237, p. 29].

List of Figures

List of Tables

Bibliography

[1] Volkswagen AG. „*Annual Report*,“ 2017. Available: https://annualreport2017.volkswagenag.com/servicepages/filelibrary/files/collection.php [visited on 05/08/2021].

[2] T. Herrmann, F. Christ, J. Betz and M. Lienkamp, „Energy Management Strategy for an Autonomous Electric Racecar using Optimal Control,“ in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, 2019, pp. 720–725, ISBN: 978-1-5386-7024-8. DOI: 10.1109/ITSC.2019.8917154.

[3] T. Herrmann, F. Passigato, J. Betz and M. Lienkamp, „Minimum Race-Time Planning-Strategy for an Autonomous Electric Racecar,“ in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Rhodes, Greece, 2020, pp. 1–6, ISBN: 978-1-7281-4149-7. DOI: 10.1109/ITSC45102.2020.9294681.

[4] T. Herrmann, A. Wischnewski, L. Hermansdorfer, J. Betz and M. Lienkamp, „Real-Time Adaptive Velocity Optimization for Autonomous Electric Cars at the Limits of Handling,“ *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 4, pp. 665–677, 2021, DOI: 10.1109/TIV.2020.3047858.

[5] T. Herrmann. „Velocity Optimization: Manual,“ 2021. [Online]. Available: https://velocity-optimization.readthedocs.io/en/latest/introduction.html [visited on 10/07/2021].

[6] T. Herrmann. „Global static Energy Strategy,“ 2021. [Online]. Available: https://github.com/TUMFTM/global_racetrajectory_optimization/tree/master/opt_mintime_traj/powertrain_src [visited on 10/07/2021].

[7] T. Herrmann. „Velocity Optimization: Python package,“ 2021. [Online]. Available: https://pypi.org/project/velocity-optimization/ [visited on 10/07/2021].

[8] T. Herrmann. „Velocity Optimization: GitHub source,“ 2021. [Online]. Available: https://github.com/TUMFTM/velocity_optimization [visited on 10/07/2021].

[9] T. Herrmann, F. Sauerbeck, M. Bayerlein, J. Betz and M. Lienkamp, „Optimization-Based Real-Time-Capable Energy Strategy for Autonomous Electric Race Cars,“ *SAE International Journal of Connected and Automated Vehicles*, vol. 5, no. 1, pp. 45–59, 2022.

[10] T. Herrmann. „Global dynamic Energy Strategy,“ 2021. [Online]. Available: https://github.com/TUMFTM/embes [visited on 10/07/2021].

[11] T. Herrmann. „Global dynamic Energy Strategy: Manual,“ 2021. [Online]. Available: https://embes.readthedocs.io/en/latest/introduction.html [visited on 10/07/2021].

[12] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, L. Hermansdorfer, T. Stahl, T. Herrmann and M. Lienkamp, „A Software Architecture for the Dynamic Path Planning of an Autonomous Racecar at the Limits of Handling,“ in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, Graz, Austria, 2019, pp. 1–8, ISBN: 978-1-7281-0142-2. DOI: 10.1109/ICCVE45908.2019.8965238.

[13] S. Kalt, K. L. Stolle, P. Neuhaus, T. Herrmann, A. Koch and M. Lienkamp, „Dependency of Machine Efficiency on the Thermal Behavior of Induction Machines," *Machines*, vol. 8, no. 1, p. 9, 2020, DOI: 10.3390/machines8010009.

[14] A. Koch, T. Bürchner, T. Herrmann and M. Lienkamp, „Eco-Driving for Different Electric Powertrain Topologies Considering Motor Efficiency," *World Electric Vehicle Journal*, vol. 12, no. 1, p. 6, 2021, DOI: 10.3390/wevj12010006.

[15] A. Koch, L. Nicoletti, T. Herrmann and M. Lienkamp, „Eco-Driving Algorithm for Electric Powertrains using Detailed Loss Models," *eTransportation*, 2022 (Submitted).

[16] K. Minnerup, T. Herrmann, M. Steinstraeter and M. Lienkamp, „Concept for a Holistic Energy Management System for Battery Electric Vehicles Using Hybrid Genetic Algorithms," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, Chicago, IL, USA, 2018, pp. 1–6, ISBN: 978-1-5386-6358-5. DOI: 10.1109/VTCFall.2018.8690563.

[17] K. Minnerup, T. Herrmann, M. Steinstraeter and M. Lienkamp, „Case Study of Holistic Energy Management Using Genetic Algorithms in a Sliding Window Approach," *World Electric Vehicle Journal*, vol. 10, no. 2, p. 46, 2019, DOI: 10.3390/wevj10020046.

[18] N. Wassiliadis, T. Herrmann, L. Wildfeuer, C. Reiter and M. Lienkamp, „Comparative Study of State-Of-Charge Estimation with Recurrent Neural Networks," in *2019 IEEE Transportation Electrification Conference and Expo (ITEC)*, Detroit, MI, USA, 2019, pp. 1–6, ISBN: 978-1-5386-9310-0. DOI: 10.1109/ITEC.2019.8790597.

[19] A. Wischnewski, M. Geisslinger, J. Betz, T. Betz, F. Fent, A. Heilmeier, L. Hermansdorfer, T. Herrmann, S. Huch, P. Karle, F. Nobis, L. Ögretmen, M. Rowold, F. Sauerbeck, T. Stahl, R. Trauth and M. Lienkamp, „Indy Autonomous Challenge - Autonomous Race Cars at the Handling Limits," in *12th International Munich Chassis Symposium 2021: Chassis.tech plus* (Proceedings), P. Pfeffer, ed. Berlin: Springer Berlin and Springer Vieweg, 2022, ISBN: 978-3-662-64549-9.

[20] A. Wischnewski, T. Herrmann, F. Werner and B. Lohmann, „A Tube-MPC Approach to Autonomous Multi-Vehicle Racing on High-Speed Ovals," *IEEE Transactions on Intelligent Vehicles*, 2022 (In Press).

[21] J. v. Kampen, T. Herrmann and M. Salazar, „Maximum-distance Race Strategies for a Fully Electric Endurance Race Car," *European Journal of Control*, 2022 (Invited Contribution, Submitted).

[22] J. v. Kampen, T. Herrmann and M. Salazar, „Maximum-distance Race Strategies for a Fully Electric Endurance Race Car," in *2021 European Control Conference (ECC)*, London, 2022 (In Press).

[23] T. Herrmann and L. F. Peiss, „Verstärkendes Lernen," in *Wie Maschinen lernen*. vol. 1, K. Kersting, C. Lampert and C. Rothkopf, ed. Wiesbaden: Springer Fachmedien Wiesbaden, 2019, pp. 203–212, ISBN: 978-3-658-26762-9. DOI: 10.1007/978-3-658-26763-6_26.

[24] J. K. Subosits and J. C. Gerdes, „From the Racetrack to the Road: Real-Time Trajectory Replanning for Autonomous Driving," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 309–320, 2019, DOI: 10.1109/TIV.2019.2904390.

[25] Roborace Ltd. „Roborace," 2021. [Online]. Available: https://roborace.com/ [visited on 05/08/2021].

[26] M. Alawadhi, J. Almazrouie, M. Kamil and K. A. Khalil, „A systematic literature review of the factors influencing the adoption of autonomous driving," *International Journal of System Assurance Engineering and Management*, vol. 11, no. 6, pp. 1065–1082, 2020, DOI: 10.1007/s13198-020-00961-4.

[27] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*, Berlin, Heidelberg, Springer Science+Business Media, 2008, ISBN: 978-3-540-23957-4. DOI: 10.1007/978-3-540-30301-5. Available: http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10284823.

[28] D. J. Fagnant and K. Kockelman, „Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations," *Transportation Research Part A: Policy and Practice*, vol. 77, no. 1, pp. 167–181, 2015, DOI: 10.1016/j.tra.2015.04.003.

[29] M. Maurer, J. C. Gerdes, B. Lenz and H. Winner, *Autonomous Driving*, Berlin, Heidelberg, Springer Berlin Heidelberg, 2016, ISBN: 978-3-662-48845-4. DOI: 10.1007/978-3-662-48847-8.

[30] B. Reuse and R. Vollmar, *Informatikforschung in Deutschland*, Berlin, Heidelberg, Springer Berlin Heidelberg, 2008, ISBN: 978-3-540-76549-3. DOI: 10.1007/978-3-540-76550-9.

[31] Y. Zhang, Z. Ai, J. Chen, T. You, C. Du and L. Deng, „Energy-Saving Optimization and Control of Autonomous Electric Vehicles With Considering Multiconstraints," *IEEE transactions on cybernetics*, vol. PP, 2021, DOI: 10.1109/TCYB.2021.3069674.

[32] World Health Organization, „Global status report on road safety," 2018. Available: https://www.who.int/publications/i/item/9789241565684 [visited on 08/31/2021].

[33] D. Watzenig and M. Horn, *Automated Driving: Safer and More Effi cient Future Driving*, Cham, Springer International Publishing, 2017, ISBN: 978-3-319-31893-6. DOI: 10.1007/978-3-319-31895-0.

[34] Z. Chai, T. Nie and J. Becker, *Autonomous Driving Changes the Future*, Singapore, Springer Singapore, 2021, ISBN: 978-981-15-6727-8. DOI: 10.1007/978-981-15-6728-5.

[35] T. I. Fossen, K. Y. Pettersen and H. Nijmeijer, *Sensing and Control for Autonomous Vehicles*. vol. 474, Cham, Springer International Publishing, 2017, ISBN: 978-3-319-55371-9. DOI: 10.1007/978-3-319-55372-6.

[36] M. Luccarelli, D. T. Matt and P. R. Spena, „Modular architectures for future alternative vehicles," *International Journal of Vehicle Design*, vol. 67, no. 4, p. 368, 2015, DOI: 10.1504/ijvd.2015.070412.

[37] International Energy Agency, „Global EV Outlook 2021," 2021. Available: https://www.iea.org/reports/global-ev-outlook-2021 [visited on 09/01/2021].

[38] S. Habib, M. M. Khan, F. Abbas, L. Sang, M. U. Shahid and H. Tang, „A Comprehensive Study of Implemented International Standards, Technical Challenges, Impacts and Prospects for Electric Vehicles," *IEEE Access*, vol. 6, pp. 13866–13890, 2018, DOI: 10.1109/ACCESS.2018.2812303.

[39] G. Haddadian, M. Khodayar and M. Shahidehpour, „Accelerating the Global Adoption of Electric Vehicles: Barriers and Drivers," *The Electricity Journal*, vol. 28, no. 10, pp. 53–68, 2015, DOI: 10.1016/j.tej.2015.11.011.

[40] K. T. Ulrich, „Estimating the technology frontier for personal electric vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 5-6, pp. 448–462, 2005, DOI: 10.1016/j.trc.2006.01.002.

[41] S. Li, D. Han and B. Sarlioglu, „Analysis of the influence of temperature variation on performance of flux-switching permanent magnet machines for traction applications," in *2017 IEEE International Electric Machines and Drives Conference (IEMDC)*, Miami, FL, USA, 2017, pp. 1–6, ISBN: 978-1-5090-4281-4. DOI: 10.1109/IEMDC.2017.8002084.

[42] L. Lu, X. Han, J. Li, J. Hua and M. Ouyang, „A review on the key issues for lithium-ion battery management in electric vehicles," *Journal of Power Sources*, vol. 226, no. 3, pp. 272–288, 2013, DOI: 10.1016/j.jpowsour.2012.10.060.

[43] FIA. „Formula E: Cars + Technology," 2021. [Online]. Available: https://www.fiaformulae.com/en/discover/cars-and-technology [visited on 05/08/2021].

[44] Indy Autonomous Challenge. „Indy Autonomous Challenge," 2022. [Online]. Available: https://www.indyautonomouschallenge.com/ [visited on 01/27/2022].

[45] P. Pfeffer, *9th International Munich Chassis Symposium 2018*, (Proceedings), Wiesbaden, Springer Fachmedien Wiesbaden, 2019, ISBN: 978-3-658-22049-5. DOI: 10.1007/978-3-658-22050-1.

[46] W. Cai, X. Wu, M. Zhou, Y. Liang and Y. Wang, „Review and Development of Electric Motor Systems and Electric Powertrains for New Energy Vehicles," *Automotive Innovation*, vol. 4, no. 1, pp. 3–22, 2021, DOI: 10.1007/s42154-021-00139-z.

[47] A. Schönknecht, A. Babik and V. Rill, „Electric powertrain system design of BEV and HEV applying a multi objective optimization methodology," *Transportation Research Procedia*, vol. 14, pp. 3611–3620, 2016, DOI: 10.1016/j.trpro.2016.05.429.

[48] E. A. Grunditz, „Design and Assessment of Battery Electric Vehicle Powertrain, with Respect to Performance, Energy Consumption and Electric Motor Thermal Capability," PhD Thesis, Chalmers University of Technology, Sweden, 2016.

[49] K. Rajashekara, „Present Status and Future Trends in Electric Vehicle Propulsion Technologies," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 1, no. 1, pp. 3–10, 2013, DOI: 10.1109/JESTPE.2013.2259614.

[50] A. Sciarretta and A. Vahidi, *Energy-Efficient Driving of Road Vehicles: Toward Cooperative, Connected, and Automated Mobility*, (Lecture Notes in Intelligent Transportation and Infrastructure), 1st ed. 2020, Cham, Springer International Publishing and Springer, 2020, ISBN: 978-3-030-24127-8. DOI: 10.1007/978-3-030-24127-8.

[51] M. A. Hannan, M. Lipu, A. Hussain and A. Mohamed, „A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations," *World Electric Vehicle Journal*, vol. 78, no. 9, pp. 834–854, 2017, DOI: 10.1016/j.rser.2017.05.001.

[52] E. Raszmann, K. Baker, Y. Shi and D. Christensen, „Modeling Stationary Lithium-Ion Batteries for Optimization and Predictive Control," in *2017 IEEE Power and Energy Conference at Illinois (PECI)*, Champaign, IL, USA, 2017, pp. 1–7, ISBN: 978-1-5090-5550-0. DOI: 10.1109/PECI.2017.7935755.

[53] D. M. Rosewater, D. A. Copp, T. A. Nguyen, R. H. Byrne and S. Santoso, „Battery Energy Storage Models for Optimal Control," *IEEE Access*, vol. 7, pp. 178357–178391, 2019, DOI: 10.1109/ACCESS.2019.2957698.

[54] U. Krewer, F. Röder, E. Harinath, R. D. Braatz, B. Bedürftig and R. Findeisen, „Review—Dynamic Models of Li-Ion Batteries for Diagnosis and Operation: A Review and Perspective," *Journal of The Electrochemical Society*, vol. 165, no. 16, A3656–A3673, 2018, DOI: 10.1149/2.1061814jes.

[55] V. Boovaragavan, S. Harinipriya and V. R. Subramanian, „Towards real-time (milliseconds) parameter estimation of lithium-ion batteries using reformulated physics-based models," *Journal of Power Sources*, vol. 183, no. 1, pp. 361–365, 2008, DOI: 10.1016/j.jpowsour.2008.04.077.

[56] G. G. Botte, V. R. Subramanian and R. E. White, „Mathematical modeling of secondary lithium batteries," *Electrochimica Acta*, vol. 45, no. 15-16, pp. 2595–2609, 2000, DOI: 10.1016/S0013-4686(00)00340-6.

[57] M. Heinrich, N. Wolff, N. Harting, V. Laue, F. Röder, S. Seitz and U. Krewer, „Physico–Chemical Modeling of a Lithium–Ion Battery: An Ageing Study with Electrochemical Impedance Spectroscopy," *Batteries & Supercaps*, vol. 2, no. 6, pp. 530–540, 2019, DOI: 10.1002/batt.201900011.

[58] T. Bruen and J. Marco, „Modelling and experimental evaluation of parallel connected lithium ion cells for an electric vehicle battery system," *Journal of Power Sources*, vol. 310, pp. 91–101, 2016, DOI: 10.1016/j.jpowsour.2016.01.001.

[59] H. He, R. Xiong and J. Fan, „Evaluation of Lithium-Ion Battery Equivalent Circuit Models for State of Charge Estimation by an Experimental Approach," *Energies*, vol. 4, no. 4, pp. 582–598, 2011, DOI: 10.3390/en4040582.

[60] H. He, R. Xiong, H. Guo and S. Li, „Comparison study on the battery models used for the energy management of batteries in electric vehicles," *Energy Conversion and Management*, vol. 64, no. 12, pp. 113–121, 2012, DOI: 10.1016/j.enconman.2012.04.014.

[61] H. He, X. Zhang, R. Xiong, Y. Xu and H. Guo, „Online model-based estimation of state-of-charge and open-circuit voltage of lithium-ion batteries in electric vehicles," *Energy*, vol. 39, no. 1, pp. 310–318, 2012, DOI: 10.1016/j.energy.2012.01.009.

[62] X. Hu, S. Li and H. Peng, „A comparative study of equivalent circuit models for Li-ion batteries," *Journal of Power Sources*, vol. 198, pp. 359–367, 2012, DOI: 10.1016/j.jpowsour.2011.10.013.

[63] Z. Chen, Y. Fu and C. C. Mi, „State of Charge Estimation of Lithium-Ion Batteries in Electric Drive Vehicles Using Extended Kalman Filtering," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 3, pp. 1020–1030, 2013, DOI: 10.1109/TVT.2012.2235474.

[64] D. V. Do, C. Forgez, K. El Kadri Benkara and G. Friedrich, „Impedance Observer for a Li-Ion Battery Using Kalman Filter," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 8, pp. 3930–3937, 2009, DOI: 10.1109/TVT.2009.2028572.

[65] N. Wassiliadis, J. Adermann, A. Frericks, M. Pak, C. Reiter, B. Lohmann and M. Lienkamp, „Revisiting the dual extended Kalman filter for battery state-of-charge and state-of-health estimation: A use-case life cycle analysis," *Journal of Energy Storage*, vol. 19, no. 6, pp. 73–87, 2018, DOI: 10.1016/j.est.2018.07.006.

[66] M. Charkhgard and M. Farrokhi, „State-of-Charge Estimation for Lithium-Ion Batteries Using Neural Networks and EKF," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 12, pp. 4178–4187, 2010, DOI: 10.1109/TIE.2010.2043035.

[67] X. Chen, W. Shen, M. Dai, Z. Cao, J. Jin and A. Kapoor, „Robust Adaptive Sliding-Mode Observer Using RBF Neural Network for Lithium-Ion Battery State of Charge Estimation in Electric Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 1936–1947, 2016, DOI: 10.1109/TVT.2015.2427659.

[68] M. Tomasov, M. Kajanova, P. Bracinik and D. Motyka, „Overview of Battery Models for Sustainable Power and Transport Applications," *Transportation Research Procedia*, vol. 40, no. 1, pp. 548–555, 2019, DOI: 10.1016/j.trpro.2019.07.079.

[69] K. A. Smith, C. D. Rahn and C.-Y. Wang, „Control oriented 1D electrochemical model of lithium ion battery," *Energy Conversion and Management*, vol. 48, no. 9, pp. 2565–2578, 2007, DOI: 10.1016/j.enconman.2007.03.015.

[70] A. Seaman, T.-S. Dao and J. McPhee, „A survey of mathematics-based equivalent-circuit and electrochemical battery models for hybrid and electric vehicle simulation," *Journal of Power Sources*, vol. 256, no. 12, pp. 410–423, 2014, DOI: 10.1016/j.jpowsour.2014.01.057.

[71] S. M. Mousavi G. and M. Nikdel, „Various battery models for various simulation studies and applications," *World Electric Vehicle Journal*, vol. 32, no. 1, pp. 477–485, 2014, DOI: 10.1016/j.rser.2014.01.048.

[72] H. L. Chan and D. Sutanto, „A New Battery Model for use with Battery Energy Storage Systems and Electric Vehicles Power Systems," in *2000 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No.00CH37077)*, Singapore, 2000, pp. 470–475, ISBN: 0-7803-5935-6. DOI: 10.1109/PESW.2000.850009.

[73] B.-Y. Chang and S.-M. Park, „Electrochemical Impedance Spectroscopy," *Annual review of analytical chemistry (Palo Alto, Calif.)*, vol. 3, pp. 207–229, 2010, DOI: 10.1146/annurev.anchem.012809.102211.

[74] Y. Xing, W. He, M. Pecht and K. L. Tsui, „State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures," *Applied Energy*, vol. 113, pp. 106–115, 2014, DOI: 10.1016/j.apenergy.2013.07.008.

[75] R. Xiong, X. Gong, C. C. Mi and F. Sun, „A robust state-of-charge estimator for multiple types of lithium-ion batteries using adaptive extended Kalman filter," *Journal of Power Sources*, vol. 243, no. 10, pp. 805–816, 2013, DOI: 10.1016/j.jpowsour.2013.06.076.

[76] M. Dubarry and B. Y. Liaw, „Development of a universal modeling tool for rechargeable lithium batteries," *Journal of Power Sources*, vol. 174, no. 2, pp. 856–860, 2007, DOI: 10.1016/j.jpowsour.2007.06.157.

[77] H. He, R. Xiong, X. Zhang, F. Sun and J. Fan, „State-of-Charge Estimation of the Lithium-Ion Battery Using an Adaptive Extended Kalman Filter Based on an Improved Thevenin Model," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 4, pp. 1461–1469, 2011, DOI: 10.1109/TVT.2011.2132812.

[78] S. Jung and D. Kang, „Multi-dimensional modeling of large-scale lithium-ion batteries," *Journal of Power Sources*, vol. 248, no. 1, pp. 498–509, 2014, DOI: 10.1016/j.jpowsour.2013.09.103.

[79] S. Mohan, J. B. Siegel, A. G. Stefanopoulou and R. Vasudevan, „An Energy-Optimal Warm-Up Strategy for Li-Ion Batteries and Its Approximations," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 3, pp. 1165–1180, 2019, DOI: 10.1109/TCST.2017.2785833.

[80] N. Kularatna and K. Gunawardane, „Rechargeable battery technologies: An electronic circuit designer's viewpoint," in *Energy Storage Devices for Renewable Energy-Based Systems*. vol. 44 Elsevier, 2021, pp. 65–98, ISBN: 9780128207789. DOI: 10.1016/B978-0-12-820778-9.00001-2.

[81] F. Blaabjerg, U. Jaeger and S. Munk-Nielsen, „Power Losses in PWM-VSI Inverter Using NPT or PT IGBT Devices," *IEEE Transactions on Power Electronics*, vol. 10, no. 3, pp. 358–367, 1995, DOI: 10.1109/63.388002.

[82] N. Borchardt, R. Kasper and W. Heinemann, „Design of a Wheel-hub Motor with Air Gap Winding and Simultaneous Utilization of all Magnetic Poles," in *2012 IEEE International Electric Vehicle Conference*, Greenville, SC, USA, 2012, pp. 1–7, ISBN: 978-1-4673-1561-6. DOI: 10.1109/IEVC.2012.6183222.

[83] T. J. Donnelly, S. D. Pekarek, D. R. Fudge and N. Zarate, „Thévenin Equivalent Circuits for Modeling Common-Mode Behavior in Power Electronic Systems," *IEEE Open Access Journal of Power and Energy*, vol. 7, pp. 163–172, 2020, DOI: 10.1109/OAJPE.2020.2996029.

[84] A. Emadi, *Handbook of Automotive Power Electronics and Motor Drives*, (Electrical and computer engineering). vol. 125, Boca Raton, Taylor & Francis, 2005, ISBN: 9781420028157.

[85] S.-H. Kim, „Pulse width modulation inverters," in *Electric Motor Control*. vol. IA-17 Elsevier, 2017, pp. 265–340, ISBN: 9780128121382. DOI: 10.1016/B978-0-12-812138-2.00007-6.

[86] C.-W. Hung, „An ultra-high-speed motor driver with hybrid modulations," *Artificial Life and Robotics*, vol. 18, no. 1-2, pp. 70–75, 2013, DOI: 10.1007/s10015-013-0101-5.

[87] W. Mazgaj, B. Rozegnał and Z. Szular, „Switching losses in three-phase voltage source inverters," *Technical Transactions*, vol. 2015, no. 2-E, pp. 47–60, 2015, DOI: 10.4467/2353737XCT.15.087.3919.

[88] S. N. Manias, „Inverters (DC–AC Converters)," in *Power Electronics and Motor Drive Systems*. vol. 32 Elsevier, 2017, pp. 271–500, ISBN: 9780128117989. DOI: 10.1016/B978-0-12-811798-9.00006-8.

[89] M. Aguirre, P. Madina, J. Poza, A. Aranburu and T. Nieva, „Analysis and Comparison of PWM Modulation Methods in VSI-Fed PMSM Drive Systems," in *2012 International Conference on Electrical Machines*, Marseille, France, 2012, pp. 851–857, ISBN: 978-1-4673-0142-8. DOI: 10.1109/ICElMach.2012.6349976.

[90] B. J. Baliga, *Fundamentals of Power Semiconductor Devices*, 2nd ed. 2019, Cham, Springer International Publishing, 2019, ISBN: 978-3-319-93987-2. DOI: 10.1007/978-3-319-93988-9. Available: http://dx.doi.org/10.1007/978-3-319-93988-9.

[91] R. W. Erickson and D. Maksimovic, *Fundamentals of Power Electronics*, Third edition, Cham, Springer, 2020, ISBN: 978-3-030-43879-1.

[92] W. Xie, X. Wang, F. Wang, W. Xu, R. Kennel and D. Gerling, „Dynamic Loss Minimization of Finite Control Set-Model Predictive Torque Control for Electric Drive System," *IEEE Transactions on Power Electronics*, vol. 31, no. 1, pp. 849–860, 2016, DOI: 10.1109/TPEL.2015.2410427.

[93] D. Graovac, M. Puerschel and A. Kiep. „MOSFET Power Losses Calculation Using the Data- Sheet Parameters: Application Note, V1.1," 2022. [Online]. Available: https://application-notes.digchip.com/070/70-41484.pdf [visited on 01/27/2022].

[94] O. Onederra, I. Kortabarria, I. M. de Alegria, J. Andreu and J. I. Garate, „Three-Phase VSI Optimal Switching Loss Reduction Using Variable Switching Frequency," *IEEE Transactions on Power Electronics*, vol. 32, no. 8, pp. 6570–6576, 2017, DOI: 10.1109/TPEL.2016.2616583.

[95] A. D. Rajapakse, A. M. Gole and P. L. Wilson, „Approximate Loss Formulae for Estimation of IGBT Switching Losses through EMTP-type Simulations," in *Proceedings of the International Conference on Power Systems Transients (IPST)*, Canada, 2005, pp. 1–6.

[96] M. H. Bierhoff and F. W. Fuchs, „Semiconductor Losses in Voltage Source and Current Source IGBT Converters Based on Analytical Derivation," in *2004 IEEE 35th Annual Power Electronics Specialists Conference (IEEE Cat. No.04CH37551)*, Aachen, Germany, 2004, pp. 2836–2842, ISBN: 0-7803-8399-0. DOI: 10.1109/PESC.2004.1355283.

[97] R. Molina Llorente, *Practical Control of Electric Machines: Model-based design and simulation*, (Advances in industrial control), Cham, Springer, 2020, ISBN: 978-3-030-34757-4.

[98] M. Popescu, I. Foley, D. A. Staton and J. E. Goss, „Multi-physics Analysis of a High Torque Density Motor for Electric Racing Cars," in *2015 IEEE Energy Conversion Congress and Exposition (ECCE)*, Montreal, QC, Canada, 2015, pp. 6537–6544, ISBN: 978-1-4673-7151-3. DOI: 10.1109/ECCE.2015.7310575.

[99] A. EL-Refaie and M. Osama, „High Specific Power Electrical Machines: A System Perspective," in *2017 20th International Conference on Electrical Machines and Systems (ICEMS)*, Sydney, Australia, 2017, pp. 1–6, ISBN: 978-1-5386-3246-8. DOI: 10.1109/ICEMS.2017.8055931.

[100] N. P. Quang and J.-A. Dittrich, *Vector Control of Three-Phase AC Machines: System Development in the Practice*, (Power systems), 2nd ed., Heidelberg, Springer, 2015, ISBN: 978-3-662-46914-9. DOI: 10.1007/978-3-662-46915-6. Available: http://dx.doi.org/10.1007/978-3-662-46915-6.

[101] K. Reif, K.-E. Noreikat and K. Borgeest, *Kraftfahrzeug-Hybridantriebe: Grundlagen, Komponenten, Systeme, Anwendungen*, (ATZ / MTZ-Fachbuch), Wiesbaden, Vieweg+Teubner Verlag, 2012, ISBN: 9783834807229. DOI: 10.1007/978-3-8348-2050-1. Available: http://dx.doi.org/10.1007/978-3-8348-2050-1.

[102] E. Clarke, *Circuit analysis of A-C power systems*, London, New York, J. Wiley & sons, inc., 1943. Available: https://www.worldcat.org/title/circuit-analysis-of-a-c-power-systems/oclc/1563693 [visited on 08/22/2021].

[103] R. H. Park, „Two Reaction Theory of Synchronous Machines Generalized Method of Analysis-Part I," *Transactions of the American Institute of Electrical Engineers*, vol. 48, no. 3, pp. 716–727, 1929, DOI: 10.1109/T-AIEE.1929.5055275.

[104] N. Mohan and S. Raju, *Analysis and Control of Electric Drives: Simulations and laboratory implementation*, Hoboken, NJ, John Wiley & Sons, Inc, 2021, ISBN: 9781119584537.

[105] R. S. Colby and D. W. Novotny, „Efficient Operation of Surface-Mounted PM Synchronous Motors," *IEEE Transactions on Industry Applications*, vol. IA-23, no. 6, pp. 1048–1054, 1987, DOI: 10.1109/TIA.1987.4505028.

[106] S. Morimoto, „Trend of Permanent Magnet Synchronous Machines," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 2, no. 2, pp. 101–108, 2007, DOI: 10.1002/tee.20116.

[107] H. D. Perassi, „Feldorientierte Regelung der permanenterregten Synchronmaschine ohne Lagegeber für den gesamten Drehzahlbereich bis zum Stillstand," PhD thesis, TU Ilmenau, Ilmenau, 2006.

[108] C. Mi, „Modelling of Iron Losses of Permanent Magnet Synchronous Motors," PhD thesis, University of Toronto, Toronto, 2001.

[109] A. McDonald and J. Carroll, „Design of generators for offshore wind turbines," in *Offshore Wind Farms*. vol. 30 Elsevier, 2016, pp. 159–192, ISBN: 9780081007792. DOI: 10.1016/B978-0-08-100779-2.00008-8.

[110] D. Joo, J.-H. Cho, K. Woo, B.-T. Kim and D.-K. Kim, „Electromagnetic Field and Thermal Linked Analysis of Interior Permanent-Magnet Synchronous Motor for Agricultural Electric Vehicle," *IEEE Transactions on Magnetics*, vol. 47, no. 10, pp. 4242–4245, 2011, DOI: 10.1109/TMAG.2011.2149504.

[111] W. Leonhard, *Control of Electrical Drives*, 2nd Completely Revised and Enlarged Edition, Berlin, Heidelberg, Springer Berlin Heidelberg, 1996, ISBN: 978-3-642-97648-3. DOI: 10.1007/978-3-642-97646-9. Available: http://dx.doi.org/10.1007/978-3-642-97646-9.

[112] C. Rollbuehler, S. Peukert, D. Fritz, J.-F. Heyd, J. Kolb and M. Doppelbauer, „Investigations on the Experimental Identification of AC-Copper Losses in Permanent Magnet Synchronous Machines using a Motor Sub-Assembly," in *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, Lisbon, Portugal, 2019, pp. 1150–1156, ISBN: 978-1-7281-4878-6. DOI: 10.1109/IECON.2019.8926835.

[113] S. Morimoto, Y. Tong, Y. Takeda and T. Hirasa, „Loss Minimization Control of Permanent Magnet Synchronous Motor Drives," *IEEE Transactions on Industrial Electronics*, vol. 41, no. 5, pp. 511–517, 1994, DOI: 10.1109/41.315269.

[114] N. Chiodetto, N. Bianchi and L. Alberti, „Improved Analytical Estimation of Rotor Losses in High-Speed Surface-Mounted PM Synchronous Machines," *IEEE Transactions on Industry Applications*, vol. 53, no. 4, pp. 3548–3556, 2017, DOI: 10.1109/TIA.2017.2693178.

[115] D. Ishak, Z. Q. Zhu and D. Howe, „Eddy-Current Loss in the Rotor Magnets of Permanent-Magnet Brushless Machines Having a Fractional Number of Slots Per Pole," *IEEE Transactions on Magnetics*, vol. 41, no. 9, pp. 2462–2469, 2005, DOI: 10.1109/TMAG.2005.854337.

[116] D. Zhang, A. Ebrahimi, C. Wohlers, J. Redlich and B. Ponick, „On the analytical calculation of eddy-current losses in permanent magnets of electrical machines," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, Singapore, Singapore, 2020, pp. 1052–1056, ISBN: 978-1-7281-5414-5. DOI: 10.1109/IECON43393.2020.9255397.

[117] H. Ohguchi, K. Kawakami, T. Hirayama, S. Imamori and K. Yamazaki, „Analysis of In-plane Eddy-current Loss in Electrical Steel Sheets of 6-pole 36-slot Permanent-magnet Synchronous Machines," in *2020 23rd International Conference on Electrical Machines and Systems (ICEMS)*, Hamamatsu, Japan, 2020, pp. 1913–1917, ISBN: 978-4-8868-6419-2. DOI: 10.23919/ICEMS50442.2020.9291183.

[118] R. Monajemy and R. Krishnan, „Control and Dynamics of Constant Power Loss Based Operation of Permanent Magnet Synchronous Motor Drive System," in *IECON'99. Conference Proceedings. 25th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.99CH37029)*, San Jose, CA, USA, 1999, pp. 1452–1457, ISBN: 0-7803-5735-3. DOI: 10.1109/IECON.1999.819438.

[119] N. Boubaker, D. Matt, P. Enrici, F. Nierlich and G. Durand, „Measurements of Iron Loss in PMSM Stator Cores Based on CoFe and SiFe Lamination Sheets and Stemmed From Different Manufacturing Processes," *IEEE Transactions on Magnetics*, vol. 55, no. 1, pp. 1–9, 2019, DOI: 10.1109/TMAG.2018.2877995.

[120] A. Mahmoudi, W. L. Soong, G. Pellegrino and E. Armando, „Loss Function Modeling of Efficiency Maps of Electrical Machines," *IEEE Transactions on Industry Applications*, vol. 53, no. 5, pp. 4221–4231, 2017, DOI: 10.1109/TIA.2017.2695443.

[121] N. Patin, „Passive Components – Technologies and Dimensioning," in *Power Electronics Applied to Industrial Systems and Transports, Volume 1* Elsevier, 2015, pp. 75–115, ISBN: 9781785480003. DOI: 10.1016/B978-1-78548-000-3.50004-5.

[122] V. Ruuskanen, J. Nerg, M. Rilla and J. Pyrhonen, „Iron Loss Analysis of the Permanent-Magnet Synchronous Machine Based on Finite-Element Analysis Over the Electrical Vehicle Drive Cycle," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 7, pp. 4129–4136, 2016, DOI: 10.1109/TIE.2016.2549005.

[123] Y. Yang, B. Bilgin, M. Kasprzak, S. Nalakath, H. Sadek, M. Preindl, J. Cotton, N. Schofield and A. Emadi, „Thermal management of electric machines," *IET Electrical Systems in Transportation*, vol. 7, no. 2, pp. 104–116, 2017, DOI: 10.1049/iet-est.2015.0050.

[124] A. M. EL-Refaie, N. C. Harris, T. M. Jahns and K. M. Rahman, „Thermal Analysis of Multibarrier Interior PM Synchronous Machine Using Lumped Parameter Model," *IEEE Transactions on Energy Conversion*, vol. 19, no. 2, pp. 303–309, 2004, DOI: 10.1109/TEC.2004.827011.

[125] J. Fan, C. Zhang, Z. Wang, Y. Dong, C. E. Nino, A. R. Tariq and E. G. Strangas, „Thermal Analysis of Permanent Magnet Motor for the Electric Vehicle Application Considering Driving Duty Cycle," *IEEE Transactions on Magnetics*, vol. 46, no. 6, pp. 2493–2496, 2010, DOI: 10.1109/TMAG.2010.2042043.

[126] Y. Zhang, S. McLoone, W. Cao, F. Qiu and C. Gerada, „Power Loss and Thermal Analysis of a MW High-Speed Permanent Magnet Synchronous Machine," *IEEE Transactions on Energy Conversion*, vol. 32, no. 4, pp. 1468–1478, 2017, DOI: 10.1109/TEC.2017.2710159.

[127] A. Locatello, M. Konda, O. Borsboom, T. Hofmann and M. Salazar, „Time-optimal Control of Electric Race Cars under Thermal Constraints," in *2021 European Control Conference (ECC)*, Rotterdam, 2021.

[128] X.-S. Yang, *Engineering Optimization: An introduction with metaheuristic applications*, Hoboken, NJ, Wiley, 2010, ISBN: 978-0-470-58246-6. DOI: 10.1002/9780470640425. Available: http://site.ebrary.com/lib/academiccompletetitles/home.action.

[129] S. P. Boyd and L. Vandenberghe, *Convex Optimization*, (Safari Tech Books Online), Cambridge, UK and New York, Cambridge University Press, 2004, ISBN: 9780521833783.

[130] D. E. Kvasov and M. S. Mukhametzhanov, „Metaheuristic vs. deterministic global optimization algorithms: The univariate case," *Applied Mathematics and Computation*, vol. 318, no. 9, pp. 245–259, 2018, DOI: 10.1016/j.amc.2017.05.014.

[131] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, 2nd ed., Chichester, England and Hoboken, NJ, John Wiley & Sons, 2007, ISBN: 978-0-470-03561-0. DOI: 10.1002/9780470512517. Available: http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10297668.

[132] M.-H. Lin, J.-F. Tsai and C.-S. Yu, „A Review of Deterministic Optimization Methods in Engineering and Management," *Mathematical Problems in Engineering*, vol. 2012, no. 5, pp. 1–15, 2012, DOI: 10.1155/2012/756023.

[133] G. Venter, „Review of Optimization Techniques," in *Encyclopedia of Aerospace Engineering*. vol. 156, R. Blockley and W. Shyy, ed. Chichester, UK: John Wiley & Sons, Ltd, 2010, p. 215, ISBN: 9780470754405. DOI: 10.1002/9780470686652.eae495.

[134] R. Sioshansi and A. J. Conejo, *Optimization in Engineering: Models and Algorithms*, (Springer optimization and its applications). vol. volume 120, Cham, Springer, 2017, ISBN: 9783319567679.

[135] I. L. Lopez Cruz, L. G. van Willigenburg and G. van Straten, „Optimal Control of Nitrate in Lettuce by Gradient and Differential Evolution Algorithms," *IFAC Proceedings Volumes*, vol. 34, no. 26, pp. 119–124, 2001, DOI: 10.1016/S1474-6670(17)33643-1.

[136] A. Chehouri, R. Younes, J. Perron and A. Ilinca, „A Constraint-Handling Technique for Genetic Algorithms using a Violation Factor," *Journal of Computer Science*, vol. 12, no. 7, pp. 350–362, 2016, DOI: 10.3844/jcssp.2016.350.362.

[137] D. J. Reid, „Genetic Algorithms in Constrained Optimization," *Mathematical and Computer Modelling*, vol. 23, no. 5, pp. 87–111, 1996, DOI: 10.1016/0895-7177(96)00014-3.

[138] J. B. Rawlings, D. Q. Mayne and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed., paperback ed., Santa Barbara (California), Nob Hill Publishing, 2020, ISBN: 9780975937754.

[139] J. Nocedal and S. J. Wright, *Numerical Optimization*, (Springer Series in Operations Research and Financial Engineering), Second Edition, New York, NY, Springer Science+Business Media LLC, 2006, ISBN: 978-0387-30303-1.

[140] D. P. Bertsekas, *Nonlinear Programming*, Third edition, Belmont, Massachusetts, Athena Scientific, 2016, ISBN: 978-1-886529-05-2.

[141] S. A. Piyavskii, „An algorithm for finding the absolute extremum of a function," *USSR Computational Mathematics and Mathematical Physics*, vol. 12, no. 4, pp. 57–67, 1972, DOI: 10.1016/0041-5553(72)90115-2.

[142] B. O. Shubert, „A Sequential Method Seeking the Global Maximum of a Function," *SIAM Journal on Numerical Analysis*, vol. 9, no. 3, pp. 379–388, 1972, DOI: 10.1137/0709036.

[143] M. J. Kochenderfer and T. A. Wheeler, *Algorithms for Optimization*, Cambridge Mass. and London, The MIT Press, 2019, ISBN: 9780262039420.

[144] D. R. Jones, C. D. Perttunen and B. E. Stuckman, „Lipschitzian Optimization Without the Lipschitz Constant," *Journal of Optimization Theory and Applications*, vol. 79, no. 1, pp. 157–181, 1993, DOI: 10.1007/BF00941892.

[145] D. R. Jones and J. R. R. A. Martins, „The DIRECT algorithm: 25 years Later," *Journal of Global Optimization*, vol. 79, no. 3, pp. 521–566, 2021, DOI: 10.1007/s10898-020-00952-6.

[146] P. de Fermat, „Methodus as Disquirendam Maximam et Minimam," in *Varia opera mathematica D. Petri de Fermat, senatoris tolosani*, J. Pech, ed. 1679, pp. 63–73.

[147] M. Artin et al., *Convex Analysis and Minimization Algorithms I*. vol. 305, Berlin, Heidelberg, Springer Berlin Heidelberg, 1993, ISBN: 978-3-642-08161-3. DOI: 10.1007/978-3-662-02796-7.

[148] R. Verschueren, „Convex approximation methods for nonlinear model predictive control," PhD thesis, Systems Control and Optimization Laboratory, University of Freiburg, Freiburg, 2018.

[149] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings and M. Diehl, „CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019, DOI: 10.1007/s12532-018-0139-4.

[150] R. Verschueren, G. Frison, D. Kouzoupis, N. van Duijkeren, A. Zanelli, R. Quirynen and M. Diehl, „Towards a modular software package for embedded optimization," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 374–380, 2018, DOI: 10.1016/j.ifacol.2018.11.062.

[151] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen and M. Diehl, „acados—a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, vol. 20, no. 3, p. 1655, 2021, DOI: 10.1007/s12532-021-00208-8.

[152] A. Wächter and L. T. Biegler, „On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006, DOI: 10.1007/s10107-004-0559-y.

[153] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, (International Series in Operations Research & Management Science). vol. 116, Third Edition, New York, NY, Berlin, and Heidelberg, Springer, 2008, ISBN: 978-0-387-74502-2.

[154] D. P. Bertsekas, *Convex Optimization Algorithms*, (Optimization and computation series). vol. 4, Belmont, Massachusetts, Athena Scientific, 2015, ISBN: 1-886529-28-0.

[155] P. T. Boggs and J. W. Tolle, „Sequential Quadratic Programming," *Acta Numerica*, vol. 4, pp. 1–51, 1995, DOI: 10.1017/S0962492900002518.

[156] B. Stellato, G. Banjac, P. Goulart, A. Bemporad and S. Boyd, „OSQP: An Operator Splitting Solver for Quadratic Programs," *Mathematical Programming Computation*, vol. 5, no. 1, p. 42, 2020, DOI: 10.1007/s12532-020-00179-2.

[157] G. Frison and M. Diehl, „HPIPM: a high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020, DOI: 10.1016/J.IFACOL.2020.12.073.

[158] N. I. M. Gould and P. L. Toint, „Numerical Methods for Large-Scale Non-Convex Quadratic Programming," in *Trends in Industrial and Applied Mathematics* (Applied Optimization). vol. 72, P. M. Pardalos, D. Hearn, A. H. Siddiqi and M. Kočvara, ed. Boston, MA: Springer US, 2002, pp. 149–179, ISBN: 978-1-4613-7967-6. DOI: 10.1007/978-1-4613-0263-6_8.

[159] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock and M. Diehl, „qpOASES: a parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014, DOI: 10.1007/s12532-014-0071-1.

[160] C. Kirches, S. Sager, H. G. Bock and J. P. Schlöder, „Time-optimal control of automobile test drives with gear shifts," *Optimal Control Applications and Methods*, vol. 31, no. 2, pp. 137–153, 2010, DOI: 10.1002/oca.892.

[161] M. Diehl, H. G. Bock, H. Diedam and P.-B. Wieber, „Fast Direct Multiple Shooting Algorithms for Optimal Robot Control," in *Fast Motions in Biomechanics and Robotics* (Lecture Notes in Control and Information Sciences). vol. 340, M. Diehl and K. Mombaur, ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 65–93, ISBN: 978-3-540-36118-3. DOI: 10.1007/978-3-540-36119-0_4.

[162] P. Drag, K. Styczen, M. Kwiatkowska and A. Szczurek, „A Review on the Direct and Indirect Methods for Solving Optimal Control Problems with Differential-Algebraic Constraints," in *Recent Advances in Computational Optimization* (Studies in Computational Intelligence). vol. 610, S. Fidanova, ed. Cham: Springer International Publishing, 2016, pp. 91–105, ISBN: 978-3-319-21132-9. DOI: 10.1007/978-3-319-21133-6_6.

[163] H. G. Bock and K. J. Plitt, „A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems," *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 1603–1608, 1984, DOI: 10.1016/S1474-6670(17)61205-9.

[164] D. Casanova, „On Minimum Time Vehicle Manoeveuring: The Theoretical Optimal Lap," PhD Thesis, Cranfield University, 2000.

[165] F. Christ, A. Wischnewski, A. Heilmeier and B. Lohmann, „Time-optimal trajectory planning for a race car considering variable tyre-road friction coefficients," *Vehicle System Dynamics*, vol. 59, no. 4, pp. 588–612, 2019, DOI: 10.1080/00423114.2019.1704804.

[166] N. Dal Bianco, R. Lot and M. Gadola, „Minimum time optimal control simulation of a GP2 race car," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 232, no. 9, pp. 1180–1195, 2017, DOI: 10.1177/0954407017728158.

[167] A. Heilmeier, A. Wischnewski, L. Hermansdorfer, J. Betz, M. Lienkamp and B. Lohmann, „Minimum curvature trajectory planning and control for an autonomous race car," *Vehicle System Dynamics*, vol. 58, no. 10, pp. 1497–1527, 2020, DOI: 10.1080/00423114.2019.1631455.

[168] M. Imani Masouleh and D. J. N. Limebeer, „Optimizing the Aero-Suspension Interactions in a Formula One Car," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 3, pp. 912–927, 2016, DOI: 10.1109/TCST.2015.2475396.

[169] D. J. N. Limebeer and G. Perantoni, „Optimal Control of a Formula One Car on a Three-Dimensional Track—Part 2: Optimal Control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 5, p. 051019, 2015, DOI: 10.1115/1.4029466.

[170] A. J. Tremlett and D. J. N. Limebeer, „Optimal tyre usage for a Formula One car," *Vehicle System Dynamics*, vol. 54, no. 10, pp. 1448–1473, 2016, DOI: 10.1080/00423114.2016.1213861.

[171] E. Atmaca, „Energy management of solar car in circuit race," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 23, pp. 1142–1158, 2015, DOI: 10.3906/elk-1212-37.

[172] O. Borsboom, C. A. Fahdzyana and M. Salazar, „Time-optimal Control Strategies for Electric Race Cars with Different Transmission Technologies," in *2020 IEEE Vehicle Power and Propulsion Conference (VPPC)*, Gijon, Spain, 2020, pp. 1–5, ISBN: 978-1-7281-8959-8. DOI: 10.1109/VPPC49601.2020.9330828.

[173] O. Borsboom, C. A. Fahdzyana, T. Hofman and M. Salazar, „A Convex Optimization Framework for Minimum Lap Time Design and Control of Electric Race Cars," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 8478–8489, 2021, DOI: 10.1109/TVT.2021.3093164.

[174] D. Cifuentes and L. Pradenas, „Optimization of the Velocity Profile of a Solar Car Used in the Atacama Desert," in *Hybrid Metaheuristics* (Lecture Notes in Computer Science). vol. 11299, M. J. Blesa Aguilera, C. Blum, H. Gambini Santos, P. Pinacho-Davidson and J. Godoy del Campo, ed. Cham: Springer International Publishing, 2019, pp. 164–171, ISBN: 978-3-030-05982-8. DOI: 10.1007/978-3-030-05983-5_12.

[175] P. Duhr, G. Christodoulou, C. Balerna, M. Salazar, A. Cerofolini and C. H. Onder, „Time-optimal gearshift and energy management strategies for a hybrid electric race car," *Applied Energy*, vol. 282, no. 6, p. 115980, 2021, DOI: 10.1016/j.apenergy.2020.115980.

[176] S. Ebbesen, M. Salazar, P. Elbert, C. Bussi and C. H. Onder, „Time-optimal Control Strategies for a Hybrid Electric Race Car," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 233–247, 2018, DOI: 10.1109/TCST.2017.2661824.

[177] X. Liu and A. Fotouhi, „Formula-E race strategy development using artificial neural networks and Monte Carlo tree search," *Neural Computing and Applications*, vol. 60, no. 4, p. 1516, 2020, DOI: 10.1007/s00521-020-04871-1.

[178] X. Liu, A. Fotouhi and D. J. Auger, „Optimal energy management for formula-E cars with regulatory limits and thermal constraints," *Applied Energy*, vol. 279, p. 115805, 2020, DOI: 10.1016/j.apenergy.2020.115805.

[179] N. Murgovski, L. Johannesson, X. Hu, B. Egardt and Sjoeberg J., „Convex relaxations in the optimal control of electrified vehicles," in *American Control Conference (ACC), 2015: 1 - 3 July 2015, Chicago, IL, USA*, 2015, pp. 2292–2298, ISBN: 978-1-4799-8684-2. Available: http://ieeexplore.ieee.org/servlet/opac?punumber=7160954.

[180] M. Salazar, P. Duhr, C. Balerna, L. Arzilli and C. H. Onder, „Minimum Lap Time Control of Hybrid Electric Race Cars in Qualifying Scenarios," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7296–7308, 2019, DOI: 10.1109/TVT.2019.2920777.

[181] B. Alrifaee and J. Maczijewski, „Real-time Trajectory Optimization for Autonomous Vehicle Racing using Sequential Linearization," in *2018 IEEE Intelligent Transportation Systems Conference: November 4-7, Maui, Hawaii*, 2018, pp. 476–483, ISBN: 978-1-7281-0323-5. Available: http://ieeexplore.ieee.org/servlet/opac?punumber=8543039.

[182] M. Brunner, U. Rosolia, J. Gonzales and F. Borrelli, „Repetitive Learning Model Predictive Control: An Autonomous Racing Example," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Melbourne, Australia, 2017, pp. 2545–2550, ISBN: 978-1-5090-2873-3. DOI: 10.1109/CDC.2017.8264027.

[183] A. Carvalho, Y. Gao, A. Gray, E. Tseng and F. Borrelli, „Predictive Control of an Autonomous Ground Vehicle using an Iterative Linearization Approach," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2013: 6 - 9 Oct. 2013, Kurhaus, The Hague, The Netherlands*, 2013, ISBN: 978-1-4799-2914-6. Available: http://ieeexplore.ieee.org/servlet/opac?punumber=6712176.

[184] E. Frazzoli, M. A. Dahleh and E. Feron, „Real-Time Motion Planning for Agile Autonomous Vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 116–129, 2002, DOI: 10.2514/2.4856.

[185] T. Gu, J. Snider, J. M. Dolan and J.-w. Lee, „Focused Trajectory Planning for Autonomous On-Road Driving," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, Gold Coast City, Australia, 2013, pp. 547–552, ISBN: 978-1-4673-2755-8. DOI: 10.1109/IVS.2013.6629524.

[186] X. Hu, L. Chen, B. Tang, D. Cao and H. He, „Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles," *Mechanical Systems and Signal Processing*, vol. 100, no. 5, pp. 482–500, 2018, DOI: 10.1016/j.ymssp.2017.07.019.

[187] Y. Huang, H. Ding, Y. Zhang, H. Wang, D. Cao, N. Xu and C. Hu, „A Motion Planning and Tracking Framework for Autonomous Vehicles Based on Artificial Potential Field Elaborated Resistance Network Approach," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 2, pp. 1376–1386, 2020, DOI: 10.1109/TIE.2019.2898599.

[188] J. h. Jeon, R. V. Cowlagi, S. C. Peters, S. Karaman, E. Frazzoli, P. Tsiotras and K. Iagnemma, „Optimal Motion Planning with the Half-Car Dynamical Model for Autonomous High-Speed Driving," in *American Control Conference (ACC)*, Washington, DC, USA, 2013, pp. 188–193, ISBN: 978-1-4799-0178-4.

[189] X. Li, Z. Sun, D. Cao, D. Liu and H. He, „Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles," *Mechanical Systems and Signal Processing*, vol. 87, no. 8, pp. 118–137, 2017, DOI: 10.1016/j.ymssp.2015.10.021.

[190] N. Li, E. Goubault, L. Pautet and S. Putot, „Autonomous racecar control in head-to-head competition using Mixed-Integer Quadratic Programming," in *2021 IEEE/ICRA International Conference on Robotics and Automation (ICRA)*, Online, 2021.

[191] T. Lipp and S. Boyd, „Minimum-time speed optimisation over a fixed path," *International Journal of Control*, vol. 87, no. 6, pp. 1297–1311, 2014, DOI: 10.1080/00207179.2013.875224.

[192] M. McNaughton, C. Urmson, J. M. Dolan and J.-w. Lee, „Motion Planning for Autonomous Driving with a Conformal Spatiotemporal Lattice," in *2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011, pp. 4889–4895, ISBN: 978-1-61284-386-5. DOI: 10.1109/ICRA.2011.5980223.

[193] Y. Meng, Y. Wu, Q. Gu and L. Liu, „A Decoupled Trajectory Planning Framework Based on the Integration of Lattice Searching and Convex Optimization," *IEEE Access*, vol. 7, pp. 130530–130551, 2019, DOI: 10.1109/ACCESS.2019.2940271.

[194] T. Mercy, R. van Parys and G. Pipeleers, „Spline-Based Motion Planning for Autonomous Guided Vehicles in a Dynamic Environment," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 6, pp. 2182–2189, 2018, DOI: 10.1109/TCST.2017.2739706.

[195] X. Qian, F. Altche, P. Bender, C. Stiller and A. de La Fortelle, „Optimal Trajectory Planning for Autonomous Driving Integrating Logical Constraints: An MIQP Perspective," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC): Windsor Oceanico Hotel, Rio de Janeiro, Brazil, November 1-4, 2016*, 2016, ISBN: 978-1-5090-1889-5.

[196] T. Stahl, A. Wischnewski, J. Betz and M. Lienkamp, „Multilayer Graph-Based Trajectory Planning for Race Vehicles in Dynamic Scenarios," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, 2019, pp. 3149–3154, ISBN: 978-1-5386-7024-8. DOI: 10.1109/ITSC.2019.8917032.

[197] L. Svensson, L. Masson, N. Mohan, E. Ward, A. P. Brenden, L. Feng and M. Torngren, „Safe Stop Trajectory Planning for Highly Automated Vehicles: An Optimal Control Problem Formulation," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Changshu, 2018, pp. 517–522, ISBN: 978-1-5386-4452-2. DOI: 10.1109/IVS.2018.8500536.

[198] L. Svensson, M. Bujarbaruah, N. R. Kapania and M. Torngren, „Adaptive Trajectory Planning and optimization at Limits of Handling," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, pp. 3942–3948, ISBN: 978-1-7281-4004-9. DOI: 10.1109/IROS40897.2019.

[199] M. Werling, J. Ziegler, S. Kammel and S. Thrun, „Optimal Trajectory Generation for Dynamic Street Scenarios in a Frenet Frame," in *2010 IEEE International Conference on Robotics and Automation*, Anchorage, AK, 2010, pp. 987–993, ISBN: 978-1-4244-5038-1. DOI: 10.1109/ROBOT.2010.5509799.

[200] G. Williams, P. Drews, B. Goldfain, J. M. Rehg and E. A. Theodorou, „Aggressive Driving with Model Predictive Path Integral Control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, 2016, pp. 1433–1440, ISBN: 978-1-4673-8026-3. DOI: 10.1109/ICRA.2016.7487277.

[201] Y. Zhang, H. Chen, S. L. Waslander, J. Gong, G. Xiong, T. Yang and K. Liu, „Hybrid Trajectory Planning for Autonomous Driving in Highly Constrained Environments," *IEEE Access*, vol. 6, pp. 32800–32819, 2018, DOI: 10.1109/ACCESS.2018.2845448.

[202] W. Dib, A. Chasse, P. Moulin, A. Sciarretta and G. Corde, „Optimal energy management for an electric vehicle in eco-driving applications," *Control Engineering Practice*, vol. 29, pp. 299–307, 2014, DOI: 10.1016/j.conengprac.2014.01.005.

[203] J. Fleming, X. Yan and R. Lot, „Incorporating Driver Preferences Into Eco-Driving Assistance Systems Using Optimal Control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2913–2922, 2021, DOI: 10.1109/TITS.2020.2977882.

[204] E. Guerrero Merino and M. A. Duarte-Mermoud, „Online energy management for a solar car using pseudospectral methods for optimal control," *Optimal Control Applications and Methods*, vol. 37, no. 3, pp. 537–555, 2016, DOI: 10.1002/oca.2210.

[205] R. Lot and S. A. Evangelou, „Lap Time Optimization of a Sports Series Hybrid Electric Vehicle," in *The 2013 International Conference of Data Mining and Knowledge Engineering, the 2013 International Conference of Mechanical Engineering, the 2013 International Conference of Signal and Image Engineering* (Lecture notes in engineering and computer science). vol. Vol. 3 Hong Kong: IAENG, 2013, ISBN: 978-988-19252-9-9. DOI: 10.13140/2.1.2166.3522.

[206] M. Salazar, P. Elbert, S. Ebbesen, C. Bussi and C. H. Onder, „Time-optimal Control Policy for a Hybrid Electric Race Car," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 6, pp. 1921–1934, 2017, DOI: 10.1109/TCST.2016.2642830.

[207] M. Salazar, C. Balerna, P. Elbert, F. P. Grando and C. H. Onder, „Real-Time Control Algorithms for a Hybrid Electric Race Car Using a Two-Level Model Predictive Control Scheme," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 10911–10922, 2017, DOI: 10.1109/TVT.2017.2729623.

[208] M. Salazar, C. Balerna, E. Chisari, C. Bussi and C. H. Onder, „Equivalent Lap Time Minimization Strategies for a Hybrid Electric Race Car," in *2018 IEEE Conference on Decision and Control (CDC)*, Miami Beach, FL, 2018, pp. 6125–6131, ISBN: 978-1-5386-1395-5. DOI: 10.1109/CDC.2018.8618724.

[209] A. Sciarretta, G. de Nunzio and L. L. Ojeda, „Optimal Ecodriving Control: Energy-Efficient Driving of Road Vehicles as an Optimal Control Problem," *IEEE Control Systems*, vol. 35, no. 5, pp. 71–90, 2015, DOI: 10.1109/MCS.2015.2449688.

[210] D.-m. Wu, Y. Li, C.-q. Du, H.-t. Ding, Y. Li, X.-b. Yang and X.-y. Lu, „Fast velocity trajectory planning and control algorithm of intelligent 4WD electric vehicle for energy saving using time-based MPC," *IET Intelligent Transport Systems*, vol. 13, no. 1, pp. 153–159, 2019, DOI: 10.1049/iet-its.2018.5103.

[211] P. E. Gill, W. Murray and M. A. Saunders, „SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, vol. 47, no. 1, pp. 99–131, 2005, DOI: 10.1137/S0036144504446096.

[212] D. J. N. Limebeer and A. V. Rao, „Faster, Higher, and Greener: Vehicular Optimal Control," *IEEE Control Systems*, vol. 35, no. 2, pp. 36–56, 2015, DOI: 10.1109/MCS.2014.2384951.

[213] 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 Transactions on Mathematical Software*, vol. 41, no. 1, pp. 1–37, 2014, DOI: 10.1145/2558904.

[214] E. Bertolazzi, F. Biral and M. Da Lio, „Symbolic-Numeric Indirect Method for Solving Optimal Control Problems for Large Multibody Systems," *Multibody System Dynamics*, vol. 13, no. 2, pp. 233–252, 2005, DOI: 10.1007/s11044-005-3987-4.

[215] P. Kuehl, J. Ferreau, J. Albersmeyer, C. Kirches, L. Wirsching, S. Sager, A. Potschka, G. Schulz, M. Diehl, D. B. Leineweber and A. Schäfer. „MUSCOD-II Users' Manual," 2022. [Online]. Available: https://www.researchgate.net/publication/255663337_MUSCOD-II_Users%27_Manual [visited on 01/27/2022].

[216] D. B. Leineweber, A. Schäfer, H. G. Bock and J. P. Schlöder, „An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization," *Computers & Chemical Engineering*, vol. 27, no. 2, pp. 167–174, 2003, DOI: 10.1016/S0098-1354(02)00195-3.

[217] A. Domahidi, E. Chu and S. Boyd, „ECOS: An SOCP Solver for Embedded Systems," in *2013 European Control Conference (ECC)*, Zurich, 2013, pp. 3071–3076, ISBN: 978-3-033-03962-9. DOI: 10.23919/ECC.2013.6669541.

[218] P. E. E. Gill, S. J. Hammarling, W. Murray, M. A. Saunders and M. H. Wright. „User's Guide for LSSOL (Version 1.0): a Fortran Package for Constrained Linear Least-Squares and Convex Quadratic programming. Technical Report SOL 86-1," 2022. [Online]. Available: https://web.stanford.edu/group/SOL/lssol.htm [visited on 01/27/2022].

[219] IBM Corporation. „IBM ILOG CPLEX Optimization Studio Getting Started with CPLEX for MATLAB," 2021. [Online]. Available: https://www.ibm.com/docs/en/icos/12.9.0?topic=matlab-getting-started-cplex [visited on 06/13/2021].

[220] B. Houska, H. J. Ferreau and M. Diehl, „ACADO Toolkit – An Open-Source Framework for Automatic Control and Dynamic Optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011, DOI: 10.1002/oca.939.

[221] A. Zanelli, A. Domahidi, J. Jerez and M. Morari, „FORCES NLP: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs," *International Journal of Control*, vol. 93, no. 1, pp. 13–29, 2020, DOI: 10.1080/00207179.2017.1316017.

[222] Gurobi Optimization, L. L. C. „GUROBI OPTIMIZER REFERENCE MANUAL," 2022. [Online]. Available: https://www.gurobi.com/documentation/9.0/refman/index.html [visited on 01/27/2022].

[223] B. Kaepernick and K. Graichen, „The Gradient Based Nonlinear Model Predictive Control Software GRAMPC," in *2014 European Control Conference (ECC)*, Strasbourg, France, 2014, pp. 1170–1175, ISBN: 978-3-9524269-1-3. DOI: 10.1109/ECC.2014.6862353.

[224] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, T. Stahl, L. Hermansdorfer and M. Lienkamp, „A Software Architecture for an Autonomous Racecar," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, Kuala Lumpur, Malaysia, 2019, pp. 1–6, ISBN: 978-1-7281-1217-6. DOI: 10.1109/VTCSpring.2019.8746367.

[225] T. Stahl, M. Eicher, J. Betz and F. Diermeyer, „Online Verification Concept for Autonomous Vehicles – Illustrative Study for a Trajectory Planning Module," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Rhodes, Greece, 2020, pp. 1–7, ISBN: 978-1-7281-4149-7. DOI: 10.1109/ITSC45102.2020.9294703.

[226] T. Stahl, „Safeguarding Complex and Learning-Based Automated Driving Functions via Online Verification," PhD thesis, Technical University of Munich, 2022.

[227] A. Liniger, A. Domahidi and M. Morari, „Optimization-based autonomous racing of 1:43 scale RC cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2015, DOI: 10.1002/oca.2123.

[228] T. A. Ademoye and A. Davari, „Trajectory Planning of Multiple Autonomous Systems Using Mixed-Integer Linear Programming," in *Proceedings of the 38th Southeastern Symposium on System Theory*, Cookeville, TN, USA, 2006, pp. 260–264, ISBN: 0-7803-9457-7.

[229] E. Bakker and Nyborg, L. and Pacejka, H. B., „Tyre Modelling for Use in Vehicle Dynamics Studies," *SAE Transactions*, vol. 1987, no. 96, pp. 190–204, 1987.

[230] T. Herrmann and M. Luethy. „*A Real-Time Simulation Environment for Autonomous Vehicles in Highly Dynamic Driving Scenarios*," Stuttgart, 2019. Available: https://de.mathworks.com/videos/a-real-time-simulation-environment-for-autonomous-vehicles-in-highly-dynamic-driving-scenarios-1558698299193.html.

[231] L. Wildfeuer, N. Wassiliadis, C. Reiter, M. Baumann and M. Lienkamp, „Experimental Characterization of Li-Ion Battery Resistance at the Cell, Module and Pack Level," in *2019 Fourteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, Monte-Carlo, Monaco, 2019, pp. 1–12, ISBN: 978-1-7281-3703-2. DOI: 10.1109/EVER.2019.8813578.

[232] E. Alcalá, V. Puig, J. Quevedo and U. Rosolia, „Autonomous racing using Linear Parameter Varying-Model Predictive Control (LPV-MPC)," *Control Engineering Practice*, vol. 95, p. 104270, 2020, DOI: 10.1016/j.conengprac.2019.104270.

[233] L. Hermansdorfer, J. Betz and M. Lienkamp, „A Concept for Estimation and Prediction of the Tire-Road Friction Potential for an Autonomous Racecar," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, 2019, pp. 1490–1495, ISBN: 978-1-5386-7024-8. DOI: 10.1109/ITSC.2019.8917024.

[234] M. Morari, F. Allgöwer, L. Magni, D. M. Raimondo and M. Thoma, *Nonlinear Model Predictive Control: Towards New Challenging Applications*, (Lecture Notes in Control and Information Sciences). vol. 384, Berlin, Heidelberg, Springer Berlin Heidelberg, 2009, ISBN: 978-3-642-01094-1. DOI: 10.1007/978-3-642-01094-1.

[235] T. Denton, *Automated Driving and Driver Assistance Systems*, First edition, Abingdon, Oxon and New York, NY, Routledge, 2020, ISBN: 978-0367265595.

[236] L. Liu, S. Lu, R. Zhong, B. Wu, Y. Yao, Q. Zhang and W. Shi, „Computing Systems for Autonomous Driving: State of the Art and Challenges,“ *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6469–6486, 2021, DOI: 10.1109/JIOT.2020.3043716.

[237] J. Eiche, „Development of a Simulation Framework for the Validation of an Energy Strategy for an Autonomous Electric Race Car,“ Semester Thesis, Technical University of Munich, Munich, 2021.

Prior Publications

During the development of this dissertation, publications and student theses were written in which partial aspects of this work were presented.

Journals; Scopus-listed (peer-reviewed)

[4] T. Herrmann, A. Wischnewski, L. Hermansdorfer, J. Betz and M. Lienkamp, „Real-Time Adaptive Velocity Optimization for Autonomous Electric Cars at the Limits of Handling," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 4, pp. 665–677, 2021, DOI: 10.1109/TIV.2020.3047858.

[9] T. Herrmann, F. Sauerbeck, M. Bayerlein, J. Betz and M. Lienkamp, „Optimization-Based Real-Time-Capable Energy Strategy for Autonomous Electric Race Cars," *SAE International Journal of Connected and Automated Vehicles*, vol. 5, no. 1, pp. 45–59, 2022.

Conferences; Scopus-listed (peer-reviewed)

[2] T. Herrmann, F. Christ, J. Betz and M. Lienkamp, „Energy Management Strategy for an Autonomous Electric Racecar using Optimal Control," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, 2019, pp. 720–725, ISBN: 978-1-5386-7024-8. DOI: 10.1109/ITSC.2019.8917154.

[3] T. Herrmann, F. Passigato, J. Betz and M. Lienkamp, „Minimum Race-Time Planning-Strategy for an Autonomous Electric Racecar," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Rhodes, Greece, 2020, pp. 1–6, ISBN: 978-1-7281-4149-7. DOI: 10.1109/ITSC45102.2020.9294681.

Conference; not Scopus-listed

[230] T. Herrmann and M. Luethy. „*A Real-Time Simulation Environment for Autonomous Vehicles in Highly Dynamic Driving Scenarios*," Stuttgart, 2019. Available: https://de.mathworks.com/videos/a-real-time-simulation-environment-for-autonomous-vehicles-in-highly-dynamic-driving-scenarios-1558698299193.html.

Non-thesis-relevant publications; Scopus-listed (peer-reviewed)

[12] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, L. Hermansdorfer, T. Stahl, T. Herrmann and M. Lienkamp, „A Software Architecture for the Dynamic Path Planning of an Autonomous Racecar at the Limits of Handling," in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, Graz, Austria, 2019, pp. 1–8, ISBN: 978-1-7281-0142-2. DOI: 10.1109/ICCVE45908.2019.8965238.

[13] S. Kalt, K. L. Stolle, P. Neuhaus, T. Herrmann, A. Koch and M. Lienkamp, „Dependency of Machine Efficiency on the Thermal Behavior of Induction Machines," *Machines*, vol. 8, no. 1, p. 9, 2020, DOI: 10.3390/machines8010009.

[14] A. Koch, T. Bürchner, T. Herrmann and M. Lienkamp, „Eco-Driving for Different Electric Powertrain Topologies Considering Motor Efficiency," *World Electric Vehicle Journal*, vol. 12, no. 1, p. 6, 2021, DOI: 10.3390/wevj12010006.

[15] A. Koch, L. Nicoletti, T. Herrmann and M. Lienkamp, „Eco-Driving Algorithm for Electric Powertrains using Detailed Loss Models," *eTransportation*, 2022 (Submitted).

[16] K. Minnerup, T. Herrmann, M. Steinstraeter and M. Lienkamp, „Concept for a Holistic Energy Management System for Battery Electric Vehicles Using Hybrid Genetic Algorithms," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, Chicago, IL, USA, 2018, pp. 1–6, ISBN: 978-1-5386-6358-5. DOI: 10.1109/VTCFall.2018.8690563.

[17] K. Minnerup, T. Herrmann, M. Steinstraeter and M. Lienkamp, „Case Study of Holistic Energy Management Using Genetic Algorithms in a Sliding Window Approach," *World Electric Vehicle Journal*, vol. 10, no. 2, p. 46, 2019, DOI: 10.3390/wevj10020046.

[18] N. Wassiliadis, T. Herrmann, L. Wildfeuer, C. Reiter and M. Lienkamp, „Comparative Study of State-Of-Charge Estimation with Recurrent Neural Networks," in *2019 IEEE Transportation Electrification Conference and Expo (ITEC)*, Detroit, MI, USA, 2019, pp. 1–6, ISBN: 978-1-5386-9310-0. DOI: 10.1109/ITEC.2019.8790597.

[19] A. Wischnewski, M. Geisslinger, J. Betz, T. Betz, F. Fent, A. Heilmeier, L. Hermansdorfer, T. Herrmann, S. Huch, P. Karle, F. Nobis, L. Ögretmen, M. Rowold, F. Sauerbeck, T. Stahl, R. Trauth and M. Lienkamp, „Indy Autonomous Challenge - Autonomous Race Cars at the Handling Limits," in *12th International Munich Chassis Symposium 2021: Chassis.tech plus* (Proceedings), P. Pfeffer, ed. Berlin: Springer Berlin and Springer Vieweg, 2022, ISBN: 978-3-662-64549-9.

[20] A. Wischnewski, T. Herrmann, F. Werner and B. Lohmann, „A Tube-MPC Approach to Autonomous Multi-Vehicle Racing on High-Speed Ovals," *IEEE Transactions on Intelligent Vehicles*, 2022 (In Press).

[21] J. v. Kampen, T. Herrmann and M. Salazar, „Maximum-distance Race Strategies for a Fully Electric Endurance Race Car," *European Journal of Control*, 2022 (Invited Contribution, Submitted).

[22] J. v. Kampen, T. Herrmann and M. Salazar, „Maximum-distance Race Strategies for a Fully Electric Endurance Race Car," in *2021 European Control Conference (ECC)*, London, 2022 (In Press).

[23] T. Herrmann and L. F. Peiss, „Verstärkendes Lernen," in *Wie Maschinen lernen*. vol. 1, K. Kersting, C. Lampert and C. Rothkopf, ed. Wiesbaden: Springer Fachmedien Wiesbaden, 2019, pp. 203–212, ISBN: 978-3-658-26762-9. DOI: 10.1007/978-3-658-26763-6_26.

Thesis-relevant open-source software

[6] T. Herrmann. „Global static Energy Strategy,“ 2021. [Online]. Available: https://github.com/TUMFTM/global_racetrajectory_optimization/tree/master/opt_mintime_traj/powertrain_src [visited on 10/07/2021].

[7] T. Herrmann. „Velocity Optimization: Python package,“ 2021. [Online]. Available: https://pypi.org/project/velocity-optimization/ [visited on 10/07/2021].

[8] T. Herrmann. „Velocity Optimization: GitHub source,“ 2021. [Online]. Available: https://github.com/TUMFTM/velocity_optimization [visited on 10/07/2021].

[10] T. Herrmann. „Global dynamic Energy Strategy,“ 2021. [Online]. Available: https://github.com/TUMFTM/embes [visited on 10/07/2021].

Supervised Students' Theses

The following student theses were written within the framework of the dissertation under the supervision of the author in terms of content, technical and scientific support as well as under relevant guidance of the author. In the following, the bachelor, semester and master theses relevant and related to this dissertation are listed. Many thanks to the authors of these theses for their extensive support within the framework of this research project.

[237] J. Eiche, „Development of a Simulation Framework for the Validation of an Energy Strategy for an Autonomous Electric Race Car," Semester Thesis, Technical University of Munich, Munich, 2021.

A. Bach, „Optimization of the Energy Demand in Standard Driving Cycles using Machine Learning Methods," Bachelor's Thesis, Technical University of Munich, Munich, 2019.

M. Bayerlein, „Implementation of a Real-Time Capable Energy-Strategy for Autonomous Electric Racecars," Master's Thesis, Technical University of Munich, Munich, 2020.

M. J. Betz, „Trajectory Planning for Autonomous Electric Cars using Reinforcement Learning," Semester Thesis, Technical University of Munich, Munich, 2019.

F. Gessler, „Optimization of Energy Demands in Standard Driving Cycles using Dynamic Programming," Semester Thesis, Technical University of Munich, Munich, 2019.

T. Heinrich, „Parameter Estimation of an Electric Powertrain using AI," Semester Thesis, Technical University of Munich, Munich, 2020.

L. Kiesgen, „Machine Learning vs Optimisation-Based Velocity Planner for Autonomous Racecars," Bachelor's Thesis, Technical University of Munich, Munich, 2020.

T. Klotz, „Comparison of Optimization-Based Velocity Planners for Vehicles at the Limits of Handling," Master's Thesis, Technical University of Munich, Munich, 2020.

S. Ludmann, „Roborace - Extension of a Simulation Environment for the Validation of Autonomously Operating Level-5-Race Cars," Semester Thesis, Technical University of Munich, Munich, 2019.

F. Passigato, „Race-Strategy for Autonomous Electric Racecars," Master's Thesis, Technical University of Munich, Munich, 2019.

F. Sauerbeck, „Modeling and System Identification of the Electric Powertrain of an Autonomous Race Car," Master's Thesis, Institute of Automotive Technology (FTM), Technical University of Munich, Munich, 2020.