Institute for Automation and Control
Graz University of Technology
in cooperation with AVL List GmbH

Master's Thesis

Optimum Energy Management Strategies for an Electric Vehicle integrated in an Intelligent Transportation System

Mario Pfeifer, BSc

Supervisors:

Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Anton Hofer

Emre Kural, MSc

Stephen Jones, Ph.D.

Graz, May 2013

This document was written with TEXstudio, is set in LaTeX standard font, compiled with pdfLaTeX2e and `Biber`.

The LaTeX-template (latex.tugraz.at) from K. Voit is based on KOMA script and can be found online: https://github.com/novoid/LaTeX-KOMA-template

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, _____ _____

Date Signature

Kurzfassung

Elektrofahrzeuge bieten durch ihre derzeit eingesetzten Batteriesysteme eine noch immer zu geringe Reichweite, um für potenzielle Kunden attraktiv zu sein. Die vorliegende Arbeit beschäftigt sich mit energieoptimalen Regelungsstrategien zur Verbesserung der Effizienz und damit auch der Reichweite des Fahrzeugs, ohne jedoch Änderungen an der Struktur des Antriebsstrangs vorzunehmen. Ausgehend von den Spezifikationen eines reinen Elektrofahrzeugs und dem entsprechenden Modell in der Simulationsumgebung AVL CRUISE konnte ein Algorithmus entwickelt werden, der das Drehmoment effizient auf beide Elektromotoren dynamisch während der Fahrt aufteilt. Dieses Verfahren erfordert jedoch die Implementierung und Evaluierung von Kalibrationsmethoden, um einem praktischen Einsatz gerecht zu werden. Es wurden einerseits Effekte von verschiedenen Hysterese-Verfahren untersucht, andererseits konnte ein prädizierender Algorithmus entwickelt werden, der mithilfe künstlicher neuronaler Netze das zukünftige Fahrverhalten einzuschätzen vermag. Der zweite Teil beinhaltet Untersuchungen des thermischen Verhaltens von Elektromotoren sowie die Entwicklung eines geeigneten mathematischen Modells zur Integration in eine nichtlineare Optimierungsaufgabe, die durch Anwendung der Sequentiellen Quadratischen Programmierung gelöst wurde. Abschließend wurden Messdatensätze von verschiedenen Teststrecken analysiert, um die Wirkung der entwickelten Strategien im realen Fahrzeug zu bestätigen.
Diese Arbeit erfolgte in enger Zusammenarbeit mit AVL List GmbH und trägt maßgeblich zu den Zielen des europäischen Forschungsprojektes OpEneR bei.

Abstract

Electric vehicles generally suffer from their limited driving range discouraging potential costumers and consequently raising the need for specific improvements. The present work investigates several optimum control strategies without modifying the actual powertrain configuration in order to reduce the overall energy consumption, and thus also to extend the range of the vehicle. Based on the specifications of a fully electric prototype vehicle a suitable plant model developed in AVL CRUISE allowed the implementation of an energy efficient torque split algorithm, which needed to be calibrated effectively to suit for practical applications. Various methods comprise the investigation of hysteresis effects and the development of a predictive control algorithm estimating the future driving behavior using Artificial Neural Networks. The second part focuses on the thermal behavior of electric machines and the integration of a suitable thermal model into the task of optimizing the torque distribution for given road and speed profiles. The mathematical formulation led to a nonlinear problem with constraints and was solved by means of Sequential Quadratic Programming. Concluding evaluations of real measurement data finally confirm the application of the proposed control strategies with respect to an efficient energy management system.
This thesis was written in close collaboration with AVL List GmbH and contributes to the objectives of the European research project OpEneR.

Acknowledgement

This master's thesis was performed as part of the OpEneR project, grant agreement number 285526, funded by the European Commission Seventh Framework Programme FP7.

First of all I would like to express my gratitude especially to Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Anton Hofer from TU Graz and my supervisors Emre Kural, MSc and Stephen Jones, Ph.D. from AVL Graz for guiding me through the whole thesis and providing me a pleasant working environment.

I would like to thank everybody who encouraged me during my studies at TU Graz over the last few years, especially my family, friends and Tanja - for their loving support which I truly appreciate.

Mario Pfeifer
Graz, May 2013

Contents

List of Figures

List of Tables

List of Abbreviations

4-WD ..... Four-wheel-drive
ANN ..... Artificial Neural Network
AVL ..... Anstalt für Verbrennungskraftmaschinen List
BM ..... Benchmark
CO_2 ..... Carbon dioxide
DC ..... Direct current / driving cycle (depending on context)
EM ..... Electric machine
ESP ..... Electronic Stability Program
ESP®hev .... Regenerative braking module developed by Bosch GmbH
EV ..... Electric Vehicle
FTP75 ..... Federal Test Procedure (from 1975)
GPS ..... Global Positioning System
HEV ..... Hybrid Electric Vehicle
(I)PMSM .... (Interior) Permanent Magnet Synchronous Motor
KKT ..... Karush-Kuhn-Tucker (conditions)
MSE ..... Mean squared error
NEDC ..... New European Driving Cycle
NLP ..... Nonlinear problem
OpEneR ..... Optimal Energy Consumption and Recovery based on a system network
PM ..... Permanent magnet
QP ..... Quadratic problem
rpm ..... Revolutions per minute
SOC ..... State of charge
SQP ..... Sequential Quadratic Programming

1 Introduction

Over the past few years global climate change and the impact of greenhouse gas emissions have been highly discussed and became an issue for several conferences around the world. Already in 1972 during the climate change conference in Stockholm, it was found out that global warming is a matter of *world-wide* restrictions and regulations [Uni]. Following the idea of defining obligations in terms of a global pollution control, the famous Kyoto-Protocol was established in 1997 which forces all participating countries to significantly reduce their own amount of greenhouse gas emissions.

Since 23 % of all such emissions were produced by traffic and transport in 2004, whereof around $\frac{3}{4}$ belong to CO_2 emissions of pure road vehicles [Sch09, p. 8], the potential of reducing the overall amount of greenhouse gas emissions in the transport sector is enormous. Influenced by climate conventions in the United States, the European Parliament announced a regulation in 2009 to restrict the CO_2 emissions of passenger vehicles to 95 g CO_2/km by 2020 [Eur09]. According to Statistik Austria [Sta], the average amount of CO_2 emissions based on new registrations of passenger cars in 2011 has been reduced from 176 to 138 g CO_2/km within 10 years.

Through the use of electric and hybrid electric vehicles (EV, HEV) the overall air pollution caused by the transport sector can be reduced much more since the amount of CO_2 emissions decreases through the application of electric driven parts in the powertrain. Consider table 1.1 where current vehicle statistics for Austria in 2010/2011 are shown.

Although the trend to hybrid and electric driven engines in passenger cars has not stopped during the last few years, for example in table 1.1 the number of EV has increased by 180 % from 2010 to 2011, the total amount of HEV and

Engine type	Stock				Change (%)
	2011		2010		
	absolute	rel. (%)	absolute	rel. (%)	
Petrol	1.997.066	44,2	1.988.079	44,8	0,5
Diesel	2.506.511	55,5	2.445.506	55.1	2,5
Electric	989	0,0	353	0,0	180,2
Hybrid	6.056	0,1	4.792	0,1	26,4

Table 1.1: Excerpt of passenger car statistics for Austria [Sta]

EV being sold is still very low compared to ordinary combustion-engine cars. According to several surveys, e.g. the institute for social-economic research in Frankfurt analyzed the attractiveness and acceptance of electric vehicles in 2011 [GSB11, p. 7], it was found out that mainly following criteria discourage potential costumers from buying EV:

- too short driving ranges
- (still) too expensive
- relatively long charging times
- rather slow

While E-cars can easily reach speeds up to 140 km/h nowadays, charging times can still be rather long and mostly depend on the capacity of charging stations and the installed battery technology. In general, very low charging times would require accordingly high currents in return which can damage the battery seriously. Considering typical charging times of several hours, it is clear that E-cars cannot compete against the short time usually spent on refueling vehicles with ordinary combustion engines. However, the most crucial factor might still comprise the limited driving range which nowadays restricts possible application fields of EV primarily to inner-urban areas.

The aim of this thesis is to extend the range of fully electric vehicles through several energy management strategies without increasing battery size, and thus also by limiting charging times and prices of the battery. The developed control algorithms contribute significantly to the overall objectives presented in the research project "OpEneR", which is briefly explained in the following section.

1.1 The OpEneR research project

OpEneR stands for "Optimal Energy consumption and Recovery based on system network" and is a current research project funded by the European Union with a total budget of 7.7 million Eur. It is part of the Seventh Framework Programme (FP7) launched in 2007, which is the EU's main instrument for funding research in different scientific disciplines. The OpEneR-project partners are the Austrian powertrain development company AVL List GmbH, the Spanish research institute Centro Tecnológico de Automoción de Galicia (CTAG), the German Forschungszentrum Informatik Karlsruhe (FZI), the French car manufacturer Peugeot Citroën (PSA), and the German companies Robert Bosch GmbH and Robert Bosch Car Multimedia GmbH [OpE].

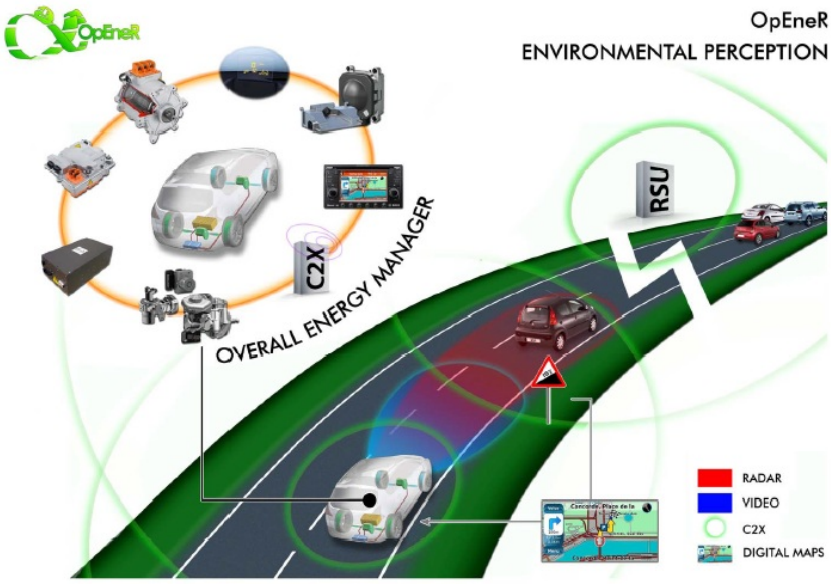

Figure 1.1: OpEneR - environmental perception [OpE]

OpEneR focuses on the development of new driving strategies which aims to extend the range of electric vehicles with respect to safety and comfort. Software control algorithms will be implemented in an overall "Energy Manager" which collects car-specific data as well as additional information from the environment (see fig. 1.1). In other words, current states of the vehicle, battery, e-machine and regenerative braking system is merged with on-

and offboard data coming from various sources such as radar sensors, GPS, car2car, car2infrastructure, etc. in order to optimize the energy management within the safety limits of the vehicle.

Through the development and integration of sophisticated energy management functions and vehicle control strategies significant improvements regarding the overall efficiency can be achieved. These functions usually rely on the optimization of utilizing the vehicle's internal components and are supported by various information coming from environmental perception and the prediction of future road or traffic conditions.

A powerful simulation tool chain forms the basis for any development and evaluation process, since any complex integration of management functions into the vehicle's control software can be replaced by virtual simulations achieving lots of benefits, such as greater comfort, higher efficiency during the development, testing and evaluation phase resulting in a decrease of error-rate, etc. Thus, the proposed tool chain comprises:

- AVL CRUISE for simulations and analysis of powertrain and vehicle dynamics,
- IPG CarMaker for complex 3D vehicle dynamics and the interaction between other traffic objects,
- Matlab/Simulink for the development of energy management and vehicle control strategies including the evaluation of simulation results.

1.2 Scope of work

In chapter 2 of this thesis a suitable vehicle model for the proposed EV used within the OpEneR project is introduced. Furthermore the most important physical relations affecting a vehicle when driving on a straight road are described, also the correspondent subsystems forming the actual powertrain of the vehicle are investigated with special focus on the transmission of torque from EM to the wheels. An essential part of this chapter comprises various reflections on the type of EM used in the proposed vehicle, including the

description of structure, efficiency, modeling and recuperation limits of the EM.

Chapter 3 investigates several strategies for an optimum distribution of the torque demand which also suits for practical applications. These so called calibration techniques comprise different ways for improving various powertrain aspects such as EM-switching times in order to ensure practical applicability with respect to lowest energy consumption. The second part of this chapter evaluates the possibility for estimating the future driving behavior which can be used to suppress certain EM switches causing unnecessary EM synchronization phases and additional energy consumption. The method decided to use for this prediction task is called Artificial Neural Networks which simply tries to generate patterns based on specified features of the driving behavior collected in the past. Finally the best designs of all presented methods are compared to each other giving clues about applicability and energy efficiency.

In contrast to all previously developed strategies, chapter 4 focuses on optimization with respect to the variation of EM temperature. The objective is to generate the optimum temperature trajectories of front and rear EM for a certain torque and speed profile within a predefined time interval. The mathematical formulation of this task leads to a nonlinear optimization problem with restrictions and will be solved using the method of Sequential Quadratic Programming. Since there is a need to integrate the EM temperature behavior into the algorithm, the development of a suitable thermal model has to be conducted in advance. As a basis for the validation step, several measurements of vehicle test runs were provided to ensure that the model correlates with the real physical behavior as much as possible. In the end results of the optimization task are presented and analyzed.

Chapter 5 finally deals with the evaluation of real measurement data provided by several runs on a dedicated test track. After the recorded measurements have been analyzed properly based on a distance oriented approach, the chapter concludes with comparisons to previously obtained simulation results and the practical applicability of investigated torque split strategies.

2 Vehicle model

In general modeling aims at representing the most important aspects of a system in order to form the basis for any further design process. In case of vehicle modeling, physical characteristics of a specified passenger car are reflected by a suitable plant model, which allows to be extended by various control algorithms and afterwards to be evaluated with the help of an appropriate simulation environment. Within OpEneR two prototype vehicles are built to implement and demonstrate the effects of a smart energy manager trying to increase the vehicle's overall efficiency.

Prototype specifications

The base prototype vehicle for the OpEneR project is a Peugeot 3008 EV provided by PSA, offering fully electric 4-wheel-drive (e4-WD) via two electric machines, built up as permanent magnet synchronous motors (PMSM). Each EM drives one axle of the vehicle and provides 50 kW power and 210 Nm of mechanical torque. The rear EM can also be used as a generator to recuperate electric energy in case of moderate deceleration. Furthermore the EV is equipped with a 40 kWh battery package providing approximately 200 km of electric range and top speeds of 140 km/h on highways. [OpE].

Consider fig. 2.1 illustrating the correspondent vehicle model for the OpEneR project car. It is adapted from the actual plant model layout of AVL CRUISE, which served as an appropriate simulation environment for vehicle and powertrain analysis throughout the whole thesis' scope of work.
All important subsystems are highlighted via frames, for instance, the dark shadowed boxes represent the four wheels of the vehicle. Dotted lines indicate electric parts/connections of the system, whereas mechanical parts such

as EM, transmission gear, clutch or differential can be distinguished through full lines. The controller consists of all necessary functions which handle the (efficient) interaction between all subsystems.

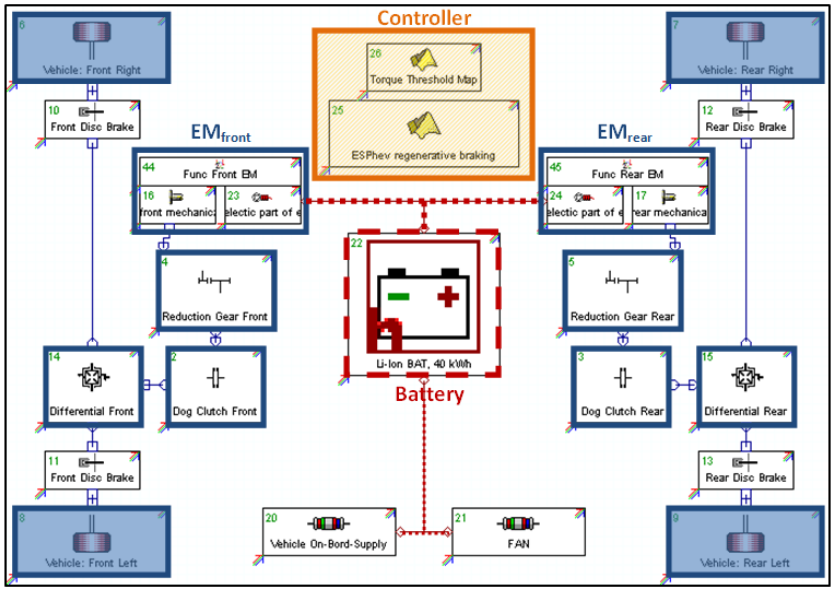

Figure 2.1: Vehicle model based on the layout of AVL CRUISE with emphasize on powertrain and control subsystems.

2.1 Longitudinal dynamics

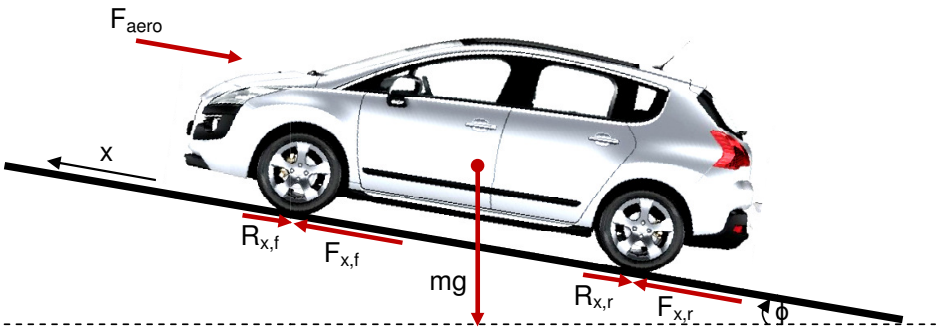

Figure 2.2: External longitudinal forces acting on a vehicle moving on an inclined road (Peugeot 3008 prototype vehicle, ©peugeot.at).

The analysis of longitudinal vehicle dynamics relates to the investigation of various effects occurring in the direction of movement, exclusively. Besides the consideration of lateral dynamics and pitch movements of a vehicle, it plays the major role in terms of energy consumption and will therefore give attention to some important explanations in this section.

Imagine a car moving on an inclined road along the way x as visualized in fig. 2.2. Several forces are acting on the vehicle, which include gravity-, rolling resistance-, aerodynamic drag- and longitudinal tire forces [Raj06].

According to Newton's 2nd law the equation of motion is expressed as follows:

$$m\,\ddot{x} = F_{x,f} + F_{x,r} - F_{aero} - R_{x,f} - R_{x,r} - mg\,\sin\phi \tag{2.1}$$

where

$F_{x,f}, F_{x,r}$ ..... longitudinal tire force due to traction at front/rear tires
F_{aero} ..... correspondent longitudinal aerodynamic drag force
$R_{x,f}, R_{x,r}$ ..... force due to rolling resistance at the front/rear tires
m ..... mass of the vehicle
g ..... acceleration due to gravity
ϕ ..... angle of road inclination

The longitudinal tire forces $F_{x,f}$ and $F_{x,r}$ acting on the drive wheels are responsible for initiating the movement of the vehicle. They depend on the normal force on the tire as well as on friction coefficients of the tire-road interface and a possible slip ratio [Raj06].

F_{aero} is the correspondent aerodynamic drag force acting on the vehicle when moving forward. It is proportional both to the front area of the car and to the square sum of the current velocity $\dot{x}$ and wind velocity v_{wind} (positive for headwind, negative for tailwind). Usually the frontal area A_F is in the range of 79-84 % of the area calculated from the vehicle's width and height [Raj06].

$$F_{aero} = \frac{1}{2}\,\rho\,c_d\,A_F\,(\dot{x} + v_{wind})^2 \tag{2.2}$$

where ρ is the mass density of air and c_d is the aerodynamic drag coefficient.

Rolling resistance forces $R_{x,f}$ and $R_{x,r}$ reflect the loss of energy when tires are rotating due to special deformation and recovering processes happening inside the tires. They act oppose the motion of the vehicle and are proportional to the normal force on the tires ($F_{z,f}, F_{z,r}$):

$$R_{x,f} + R_{x,r} = f(F_{z,f} + F_{z,r}) \tag{2.3}$$

where f is the rolling resistance coefficient [Raj06].

2.2 Powertrain dynamics

The key task for a powertrain implemented in an EV like the Peugeot 3008 project car is to transform electric energy saved in the battery into kinetic energy transmitted to the drive wheels. Consider fig. 2.3 which illustrates the powertrain for a single wheel based on the vehicle model previously introduced in fig. 2.1. Note that both front and rear axle are built up identically in the proposed EV, each supported by one separate EM.

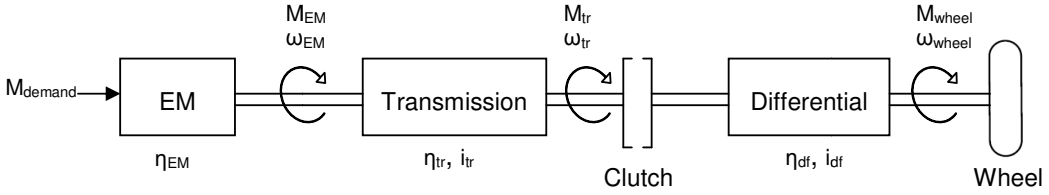

Figure 2.3: Single wheel powertrain - consisting of (front or rear) electric machine, transmission block, dog clutch and differential gear ending up at a single drive wheel.

Based on the torque demand requested by the powertrain control software, the corresponding EM delivers the actual torque M_{EM} while running the axle at the desired speed ω_{EM}. The reduction gear box is characterized by its efficiency η_{tr} and constant transmission ratio i_{tr}, which can be expressed as

$$i_{tr} = \frac{\omega_{EM}}{\omega_{tr}} \tag{2.4}$$

Furthermore it is assumed that the efficiency of the dog clutch is always 1, which means that any possible slip during closing and opening of the clutch is avoided.

The differential gear is responsible for a smooth transmission of the driving torque onto both wheels connected to the correspondent axle and adjusts different wheel speeds in a bend [Gri11]. The equivalent ratio $i_{df} = \frac{\omega_{tr}}{\omega_{wheel}}$ can be defined similarly to eq. (2.4).

M_{wheel} and ω_{wheel} finally denote the actual torque and rotational speed at the drive wheel provided by the overall powertrain.

The equation of power yields to the following relations:

$$P_{EM} = \omega_{EM}\, M_{EM} \quad = \quad \omega_{tr}\, M_{tr}\, \frac{1}{\eta_{tr}} \tag{2.5}$$

$$\omega_{tr}\, M_{tr} \quad = \quad \omega_{wheel}\, M_{wheel}\, \frac{1}{\eta_{df}} \tag{2.6}$$

Combining eq. (2.4)-(2.6) enables the calculation of the actual torque at the drive wheel as a function of M_{EM}. Note that the overall efficiency of the powertrain can be simplified to $\eta_{PT} = \eta_{tr}\, \eta_{df}$ since all parameters are assumed to be constant.

$$M_{wheel} = M_{EM} \cdot i_{tr}\, i_{df} \cdot \eta_{PT} \tag{2.7}$$

It is important to mention that the behavior of a real vehicle's drive line always requires the involvement of dynamical transitions for the torque transfer from EM to the tires. In AVL CRUISE simulation environment this aspect is modeled by 1^{st} order dynamical equations using inertias of powertrain elements.

2.3 Electric machine

The Peugeot 3008 EV proposed in the OpenER research project is equipped with two identical EMs offering full electric 4-WD at any time. Each EM is built up as a permanent magnet synchronous machine (PMSM) which generally belongs to the class of rotating field machines.

PMSM - basic structure and mode of operation:
The physical principle for generating torque in electric machines relies on the "Lorentz-force" $\vec{F} = q\,(\vec{v} \times \vec{B})$, which describes the force $\vec{F}$ acting on a current flowing electric conductor being exposed to a magnetic field $\vec{B}$.

Consider fig. 2.4 visualizing the schematic structure of a PMSM. The stator forms the chassis of the EM and consists of a three-phase winding ordered in several channels. Depending on how the permanent magnet is realized in the rotor, PMSM can be divided into surface mounted PM (see fig. 2.4) and interior PM synchronous machines. While the former is easier to construct it has the disadvantage of poor robustness. IPMSM are typically used for high speed applications since they can compromise this negative aspect by a more complex manufacture [Acq12].

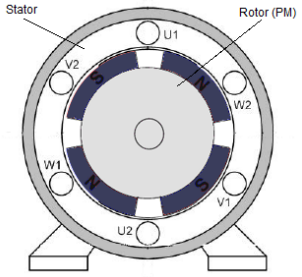

Figure 2.4: PMSM

Basically, after inversion of the supplied DC-current coming from the battery, the windings generate a rotating field according to the frequency adjusted at the inverter. In case of PMSM the rotor consists of a permanent magnet causing a static magnetic field $\vec{B}$ in the meantime. As a result of the Lorentz-force acting on the stator windings and the interlacing of the magnetic fields the rotor follows a rotation of the same (synchronous) frequency as the stator's rotating field.

The Energy-Manager as a part of the powertrain control software is responsible for an adequate load distribution on both EMs, since the choice of operating points has a high impact on the corresponding energy efficiency. For electric *motors* the term efficiency η_{EM} is defined as the relation between

delivered mechanical power and consumed electric power. Note, that the rear EM can also be used as a generator in case of "recuperation", which will be explained in the following section. Consider fig. 2.5 which illustrates typical torque- and efficiency characteristics for permanent magnet synchronous EM.

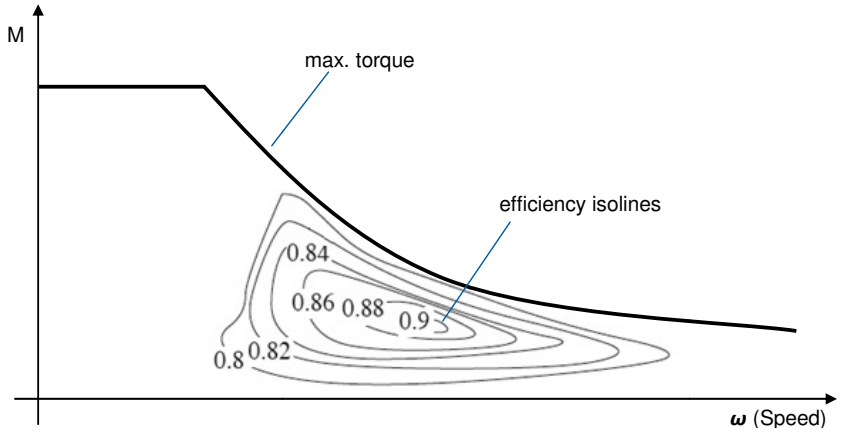

Figure 2.5: Typical torque and efficiency characteristics for PMSM (adapted from [GS07])

It is interesting to note, that the curve for the maximum torque a motor can provide follows a hyperbolic decrease after a region of constant torque. The reason for this is the limitation of current and power. For low speeds the battery provides the maximum deliverable current so that high torque requests can be supplied. In case of high speeds the power limitation is active, which means that the EM reaches its maximum transferable power. From the relation $P_{EM} = \omega_{EM} M_{EM}$ the hyperbolic decrease can be derived [GS07, p. 75].

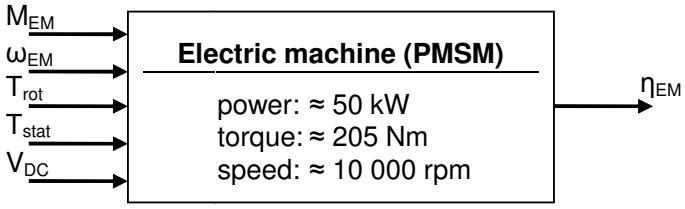

Figure 2.6: EM-model as a system with five inputs and one output η_{EM}

In Matlab/Simulink a black box model of the specified EM was provided by R. Bosch GmbH. Given a set of inputs consisting of torque M_{EM}, speed ω_{EM}, temperature T_{rot}, T_{stat} (rotor/stator) and voltage V_{DC} provided by the battery, the resulting efficiency can be obtained during simulation and used for upcoming optimization processes. Consider figure 2.6, which describes the EM-model as a system with five inputs and one output so that following relation can be expressed:

$$\eta_{EM} = f(M_{EM}, \omega_{EM}, T_{rot}, T_{stat}, V_{DC}) \tag{2.8}$$

Note that optimizing the vehicle's energy consumption is equivalent to the maximization of the combined EM-efficiency, which will be the main target for different optimum control strategies introduced in the following chapters of this work.

2.4 Regenerative braking

This section briefly introduces the recuperation ability implemented in the OpEneR project car, which was developed especially for hybrid and electric vehicles by R. Bosch GmbH under the name ESP®hev [Bos].

In case of an ordinary vehicle, pushing the braking pedal by the driver would require the mechanical brakes to be activated which implies that kinetic energy would be dissipated as heat into the environment due to friction at the brakes. This loss of energy can be partially recovered by the use of regenerative braking or so called "recuperation". Depending on the actual deceleration of the vehicle the currently required braking torque would be split onto mechanical brakes and EM, which now works as a generator transforming mechanical energy over the powertrain back into electric energy, which can be saved in the battery. Sometimes the so called e-braking torque is high enough to decelerate the vehicle completely without using any mechanical brakes, resulting in various advantages like energy savings and increasing the durability of the braking system.

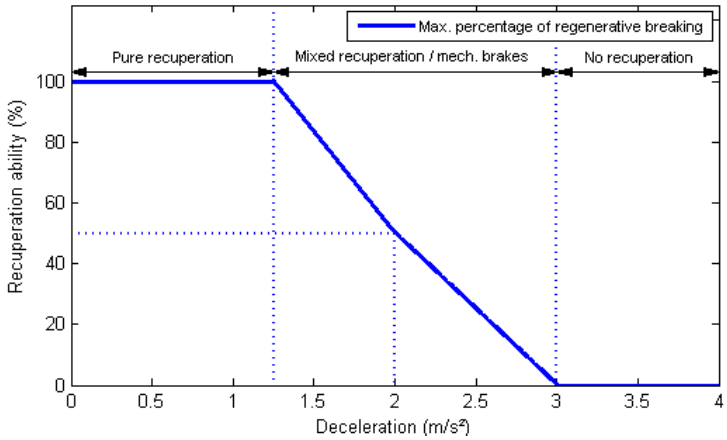

Figure 2.7: Recuperation ability for Peugeot 3008 EV defined by ESP®hev

At the time the work for this thesis was conducted, it was assumed that the proposed OpEneR project car was able to recuperate only with the rear EM. The particular e-braking torque was calculated using a predefined characteristic provided by R. Bosch GmbH, where the maximum available torque was dependent on the current braking request of the driver resulting in the correspondent deceleration level of the vehicle.
Cf. fig. 2.7 for the detailed behavior. Up to 1 m/s^2 full recuperation can be applied, whereas high deceleration values of more than 3 m/s^2 require the use of pure mechanical brakes.

3 Torque split calibration

A powertrain management software relies on instantaneous inputs of the driver as well as on current vehicle and powertrain states. The overall torque demand in an EV primarily depends on the acceleration pedal position, which is controlled by the driver in order to accelerate the vehicle. In the OpEneR project-cars, two equivalent electric machines are used to drive each axle, front and rear, separately. For this special case the question arises how to divide the actual torque demand among both EM during the drive.

In [Mas12] an online algorithm for the optimum distribution of the currently requested torque amount has been developed. The main idea that is followed in this approach is to maximize the overall efficiency of both EM, which generally depends on several factors such as speed, torque demand, battery voltage and EM- temperature. As it will be derived within the first section of this chapter, a pure application of this algorithm leads to an undesired behavior which needs to be adjusted before a car-integration process can be conducted.
The "Direct calibration" methodology uses hysteresis effects based on the current state of the vehicle to improve the overall control strategy. Within this approach the most promising design will be selected among various hysteresis implementations based on different selection criteria.
The second investigated method focuses on the prediction of the future driving behavior especially to reduce unnecessary changes of the powertrain control states without using hysteresis effects. AVL CRUISE, an appropriate simulation environment for vehicle and driveline analysis, will show possible impacts primarily in terms of energy consumption for all evaluated designs.

3.1 Review and problem definition

In this section the basic approach towards the optimum torque split algorithm for the proposed EV is briefly reviewed before several issues concerning the pure application of this algorithm are discussed.

Again picking up the question how to divide the torque demand efficiently among both EM implemented in the OpEneR-project cars, one trivial option might be to use an equal distribution. This means to split the demand via a 50:50-distribution onto front and rear EM *constantly* during the drive. However, simulations on various driving cycles have shown that this option yields to the worst results in terms of energy consumption. Given the fact that this setup is initially used in the original vehicle, it will serve as an important benchmark for comparisons later on.

While relatively high torque demands generally require both EM being active during traction, significantly better results can be achieved by switching off one EM in case of low torque demands. From the mathematical point of view, calculating the dynamical load distribution with respect to both EM requires the definition of the torque split factor at first:

$$u_f := \frac{M_{1EM,f}}{M_{1EM,f} + M_{1EM,r}} = \frac{M_{1EM,f}}{M_{2EM}} \tag{3.1}$$

$M_{1EM,f}$ ..... Torque demand at front EM in Nm
$M_{1EM,r}$ ..... Torque demand at rear EM in Nm
M_{2EM} ..... Total torque demand in Nm

According to eq. (3.1) u_f denotes the amount of torque provided by the front EM relatively to the total torque demand. Note that $0 \leq u_f \leq 1$. However, imagine two identical EM with exactly the same temperature and voltage provided by the battery. Since relation (2.8) holds for both EM it is clear, that for a given speed ω a torque split factor u_f will now give the exact same combined result as for the factor $(1 - u_f)$. Therefore it is sufficient to consider only values between the interval [0, 0.5].

Since the combined efficiency of both EM will be maximized in the algorithm, following equation expresses the degree of efficiency for electric machines both in motor mode as well as in generator mode (recuperation case).

$$\eta_{1EM} = \begin{cases} \dfrac{P_{1EM}}{P_{el}} = \dfrac{M_{1EM}\,\omega}{P_{el}} & \text{for motor mode} \quad (3.2) \\ \dfrac{P_{el}}{P_{1EM}} = \dfrac{P_{el}}{M_{1EM}\,\omega} & \text{for generator mode} \quad (3.3) \end{cases}$$

η_{1EM} ..... EM efficiency
P_{el} ..... electric power of 1 EM in W
P_{1EM} ..... mechanical power of 1 EM in W
ω ..... EM speed in $\frac{\text{rad}}{\text{s}}$

Combining the efficiency of each EM with the definition of the torque split factor, the overall efficiency of both EM can be expressed as a nonlinear function in u_f and η_f, η_r (cf. [Mas12, p. 15]).

$$P_{el,tot} = \frac{u_f\,P_{2EM}}{\eta_f} + \frac{(1-u_f)\,P_{2EM}}{\eta_r} \qquad \text{for } P_{2EM}, P_{el,tot} > 0 \quad (3.4)$$

$$\eta_{2EM} = \frac{P_{2EM}}{P_{el,tot}} = \frac{\eta_f\,\eta_r}{u_f\,\eta_r + (1-u_f)\,\eta_f} \qquad \text{for } P_{2EM}, P_{el,tot} > 0 \quad (3.5)$$

$P_{el,tot}$ ..... overall electrical power in W
P_{2EM} ..... overall mechanical power in W
η_{2EM} ..... combined efficiency of both EM

In case of negative torques where the rear EM runs in generator mode for energy recuperation, eq. (3.4) and (3.5) have to be slightly changed with respect to relation (3.3).

The approach for calculating the optimum torque split factor can now be described as follows. Basically, given the EM model in Matlab/Simulink efficiency values for different torque-, speed- and voltage inputs covering the whole operating area of both EM have to be collected. Since the EM temperature was kept constant for the upcoming study in this chapter, possible influence of temperature variations in this step can be neglected as well.

In order to ensure feasibility of the algorithm on a digital computer the torque split factor $u_f \in [0,\ 0.5]$ has to be discretized into a sufficiently large amount of values $u_{f,i}$ for $i = 1, 2, .., n$, so that: $u_f = \{u_{f,1}, u_{f,2}, .., u_{f,n}\}$.

In the next step, combined efficiency values $\eta_{2EM,i}$ are determined based on n different distributions of the torque demand onto both EM. For instance at step i the torque split is defined according to eq. (3.1):

$$\begin{aligned} M_{1EM,f} &= M_{2EM} \cdot u_{f,i} \\ M_{1EM,r} &= M_{2EM} \cdot (1 - u_{f,i}) \end{aligned}$$

The optimal torque split factor for the triple (M_{2EM}, ω, V_{DC}) can now be derived according to the previously obtained efficiency values $\eta_{2EM,i}$ for all steps $i = 1, 2, .., n$ and saved in a lookup table for a fast and efficient access:

$$\begin{aligned} j &= \arg\max_i \ \eta_{2EM,i} \\ u_{f,opt} &= u_{f,j} \end{aligned}$$

Consider fig. 3.1 illustrating the optimal torque split factor over the feasible torque-speed area of both EM during traction:

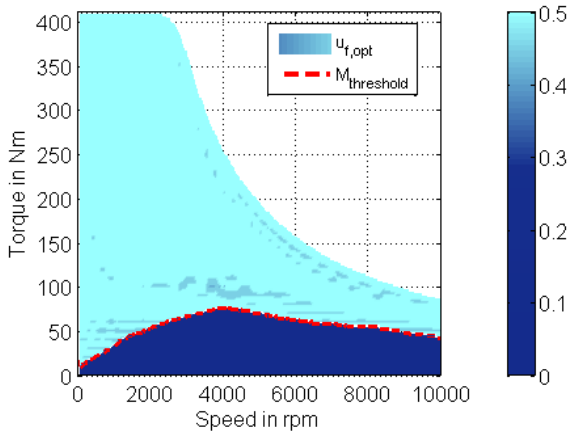

Figure 3.1: Optimal torque split factor $u_{f,opt}$ for traction mode with step size 0.05 and selected parameters $V_{DC} = 305\,V$, $T_{rot} = T_{stat} = 60\,°C$. The threshold line separates two major split strategies: In the lower torque area $u_{f,opt}$ is zero which corresponds to traction with rear EM only. Above the threshold line the majority of all optimal split values within the feasible operating area implies an equal distribution between front and rear EM.

Taking a closer look on fig. 3.1, the majority of all calculated optimal torque split factors comprises values of 0 or 0.5, which corresponds to use either the rear EM or both EM equally for a certain input configuration. Thus it is useful to separate both modes with a threshold line in order to clarify the use of two major split strategies for further considerations. With respect to the upcoming study, this option will be called as the *suboptimal* design approach, since only two discrete values $u_{f,subopt} \in \{0, 0.5\}$ are considered here - compared to the *optimal* design, respectively, where all (simulated) values of u_f in the interval [0, 0.5] are used.

Given the fact that computations generally need to be carried out as fast as possible especially with respect to a further integration in the vehicle's control software, the use of the suboptimal design approach can be clearly justified since it simplifies the optimum algorithm essentially. But also in terms of energy consumption the suboptimal design still reaches satisfying results compared to the optimal approach which will also be proved by simulation results in the next sections.

Problem definition

Recent simulations with an applied version of the previously introduced suboptimal torque split algorithm have revealed that the resulting driving behavior is not sufficient for practical reasons and thus the path for further modifications and improvements is opened. For illustration, in fig. 3.2 the velocity profile and in fig. 3.3 the simulated torque split characteristics for two different driving cycles, NEDC[1] and FTP-75[2], are shown.

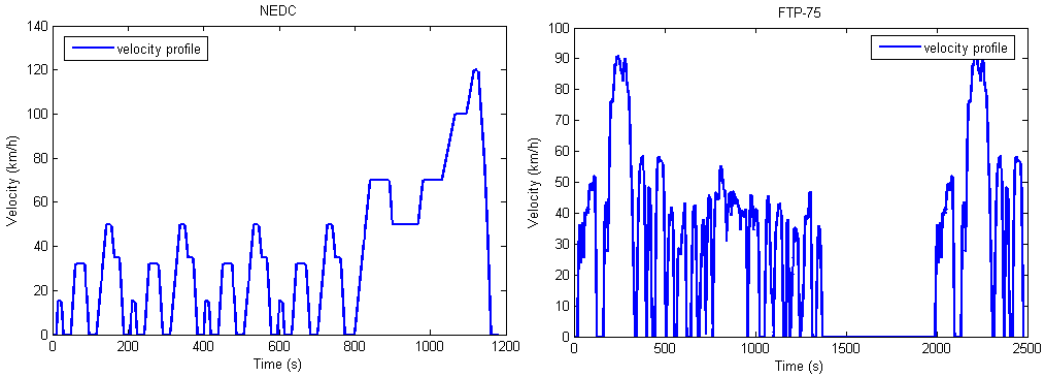

Figure 3.2: Velocity profile for two different driving cycles: NEDC (left) and FTP75 (right)

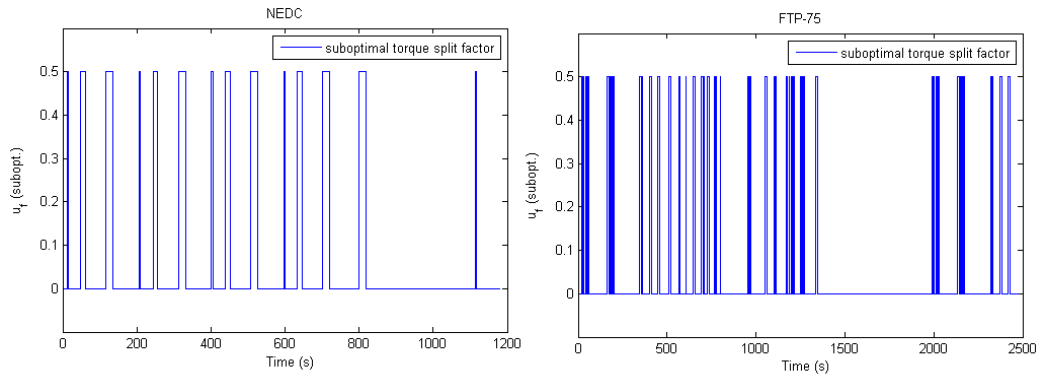

Figure 3.3: Corresponding suboptimal torque split characteristic for both driving cycles. In case of FTP-75 an oscillating behavior can be observed since the minimum time frame between two consecutive switches of the driving mode is remarkably small compared to NEDC.

[1] New European Driving Cycle
[2] Federal Test Procedure - established in 1975

In NEDC, the minimum time interval between two different (suboptimal) driving modes is about 1.15 s. Since NEDC is a rather artificial driving cycle with several redundant passages of easy acceleration and brake maneuvers, this value is not fully significant for further evaluations although the minimum switching time is still very low.
On the right side of 3.3, u_f-characteristics for FTP-75 are illustrated. In opposite to NEDC, FTP-75 does not consist of clearly delimited regions and tries to reproduce a more realistic driving behavior instead. Here, the minimum time interval between two changes of the torque split factor is already close to zero or, to be more precise, at the simulation time limits.

Different aspects showing the overall problem situation for the presented torque split algorithm can be listed as follows:

1.) Comfort, durability and safety considerations
In general, applying the pure suboptimal control algorithm to an EV's Energy Manager causes very high switching rates of the driving modes, which results in an oscillating behavior of the front EM. Especially for durability reasons this effect is considered as insistently undesirable since fast on/off-switching of an EM causes inadvertent operations of internal parts which consequently leads to an unnecessary wearing of mechanical components and can damage the whole EM in the long term. Furthermore it has to mentioned, that fast changes of the driving mode are generally regarded as infeasible for practical applications due to the physical time constant of engaging and disengaging of electric machines. Additional comfort and safety concerns regarding the decrease of torque split switches with respect to current road and driving conditions conclude the requirement for an appropriate adjustment of the proposed control strategy.

2.) Additional energy consumption
Another issue regarding the change of driving modes covers additional energy consumption that occurs, when one specific powertrain state has to be moved to another state due to the change of the torque split factor. For instance, if the transition corresponds to $u_f : 0 \rightarrow 0.5$, hence

the torque demand of the rear EM will be splitted among both EM in the next step, a speed synchronization process has to be applied to the unused (front) EM before the clutches close in order to facilitate a smooth torque transfer. This implies a certain time delay until the whole powertrain works in its requested state again, since lags between the detection of a mode change and the physical execution occur due to the previous preparation and synchronization phase. Especially when u_f-switches are really close to each other like in fig. 3.3, additional efforts for a mode change might result in higher energy consumption and should therefore be avoided.

In the following sections two different approaches as a possibility to deal with these problems will be introduced. While the first one tries to apply direct calibration methods based on the past and current states of the powertrain, the second approach tries to predict the driving behavior itself and avoid unnecessary switches in advance. Furthermore, several simulations will show that adequate designs can also limit an additional increase of energy consumption which always occurs when optimality of the overall system is compromised.

3.2 Direct calibration

In this section two different calibration methods will be introduced, each applying special hysteresis effects to improve the overall control strategy. The basis for this calibration technique is the suboptimal design approach where two major split designs are separated through a single threshold line. The major objective is to improve the oscillating torque split behavior with strong emphasize on retaining energy improvements which have been gained through the use of the suboptimal design approach for different driving cycles. Both methods only use past and present states of the powertrain in order to directly choose the next pending driving mode.

3.2.1 Time based hysteresis

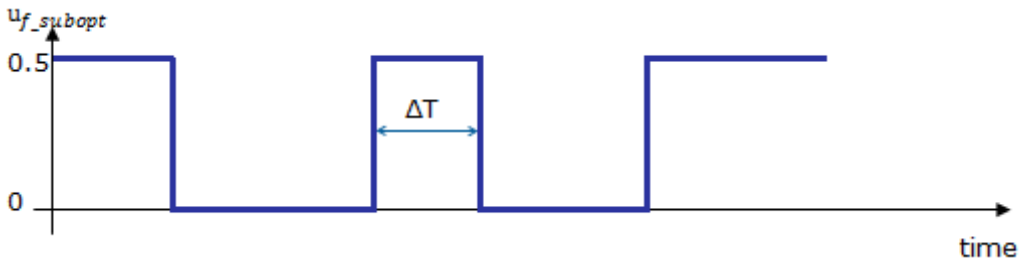

Figure 3.4: Time based hysteresis design approach with ΔT as a varying parameter to limit the minimum switching time

The idea behind the method of a time based hysteresis is to force the torque split factor to stay at a certain value for a previously defined time span ΔT in order to prevent fast changes of u_f already in the first place. Compare fig. 3.4, where ΔT is the minimum time where no switches are allowed, both for trajectories $u_f : 0.5 \rightarrow 0$ and $u_f : 0 \rightarrow 0.5$.

Varying ΔT leads to several options for possible designs which will be evaluated for different driving cycles based on several metrics, such as the total change of battery state of charge and the absolute number of switches.

3.2.2 Threshold based hysteresis

This method tries to apply a *threshold zone* to the combined operating area of both EM. At first consider fig. 3.5 where the solid line corresponds to the suboptimal torque split strategy for $u_f \in \{0, 0.5\}$, which was previously defined by the suboptimal design approach in section 3.1. The dashed lines above and below the threshold line define the hysteresis bounds. The purpose of introducing a broader area around the threshold is to avoid consecutive switches caused by torque demands which lie very close to this line. An upper and lower boundary line can prevent this behavior since no switches will be allowed to occur inside this area.

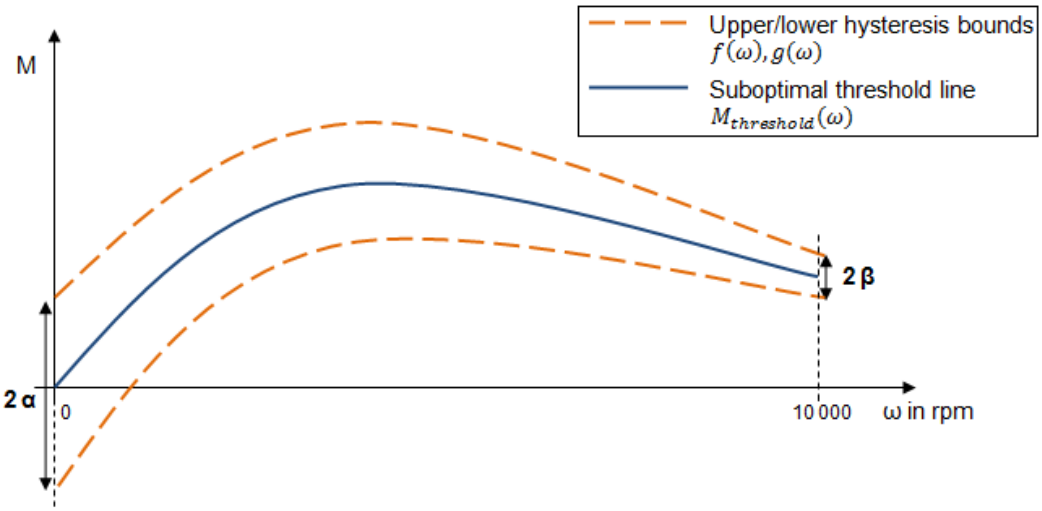

Figure 3.5: Threshold based hysteresis design approach with α and β as varying parameters to define the hysteresis bounds in the torque-speed-map

Various benchmarks of this improvement strategy will differ in the parameters α and β, which define the gap between the threshold and the border lines for EM-speed $\omega_{min} = 0\,\text{rpm}$ and $\omega_{max} = 10\,000\,\text{rpm}$, respectively. Note, that an affine linear characteristic will be used to define the upper and lower bounds for the complete EM-speed range according to eq. (3.6) and (3.7),

$$f(\omega) = \frac{-(\alpha - \beta)}{\omega_{max} - \omega_{min}} \omega + M_{threshold}(\omega) + \alpha \tag{3.6}$$

$$g(\omega) = \frac{(\alpha - \beta)}{\omega_{max} - \omega_{min}} \omega + M_{threshold}(\omega) - \alpha \tag{3.7}$$

where:

$f(\omega)$ ..... upper hysteresis line
$g(\omega)$ .....lower hysteresis line
$M_{threshold}(\omega)$ ... suboptimal threshold line

As can also be seen in fig. 3.5 the width of the threshold zone does not necessarily have to be equal for the entire operating area. It can truly be useful to apply a distribution with $\alpha > \beta$ implicating that more switches in the lower torque demand area are suppressed. In the following consider fig. 3.6 which shows the EM-operating points for several driving cycles - NEDC, FTP75 and AVL-internal test cycles which represent urban-, extraurban- and highway driving characteristics. Note, that the last three DC also include altitude profiles in their road description.

Obviously most of the torque demands lie in the lower EM-speed area, which leads to the assumption that during realistic driving most of the switching behavior happens during low speeds and should therefore give attention to an adequate adjustment of the border line especially in this area.

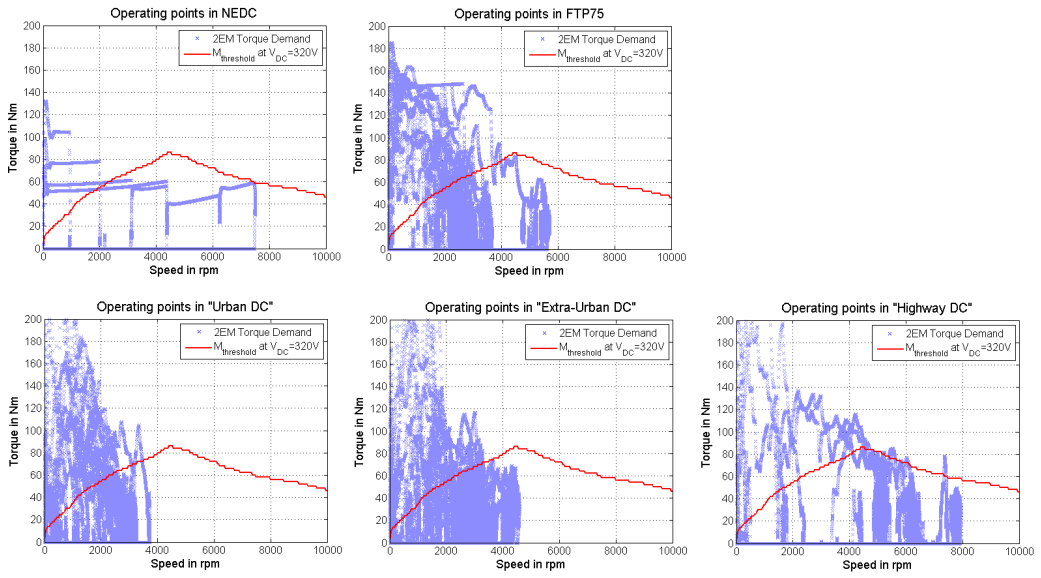

Figure 3.6: Total torque demand over EM-speed during traction for different driving cycles

Based on fig. 3.6 it can also be observed that NEDC contains rather artificial driving maneuvers where all distinct stages of the DC trace out easily distinguishable trajectories in the overall EM operating area. In contrast to NEDC, for all other remaining cycles the driving behavior is generally more complex so that nearly the complete area is covered by the simulated operating points. For this purpose also consider fig. C.1 in Appendix C where the velocity profile of each DC is illustrated.

3.2.3 Simulation and evaluation process

In chapter 2, a vehicle model of the Peugeot 3008 OpEneR project cars has been introduced. AVL CRUISE simulation environment was used to implement a complete parametrized version of this model in order to simulate the effects of longitudinal vehicle dynamics for different driving maneuvers.

During the integration phase several torque split designs based on the previously defined methods have been implemented in the model to adapt the powertrain control software in a proper way. In the following each design will be called benchmark to emphasize the possibility of comparison and further evaluation based on different selection criteria.

All developed designs are analyzed based on 5 different driving cycles:

a) NEDC,
b) FTP75,
c) AVL Urban DC,
d) AVL Extraurban DC
e) and AVL Highway DC.

In each case different system parameters are collected to define a metric which allows a further evaluation of the developed designs. On the one hand these parameters include the investigation of battery state of charge, which directly links to the consumed energy during the drive. On the other hand characteristics of the torque split behavior are needed to check if possible improvements are reached. Summarized, following parameters are collected during each simulation run:

	BM1 optimal	BM2 suboptimal	BM3 rear EM	BM4 front EM	BM5 50:50
ΔSOC (%)	3.9396	3.9403	3.9704	3.9697	4.2216
$\Delta SOC_{imp,BM5}$ (%)	6.68	6.66	5.95	5.96	-
# switches	39	28	0	0	0
# switches/s	0.033	0.023	0	0	0
min ΔT (s)	0.025	1.15	-	-	-

Table 3.1: NEDC results for basic torque split approaches

- ΔSOC (in %)
  absolute decrease of battery SOC (state of charge)
  $\Delta SOC = SOC_{initial} - SOC_{final}$
- $\Delta SOC_{imp,BM5}$ (in %)
  SOC improvement relative to ΔSOC of BM5 (50 % const. distribution)
- # switches
  total number of mode changes
- # switches/s (in $\frac{1}{s}$)
  number of mode changes per second, underlying the total DC-duration
- min ΔT (in s)
  minimum time between two switches

At first consider table 3.1, where NEDC simulation results for basic torque split designs are illustrated. BM1 and BM2 denote the optimal and suboptimal design as earlier mentioned, BM3 (BM4) implicate traction with rear (front) EM only and BM5 finally represents the results for the 50:50 constant torque split distribution.

As earlier mentioned in section 3.1, a const. 50:50 distribution yields to the worst results not only for NEDC, but also for all other driving cycles. Obviously the optimal solution gains the best (lowest) ΔSOC-value, closely followed by the suboptimal design approach which improves already by 6.66 % in terms of energy consumption compared to the previously mentioned const. distribution design. Still the number of switches is relatively low for a total NEDC duration of 1180 s, but as explained in 3.1 this driving cycle is rather artificial. Therefore it is recommended to take a closer look on the remain-

	NEDC	FTP75	Urban	Extraurban	Highway
optimal	39	273	419	458	182
suboptimal	28	128	158	230	86

Table 3.2: Number of u_f-switches for optimal and suboptimal design

ing DCs, which represent a more realistic driving behavior instead. In BM1 $min\ \Delta T$ is about 0.025 s which exactly corresponds to the simulation time limits. As can be seen from table 3.2 the amount of driving mode switches jumps instantly to very high values, when other DC are investigated. Switches in the optimal design are determined based on the discretization of possible split values so that the simulated interval equates to [0, 0.05, 0.1,.., 0.5]. Note that a possible longer DC-duration is not the primary factor for the increase of u_f-switches, it is more the driving characteristic that has an influence on it.
Although the number of switches could be reduced significantly through the use of the suboptimal design approach in BM2, $min\ \Delta T$ is still in a non-acceptable range for all DC besides NEDC.

Time based hysteresis results

For the time hysteresis case, several designs have been created by successively increasing ΔT from 0 to 30 s. Obviously the results in terms of energy consumption will get worse, the higher ΔT is chosen which directly forces the vehicle to stay at a certain - probably inefficient - driving mode. In fig. 3.7, $\Delta SOC_{imp,BM5}$ increases monotonously until ΔT reaches 0, which results in the maximum improvement value visualized in the diagram. Note, the last benchmark (BM12) equals to the *suboptimal* case (BM2). Fig. 3.8 shows how the number of switches/s change due to various designs.

In summary it can be said, that in this design approach directly reducing the amount of mode changes by setting a time limit is done at the expense of additional increase of energy consumption, which cannot be neglected at least for high values of ΔT. For further details it is referred to Appendix B, where the complete simulation results for the time based hysteresis case are listed.

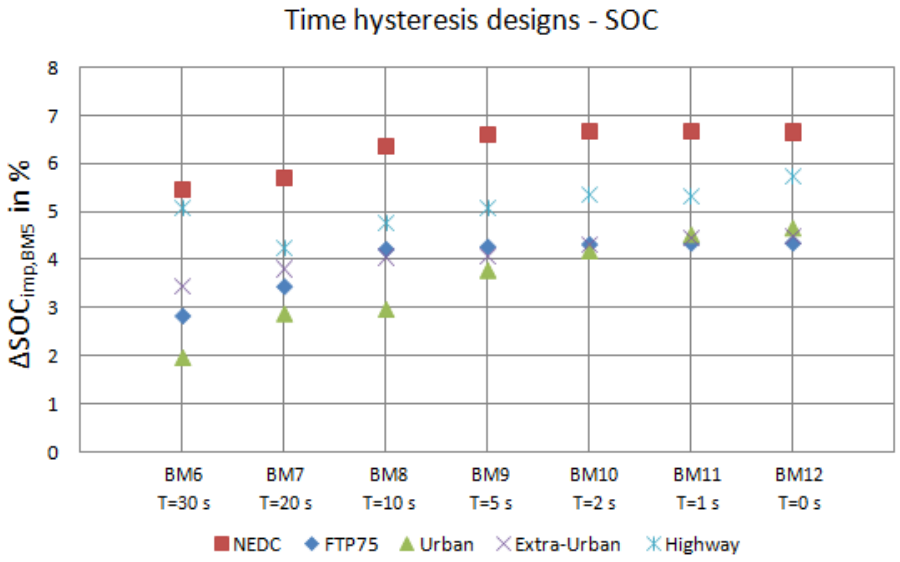

Figure 3.7: Simulation results for ΔSOC_{imp} in % relative to BM5 for several time hysteresis designs and DC

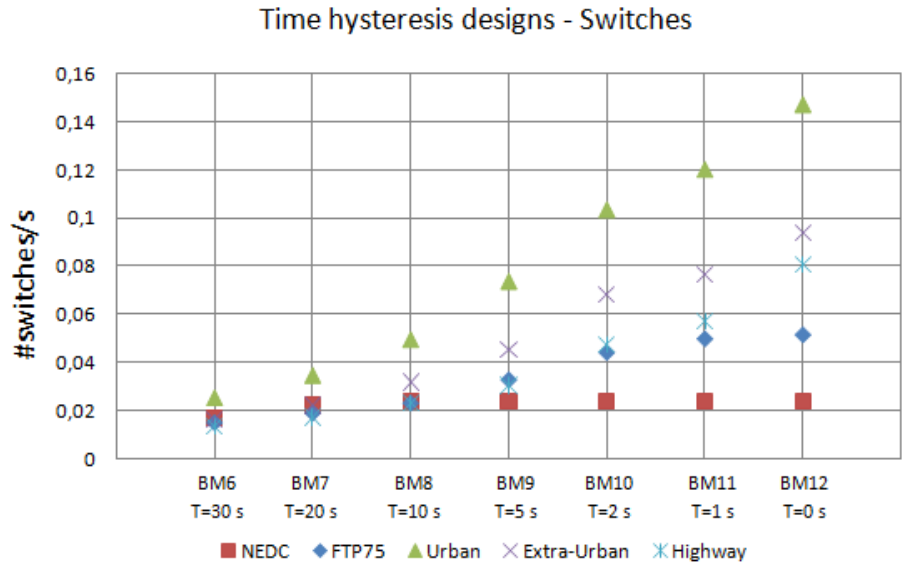

Figure 3.8: Simulation results for number of u_f-switches per second for several time hysteresis designs and DC

Threshold based hysteresis results

Like in the previous section, fig. 3.9 shows the energy improvement results compared to the 50%-constant distribution for various threshold based hysteresis designs. As can be seen from the x-axis labeling, several designs covering different possibilities to define the linear hysteresis bounds according to eq. (3.6) and (3.7) are investigated during the simulation and evaluation process. Generally, considering all listed designs the hysteresis gap defined by α and β decreases the higher the benchmark-number becomes. BM20 corresponds to the *suboptimal* case again.

One could expect that an increasing hysteresis gap negatively influences the overall efficiency and furthermore the energy consumption, which could directly be linked to a drop of ΔSOC. However, it turns out that nearly independent of various design approaches the ΔSOC characteristic stays constant for all investigated DCs. For instance, in NEDC the *suboptimal* case improves by 6.66 % compared to BM5. The design with the biggest gap (BM13: $\alpha = 40$ Nm, $\beta = 10$ Nm) reaches a value of 6.43 %, which is only negligibly worse.
Given the fact that various benchmarks yield to nearly the same ΔSOC result as the *suboptimal* case, furthermore that the constant SOC characteristic is independent on different DCs and therefore independent on the driving behavior, it is possible to focus on the number of torque-split switches and select the best design based on this criteria only.

In fig. 3.10 the number of u_f-switches divided by the total DC-duration is visualized for all investigated benchmarks and several driving cycles. Obviously the values increase when the gap of the hysteresis curve declines until $\alpha = \beta = 0$ (BM20) which represents the threshold-line in the *suboptimal* case again.

A question that might arise is how the parameter $min\ \Delta T$ changes with respect to the different designs, since in the time-hysteresis case $min\ \Delta T$ directly equals to the design parameter ΔT and does not have to be considered there explicitly. According to Appendix B where the detailed simulation results for the threshold based hysteresis case are listed, $min\ \Delta T$ varies between

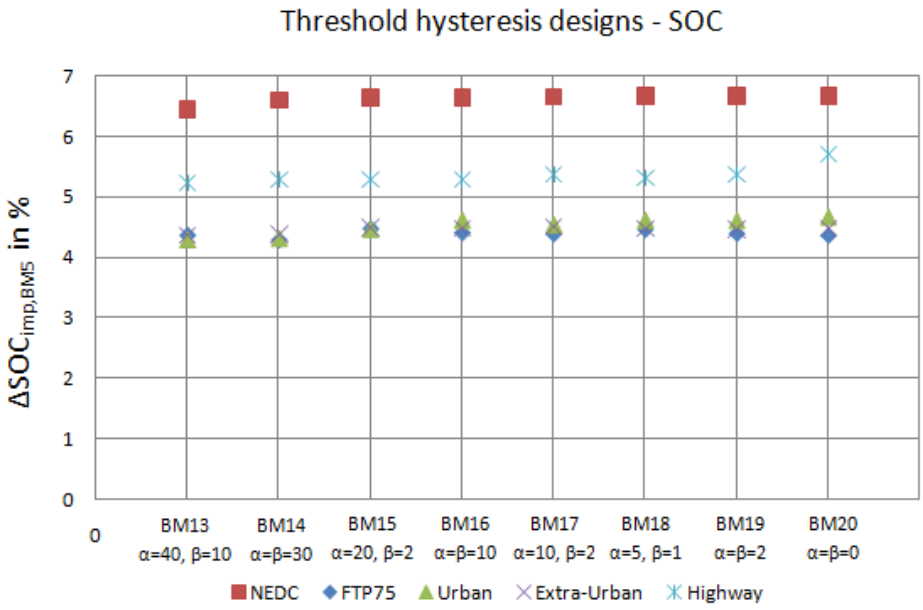

Figure 3.9: Simulation results for ΔSOC_{imp} in % relative to BM5 for several threshold based hysteresis designs and driving cycles

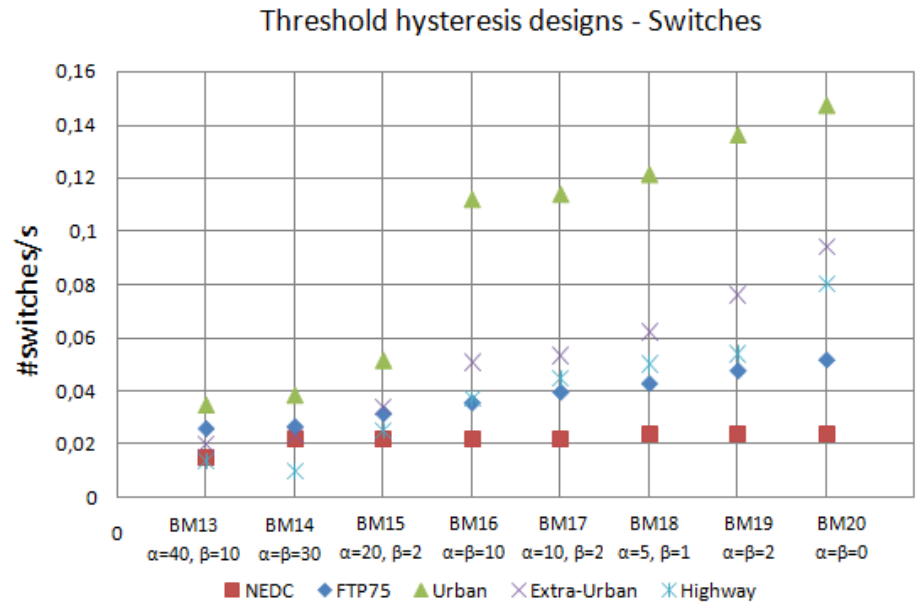

Figure 3.10: Simulation results for number of u_f-switches per second for several threshold based hysteresis designs and driving cycles

	NEDC	FTP75	Urban	Extraurban	Highway
ΔSOC (%)	3.94	7.03	2.31	7.71	11.33
$\Delta SOC_{imp,BM5}$ (%)	6.66	4.36	4.66	4.47	5.71
# switches	28	128	158	230	86
# switches/s	0.024	0.052	0.147	0.094	0.080
min ΔT (s)	1.15	0.05	0.025	0.025	0.025

Table 3.3: Complete simulation results for BM2 (suboptimal case)

	NEDC	FTP75	Urban	Extraurban	Highway
ΔSOC (%)	3.95	7.03	2.32	7.72	11.39
$\Delta SOC_{imp,BM5}$ (%)	6.43	4.36	4.29	4.37	5.22
# switches	18	64	37	49	15
# switches/s	0.015	0.026	0.034	0.020	0.014
min ΔT (s)	25.9	1.4	1.4	1.6	2
Drop of switches (%)	-36	-50	-77	-79	-83

Table 3.4: Complete simulation results for BM13 ($\alpha = 40$ Nm, $\beta = 10$ Nm) including drop of driving mode switches compared to the *suboptimal* design.

the simulation time limits and about 2 s depending on the DC, respectively (in NEDC much higher). At least for designs with rather broad hysteresis curves, values generally do not fall below 1 s which should be sufficient for most applications. In any other case further restrictions can be easily implemented using additional time hysteresis design approaches.

Conclusion

Since the overall amount of EM-switches during the drive should be kept to a minimum, *benchmark 13* with the design parameters $\alpha = 40$ Nm and $\beta = 10$ Nm is the best choice for the direct torque split calibration method.

Comparing fig. 3.9, BM13 reaches ΔSOC-values close to the *suboptimal* case for all different DCs which have been investigated during the evaluation process. Hence, the corresponding improvement results for $\Delta SOC_{imp,BM5}$ reach almost *suboptimal* values as well.
Furthermore, the number of torque-split switches can be reduced up to 83 %

compared to the *suboptimal* case (cf. also table 3.4 where the complete results for BM13 are listed). For instance, according to the suboptimal results in table 3.3 BM13 improves by 6.43 % compared to 6.66 % (suboptimal) in terms of energy consumption for NEDC and thus, BM13 is only insignificantly worse. Instead of this, if the design is applied to the powertrain control software high benefits can be achieved from the loss of driving mode changes, since in Extraurban-DC, for example, the number of switches could be decreased from 230 to 49 (-79 %).
The last evaluation parameter $min\ \Delta T$, which denotes the smallest time span between two different driving modes, reaches satisfying values for various DCs (2 s up to 26 s for NEDC) and can further be increased by an additional time hysteresis design if needed.

Summarized, after comparing the results between the suboptimal design (BM2) and the best fitted hysteresis design (BM13) the objectives for various improvements regarding the practical applicability of an optimum torque split strategy could be proved by simulations. Furthermore, the presented results fulfill all expectations towards high energy efficiency for different driving cycles.

3.3 Predictive calibration using ANN

This method tries to focus especially on the second issue introduced in section 3.1 (problem definition), where consecutive changes of the driving mode result in probably higher energy consumption due to time lags of certain preparation and EM-synchronization phases. Especially for closely arranged switches as it happens for the ordinary torque split algorithm, energy benefits of changing to every optimal/suboptimal mode might not occur or would rather turn into additional costs instead.

An appropriate algorithm could predict the short time behavior of the driver and measure the probability of oscillating switches for the next few time steps. This *prediction value* could be used in the subsequent powertrain control software to avoid unnecessary switches and improve the overall energy efficiency in the end. Furthermore, durability and comfort issues are taken into account as well since the amount of driving mode switches will also decrease within this approach.

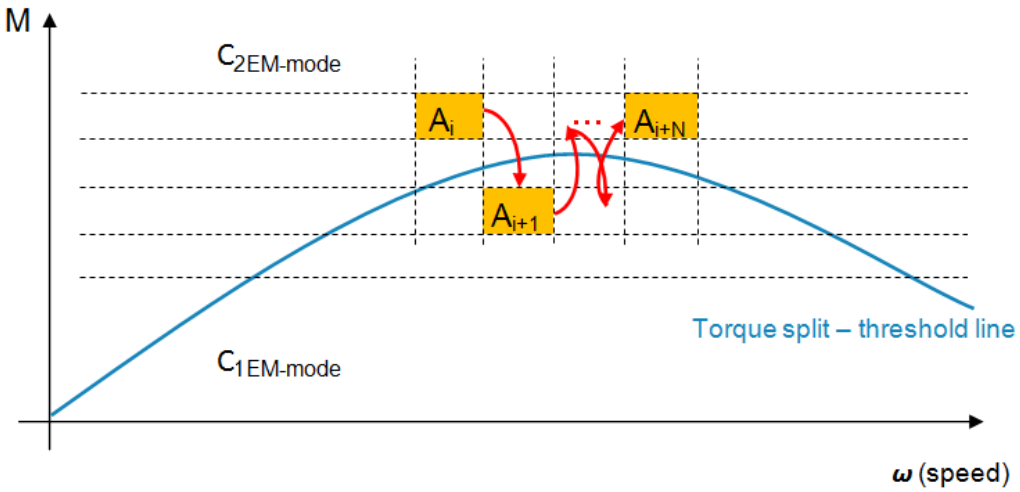

Figure 3.11: Approach for driving mode prediction. Classes $C_{1EM-mode}$ and $C_{2EM-mode}$ represent u_f values of 0 and 0.5, respectively. Arrows denote a possible driving mode trajectory over time t, where $t = i, i+1, .., i+N$

Consider fig. 3.11, where the driving mode trajectory indicated by consecutive arrows describes the following sequence: Depending on the correspondent

driver inputs at time (i) the current operation point lies in area A_i. In the next time step $(i+1)$ the powertrain control software recognizes an upcoming torque-split change since the transition from A_i to A_{i+1} implies crossing the suboptimal u_f-threshold line. At this point the question arises, whether it is useful to perform the desired switch or to neglect it with respect to an upcoming oscillating behavior and a "collectively" jump back to the 2-EM-mode area ending up at time stamp $(i+N)$, which completes the considered time interval in this approach.
Note, that primely this question brings up the need for the development of a *predictive calibration* algorithm.

According to eq. (3.10) "collectively" means the average membership of operating points belonging to the class $\mathrm{C}_{1\mathrm{EM-mode}}$ and $\mathrm{C}_{2\mathrm{EM-mode}}$, respectively, during the future-interval of N time steps. Consider eq. (3.8)-(3.10) for the mathematical expression of $\bar{\mathrm{A}}_N$.

$$\alpha = \sum_{j=i+1}^{i+N} \xi_j \qquad \xi_j = \begin{cases} 1 & A_j \in \mathrm{C}_{1\mathrm{EM-mode}} \\ 0 & A_j \in \mathrm{C}_{2\mathrm{EM-mode}} \end{cases} \tag{3.8}$$

$$\beta = N - \alpha \tag{3.9}$$

Then, $\bar{\mathrm{A}}_N$ can be defined as

$$\bar{\mathrm{A}}_N \in \begin{cases} \mathrm{C}_{1\mathrm{EM-mode}} & \text{for } \alpha > \beta \\ \mathrm{C}_{2\mathrm{EM-mode}} & \text{otherwise.} \end{cases} \tag{3.10}$$

Every time a change of the current driving mode is requested (as it is the case for $t = i + 1$ in the example), the probability of coming back to the original state in average during the short time period of N samples is needed for the further decision process. Hence, an appropriate algorithm has to calculate a prediction value which estimates the conditional probability of an upcoming oscillating behavior due to future driver inputs based on various information provided to the algorithm.

See eq. (3.11) for the prediction value definition in the case of fig. 3.11. A switch in the other direction can be derived similarly replacing class $C_{1EM-mode}$ with $C_{2EM-mode}$ and vice versa.

$$\delta_p := 1 - P\left(\bar{A}_N \in C_{2EM-mode} \,|\, A_i \in C_{2EM-mode},\, A_{i+1} \in C_{1EM-mode}\right) \quad (3.11)$$

where $\delta_p \in [0..1]$ is the output of an appropriate prediction algorithm and reflects the torque split switching recommendation:

$\delta_p = 0$ .... switch not recommended
100 % probability of jumping back instantaneously to $C_{2EM-mode}$.
$\delta_p = 1$ .... switch recommended
0 % probability of jumping back to $C_{2EM-mode}$.

It is clear that any forecasting methodology has the need for adequate information as an input to its algorithm in order to produce feasible outcomes. For this approach, both driver commands and actual powertrain states based on statistical examinations as well as information about the current/upcoming road conditions will serve as meaningful inputs to the prediction learning process.
If other sensing devices were provided by the vehicle, further data sources could consist of road related information (e.g. hills, curves) delivered by GPS-signals, current traffic information through radar sensors and car2car-communication systems, or data of upcoming speed limits provided by car2-infrastructure-communication.[3]

In the next section "Artificial Neural Networks" (ANNs) as a common and convenient possibility to deal with forecasting problems, especially with subject to pattern recognition issues, will be introduced.

[3] cf. correspondent patent "Driver behavior pattern recognition by using past behaviors and environmental sensing", AVL Graz

3.3.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) are generally adapted from the information processes going on in human brains. About 10^{11} special cells called *neurons* are provided with each 10^4 inputs coming from other cells connected via synapses in order to enable information exchange. The output of a human neuron is called axon and "fires" if certain thresholds are reached [Maa08; Haj]. See also fig. 3.12 which visualizes the principle structure of a biological neuron.

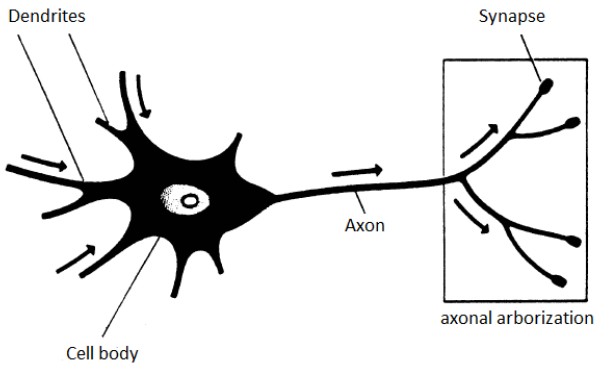

Figure 3.12: Schematic drawing of a biological neuron adapted from [Haj]. Arrows denote the information flow.

In general, ANNs form a parametrized model in order to adapt to arbitrarily defined continuous functions usually provided by training data consisting of inputs and targets in case of *supervised learning*. A training set usually contains plenty of data points describing the most important features for the learning task, each input combined with a target value to define the desired output of the ANN. Parameter values of the network refer to the *weights* each connection is characterized by and during training process these parameters are adapted based on a certain learning algorithm, e.g. *perceptron learning rule* or the *backprop*-algorithm introduced in section 3.3.1.3).

ANNs are widely used in different application areas [Maa08], such as:

- *Classification, Pattern Recognition*
  system identification, control systems (vehicle control), face and typography recognition, medical diagnosis, data mining, etc.

- *Prediction*
  financial applications, driving behavior estimation, extrapolation based on historical data, approximation of functions
- *Noise Reduction, Filter*
  recognize noisy data and produce meaningful outputs

Various properties as well as advantages and disadvantages of neural networks [Haj] can be listed as:

+ Ability to learn:
  Connecting simple elements generates an overall complex behavior which makes it possible to adapt to arbitrarily functions. The input-output behavior is characterized by the adjusted weights.
+ Ability to generalize:
  ANNs are said to generalize well when the input-output relationship computed by the network generates reasonable outputs for data not encountered during training process.
+ Distributed and parallel computing:
  Robust against errors (failure of neurons if implemented in hardware) since the performance degrades gracefully rather than ending up instantly in a catastrophe.
+ Good performance
+ Small memory requirements
− Computing time:
  Relatively high during training, but typically fast for testing (even for many test samples)
− Risk of overfitting:
  For complex networks the risk of over-training increases dramatically, which means that the network performs well on the training set, however generalizes poorly.
− Intuitive interpretability:
  Rather bad since ANNs can be seen as nonlinear black box models with high complexity inside.

The simplest form of neural network is called *perceptron* consisting of one single neuron together with a hard limit threshold gate and is used for linear

classification problems. The most successful model for ANNs in the case of pattern recognition is the *feed-forward architecture*, also known as the multi-layer perceptron which will be discussed in section 3.3.1.2 [Bis06].

3.3.1.1 Perceptron

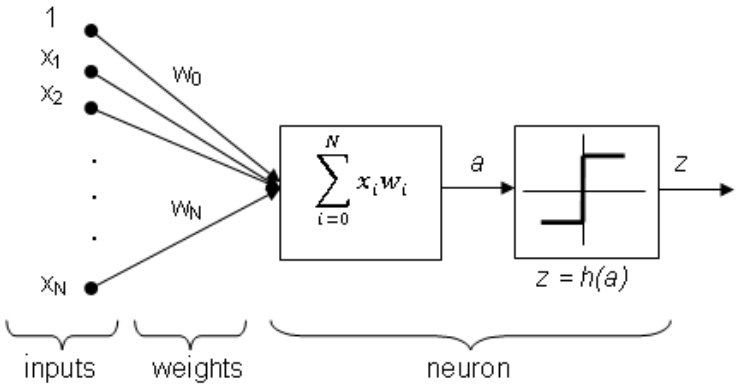

Figure 3.13: Mathematical structure of a single-layer perceptron

The single-layer perceptron shown in fig. 3.13 is provided with N inputs and an additional bias $x_0 = 1$ to support any possible linear combination of the inputs plus an additive term. The weighted sum a of all inputs is transferred to an *activation function* $h(a)$, which denotes a hard limit threshold gate in the case of perceptrons so that the output z of the neuron *always* corresponds to a binary value. Training the perceptron means to adjust all weights w_i for $i = 0, .., N$ so that the output z corresponds to the desired target specified in the training set. Consider the *perceptron learning rule* in eq. (3.12), where t indicates a specific target value and α denotes the so called learning rate. Note that all weights and inputs have been replaced by vectors $\mathbf{w}$ and $\mathbf{x}$.

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \alpha\,(t - z) \cdot \mathbf{x} \tag{3.12}$$

According to the perceptron convergence theorem, it is proved that for linear separable training data sets the perceptron learning rule in eq. (3.12) con-

verges and generates a decision surface in form of a hyperplane between the classes [Haj, p. 23].

Note, that for perceptrons the activation function is always a threshold gate, which is not differentiable and therefore has to be replaced by appropriate functions in case of more complex networks with a different learning algorithm.

3.3.1.2 Feed Forward Architecture

According to the literature there exist several possibilities how to connect perceptrons to a complex network. Without considering *recurrent* ANNs, *sparse* networks or included *skip-layer* connections, in this thesis the focus lies on dense *layered feed forward networks* as shown in fig. 3.14. The first or lowest layer is called *input layer*, which receives external information to define the network inputs. The last layer consists of several output neurons, depending on the particular use case the network is designed for. For instance a classification task with $K > 2$ classes is commonly described by K output neurons to gain a maximum distance between the classes that have to be distinguished [Maa08]. The input and output layers are separated by one or more intermediate layers called *hidden layers*. Characteristics for this type of ANN are:

- The network can be described as a directed graph $G = (V, E)$ with nodes V (inputs, hidden neurons, outputs) and edges E.
- For every edge (i,j) $\in$ E from node i to node j in layer l there exists a weight $w_{ji}^{(l)}$
- For every node except input nodes there exists an activation function $h^{(l)}(\cdot)$, which equals for all nodes in layer l.

The overall input to each neuron in fig. 3.14 is called *activation* and can be expressed as the weighted sum of all outputs from the previous layer [Bis06; Maa08]. The activation for node j equates to:

$$a_j = \sum_{i=0}^{N} w_{ji}\, z_i \tag{3.13}$$

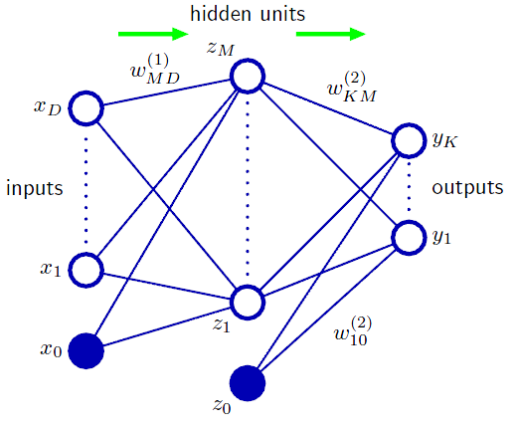

Figure 3.14: Structure of a 2-layer feed forward network with D inputs, M hidden units in the 1^{st} layer and K output neurons in the 2^{nd} layer. Bias parameters are represented by links coming from additional nodes x_0 and z_0. Arrows denote the information flow from left to right during forward propagation [Bis06].

where:

$N = \{\text{D, M}\}$ is the number of neurons in the previous layer, and

$$z_i = \begin{cases} x_i & \text{when node i corresponds to an input node} \\ h(a_i) & \text{otherwise.} \end{cases}$$

The output of a neuron (except input layer) is transformed using a differentiable activation function $h(\cdot)$, which usually denotes linear-, *tanh*-functions or the commonly used sigmoid function (in Matlab: logsig, see also fig. 3.15).

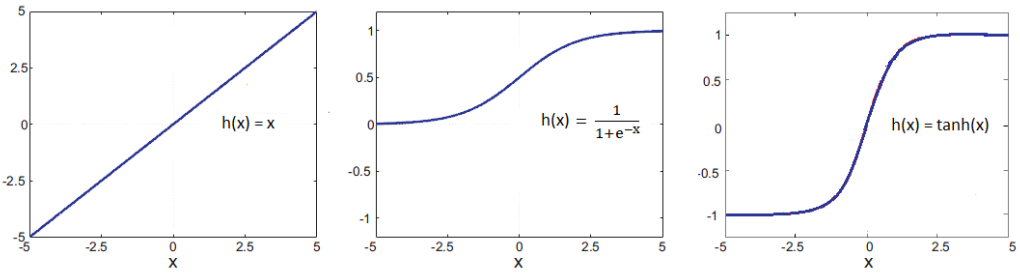

Figure 3.15: Commonly used activation functions for neural networks (linear, sigmoid, tanh)

Combining all stages the output of neuron k for a 2 layered feed forward network can be calculated as expressed in eq.(3.14). Note, that all weights and bias value have been grouped to the vector $\mathbf{w}$.

$$y_k(\mathbf{x}, \mathbf{w}) = h^{(2)} \left(\sum_{j=0}^{M} w_{kj}^{(2)} h^{(1)} \left(\sum_{i=0}^{D} w_{ji}^{(1)} x_i \right) \right) \quad \text{for } k = 0, 1, .., K \tag{3.14}$$

Remarks:
It is clear that if all activation functions in the ANN are linear, then the output y_k will be a linear function of the inputs again. In this case there can always be found an equivalent network without any hidden layers.
Furthermore it has been proved that a single hidden layer network like in fig. 3.14 is sufficient to approximate any continuous function arbitrarily exact. This is also called as the *universal approximation theorem* and can be referred in [Bis06; Maa08; Haj].

3.3.1.3 Error backpropagation

According to the literature, Error backpropagation or simply *back-prop* is an algorithm to calculate the gradient of the error function in an efficient way. It was introduced by Paul Werbos in his PhD. thesis in 1972, but did not get popular until the 80's when other authors "reinvented" this algorithm [Maa08]. It is characterized by two distinct stages denoting the pass of messages through the network:

Forward propagation
It means to pass information from the input layer over all hidden nodes to the output layer and to collect the results of each neuron according to eq. (3.14).
Backward propagation
At first derivatives of the error function with respect to the weights are calculated for the complete output layer. Afterwards those signals are propagated backwards through the network to assign the corresponding derivatives to all remaining nodes [Bis06].

Back-prop is designed for feed-forward networks with differentiable activation functions and a class of error functions which can be expressed as the sum over all training examples L:

$$E(\mathbf{w}) = \sum_{n=1}^{L} E_n(\mathbf{w}) \tag{3.15}$$

The goal is to calculate the local gradient with respect to a certain weight w_{ji} for a single training input n:

$$\nabla E_n = \frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}} \equiv \delta_j z_i \tag{3.16}$$

The second term corresponds to the output of neuron i as in eq. (3.13) expressed. For the first term the definition δ_j is used to define the local error at each node [Bis06] which will be more clear when the error for an **output neuron** is derived (with t_k as target value for neuron k):

$$E_n = \frac{1}{2} \sum_{k=1}^{K} (y_k - t_k)^2 = \frac{1}{2} \sum_{k=1}^{K} (h(a_k) - t_k)^2 \tag{3.17}$$

$$\delta_k = \frac{\partial E_n}{\partial a_k} = (y_k - t_k) \qquad \text{for linear } h(a) = a \tag{3.18}$$

The error δ_j for **hidden neurons** can be calculated using the information from previous nodes:

$$\begin{aligned} \delta_j = \frac{\partial E_n}{\partial a_j} &= \sum_{k \in post(j)} \frac{\partial E_n}{\partial a_k} \frac{\partial a_k}{\partial a_j} = \sum_{k \in post(j)} \delta_k \frac{\partial a_k}{\partial a_j} = \\ &= \sum_{k \in post(j)} \delta_k \frac{\partial}{\partial a_j} \left(\sum_{i \in prev(k)} w_{ki} \, z_i \right) = \\ &= \sum_{k \in post(j)} \delta_k \sum_{i \in prev(k)} w_{ki} \frac{\partial z_i}{\partial a_j} \end{aligned}$$

Given the fact that $z_i = h(a_i)$, what means that $\frac{\partial z_i}{\partial a_j} = 0$ for $j \neq i$, the error

for a hidden node in layer (l) equates to:

$$\delta_j = \frac{\partial h^{(l)}(a_j)}{\partial a_j} \sum_{k \in post(j)} \delta_k \, w_{kj} \tag{3.19}$$

Now, a simple gradient descent based method with learning rate η can be used to update the weights in order to train the network with respect to a decreasing error function [Bis06; Maa08]:

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta \, \nabla E(\mathbf{w}^t) \tag{3.20}$$

3.3.2 Modeling and training

In section 3.3 the problem of recognizing driving behavior patterns together with a prediction task has been stated, which forms the basis for the development of a forecasting neural network. Since the modeling and training of ANNs has a high impact on the overall performance, it is crucial to focus on the selection of an appropriate network architecture at first. According to the composition of feed-forward networks this implies the decision on the number of layers as well as the number of nodes in each layer.
Further issues contain the choice of activation function, efficient training algorithm, the selection of training and test sets together with data normalization and a suitable error function for performance measure [ZHP]. In this section all above mentioned issues about modeling ANNs in general as well as for the driving behavior prediction task will be successively investigated.

Architecture

Defining the architecture of an ANN means to determine the number of neurons for each layer. According to the literature this task is non-trivial and the resulting layout mainly depends on the application of the ANN. Furthermore there exists no optimal solution for determining the number of neurons, hence, the design of an ANN is more an "art" than a matter of science. Guidelines

are either heuristics or based on several trial and error simulations so that the network reaching the best results is selected [ZHP].

The **input layer** should consist of a small number of essential nodes, which describe the underlying features in the data as significantly as possible. Too few or too many inputs can affect the learning and generalization capability of the network in a bad way [ZHP].
For the given problem (cf. fig. 3.11) it is useful to consider the state of the vehicle at time $t = i, i + 1$ and define the inputs based on this time frame. The acceleration pedal position of the driver as well as current torque demand and EM-speed reflect the most important characteristics of the actual driving state at the specific time when a mode change is requested ($t = i + 1$). Relations to the last state ($t = i$) are considered to improve the overall learning and are therefore added to the input layer. Furthermore, information covering the road/traffic conditions should be mapped to the inputs as well since the future behavior of the driver might be dependent on changes of the environment. Hence, data about the current road inclination which could come from an integrated GPS sensor device completes the number of inputs for the prediction task.

However, determining the number of nodes in the **output layer** is rather simple instead. According to the definition of the prediction value δ_p in eq. (3.11), only 1 single output neuron is needed to indicate if a requested mode change should be performed or not.

The choice of nodes for the **hidden layer(s)** is one of the most crucial aspects for designing an ANN, since only hidden neurons have the capability of recognizing any concealed pattern in the data and to perform nonlinear approximations to reach the desired input/output mapping. It is unclear how many layers are considered to be sufficient for a forecasting problem, since several authors propose either a single-hidden-layer network or ANNs with two hidden layers. In general, more than two hidden layers are never applied to neural networks because they tend to overfit (bad generalization), which is a common problem for networks with rather high numbers of hidden nodes [ZHP; Bis06].
For the defined prediction task several simulations with different network

layouts showed that a single-hidden-layered ANN with 20 neurons reached quite as good results as more complicated networks. Hence, and according to Occam's Razor:

"If possible, always prefer the simplest hypothesis",

the 2-layer feed-forward network as illustrated in fig. 3.16 was selected during the decision process.

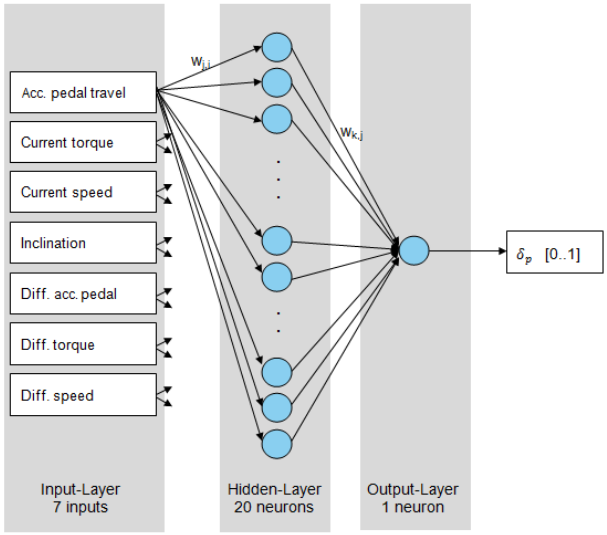

Figure 3.16: Applied 2-layer feed-forward ANN to deal with the proposed driving behavior prediction task. Inputs correspond to various vehicle and powertrain states as well as road conditions assigned when changes of the driving mode emerge. The last three inputs indicate the differences to the former driving state. The output δ_p of the network denotes the prediction value defined in eq. (3.11).

Activation function

As already mentioned in section 3.3.1.2, activation functions are used to translate the weighted sum of inputs to a bounded output value depending on the characteristics of this function. In general, those functions are responsible for introducing nonlinearities to the network which makes it valuable for most applications. Although it is important to use differentiable functions with respect to the training process, it is not clear which activation function should

be used in which case or if different functions have an impact on the performance of the network[ZHP; Bis06].
Several authors simply suggest the use of sigmoid (logistic) activation functions for both hidden and output layers. Furthermore, the addressed function seems suitable especially for output nodes in case of binary classification problems [ZHP].
To follow the convention for the given task, all nodes will be equipped with sigmoid activation functions.

Performance measure and training

Training a neural network is an unconstrained nonlinear parameter optimization problem, where the weights $\mathbf{w}$ characterizing the connections between the nodes are adjusted in a way so that the error function $E(\mathbf{w})$ reaches a minimum. Usually either the total squared error or the mean squared error (MSE) for all output nodes over all input data is used for the training and validation process [ZHP]. For the specific driving behavior prediction task, MSE as a performance measure over all training examples $n = (1, .., L)$ with targets t_n was selected according to eq. (3.21).

$$MSE = E(\mathbf{w}) = \frac{1}{L} \sum_{n=1}^{L} (\delta_p(\mathbf{w}) - t_n)^2 \tag{3.21}$$

Since there is no possibility to find a closed formula for solving this optimization problem, an iterative procedure is applied which updates the weights in each step (epoch). Unfortunately there is no guarantee to find the global optimum, which means that all algorithms generally suffer from the local minimum stuck problem [ZHP; Maa08]. In order to find a sufficient solution, several training cycles with different initial weights are performed to select the best fitted network in the end.
Generally, combining Error-backprop with a pure gradient descent algorithm results in slow convergence, inefficiency and weak robustness [ZHP]. Although there exist methods which can improve the behavior (e.g. adaptive learn rate,

momentum term, weight decay, early stopping[4]), other training algorithms usually achieve significantly better results and a faster convergence.
For the proposed prediction task, a scaled conjugate gradient method (in Matlab: trainSCG) is used to train the designed network in an efficient way. Compared to the ordinary algorithm, SCG does not update the weights in gradient descent direction, rather using conjugate directions instead.[5]

Data sets

For training and testing of ANNs typically two different data sets are used. If possible, the training set comprises a large amount of samples to cover all feasible inputs which could occur during the application of the network. Generally, as for any statistical approach the accuracy of an ANN increases the more samples are provided to an ANN during training. However, the sample size is limited in reality due to restricted availability of data.
On the other hand, the test set usually consists of only a few samples which are used for measuring the performance of the network (estimating the "true error" according to the literature). Sometimes a third validation set is used to avoid overfitting problems and to perform model selection. It is important that all data sets are disjoint to each other - cf. eq. (3.22) [Maa08; ZHP].

$$\text{Training-data} \cap \text{Test-data} = \emptyset \tag{3.22}$$

In case of the driving behavior prediction task each individual DC served as a basis for the generation of data samples. Only for NEDC it was not useful to prepare any data since driving mode changes do not tend to oscillate that much, also $min\ \Delta T$ is greater than 1 s in NEDC which makes the prediction task meaningless.
Furthermore it was found out that an appropriate selection of input data has a high impact on the performance of the ANN as well. Since the distribution of the switching recommendation is not equal for various DC in general (ca. 70 % samples for $\delta_{p,target} = 1$), it is necessary to limit all input samples to

[4] Cf. [Maa08; ZHP; Bis06] for further details on various improvement methods for ANNs.
[5] http://www.mathworks.de/de/help/nnet/ref/trainscg.html (03/21/2013)

the same amount for each class. Otherwise an undesired behavior appears, so that the network tends to prefer a certain class or in the worst case totally neglects one class in advance.

Data normalization

Referring several authors in ANN research area, data normalization is often performed before the training process begins. Advantages include the possibility to avoid computational problems and to facilitate the overall network learning. Furthermore benefits in terms of the classification rate and MSE could be measured [ZHP].
For the given prediction task all inputs have been normalized separately based on a linear transformation ("along channel normalization"):

$$x_{norm} = \frac{x_0 - x_{min}}{x_{max} - x_{min}} \tag{3.23}$$

where x_0 is a single input value for a specific training example, x_{min} and x_{max} are the minimum and maximum value of the corresponding channel (e.g. all pedal travel values), and x_{norm} is the normalized version of x_0.

Results

In order to perform realistic testing as far as possible, it is useful to divide the existing data sets in a smart way to show the effects separately for different DC. For example, in case of testing the ANN on DC_x, it is reasonable to train the network on all other DC without DC_x to achieve fair and meaningful results. Moreover eq. (3.22) claiming disjunct training and test sets is guaranteed here.

Consider fig. 3.17, which illustrates a single training cycle of the network designed for the driving behavior prediction task. Typical for ANN-learning in general, the MSE on training and test set decreases as the number of epochs rises. At some point the network is considered as best fitted when the MSE starts growing again. This indicates a potential overfitting process which can be avoided if the method of "Early stopping" is applied. It means

that training is stopped at the point where the MSE on the test set (or a dedicated validation set) reaches a minimum.

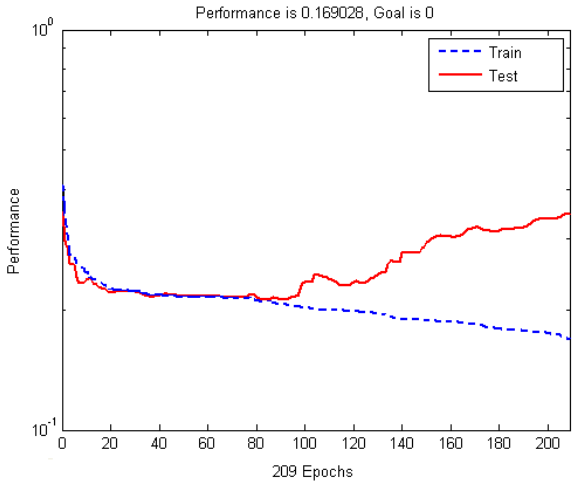

Figure 3.17: Typical ANN learning process. Here, the corresponding training set consists of FTP75-, Urban- and Highway data samples, test was performed on Extraurban-DC. Early stopping point at 91 epochs.

In table 3.5 the final classification rates of the designed ANN are presented for various DC. Note, that these results denote the percentage of correct predictions for test sets formed by each DC individually.

	FTP75	Urban	Extraurban	Highway
Correct prediction	69 %	73 %	75 %	78 %

Table 3.5: Classification rates for test sets indicated by the correspondent DC. Training was applied by providing data of the remaining cycles, respectively.

3.3.3 Integration and simulation results

For demonstration of the EM-synchronization effects described in section 3.3 together with real vehicle simulations, the powertrain model in AVL CRUISE had to be adapted accordingly. Note that due to these changes, results of the time- and threshold hysteresis benchmarks cannot be directly compared to the results presented in this section. However, BM13 as the best design for the direct calibration method will serve again for simulations with the updated model to ensure final comparisons between both introduced calibration approaches. It is important to say that the selection process in section 3.2 is not becoming void thereby, since the updated model would not have any impact on the benefits an implementation of hysteresis effects would bring with it.

The upper diagram in fig. 3.18 illustrates an undesired EM synchronization process which generally occurs if two mode changes are requested within a very short time frame. The green curve denotes the rear EM speed, which is proportional to the vehicle speed since the rear clutch is always closed during driving. At first the torque split factor u_f equals 0 which implies that the front clutch is open during this time, so that the corresponding EM speed (black line) decreases steadily due to various drag forces appearing at the motor. At the point where u_f changes to 0.5, the front EM has to be synchronized with the current rear EM speed to avoid any adverse transitions at the front clutch. This behavior is also indicated by the associated control variable (red line), which suddenly jumps to a high value to initiate the requested synchronization process.

It is clear that such preparation phases lasting for around 500 ms would have a negative impact on the overall energy efficiency, if the next switching request follows almost instantly as it is the case in fig. 3.18. Since the front clutch can not be closed until time T_x, the overall optimality would be comprised due to additional lags caused by the synchronization phase. Summarized, despite the fact that u_f always indicates an optimal switch, it is not reasonable to follow this request at any time.

The predictive ANN designed in the previous section is able to detect such un-

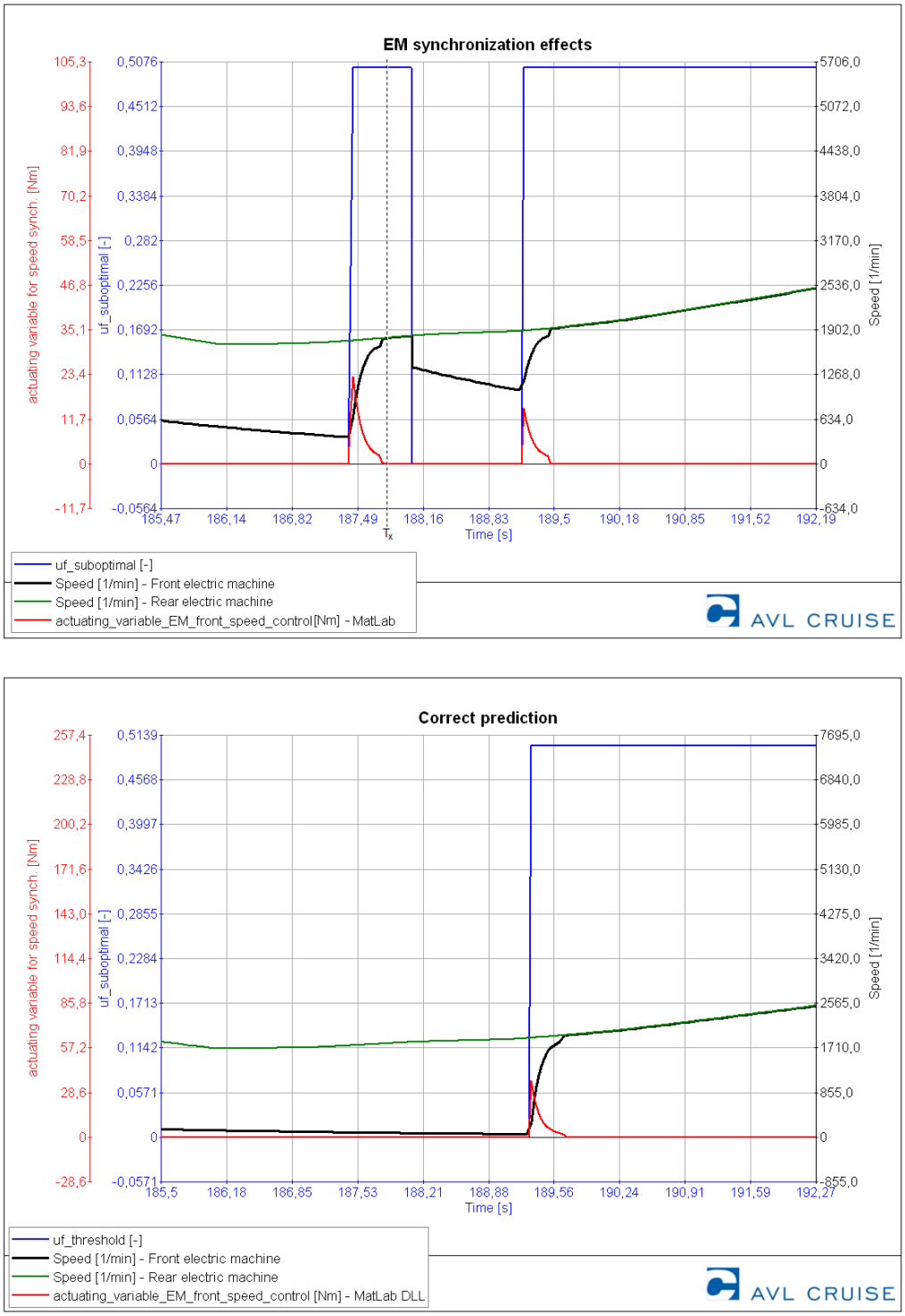

Figure 3.18: Above: Several powertrain states for the suboptimal design in a short time frame of FTP75 to show required synchronization phases when changes of the torque split factor occur. Note the front clutch cannot be closed until time T_x. Below: Improved behavior for the corresponding time frame if a correct prediction value is assumed. Energy improvement ≈ 2 %; two switches avoided.

desired switches especially to prevent oscillating mode changes. The improved driving behavior with respect to a correct prediction value is visualized in the lower diagram of fig. 3.18.
Simulations in AVL CRUISE have shown that the amount of energy ΔE_{imp}, which can be saved in this example having one unnecessary switch, is about 2 % in terms of ΔSOC.

Choice of an optimum prediction interval

In the beginning of this section it was defined, that the prediction value δ_p is estimated for the future driving behavior over a dedicated time interval of N steps. Thus, it is important to figure out if certain time spans should be preferred since they would probably have an impact on the prediction accuracy especially with respect to the total energy consumption.

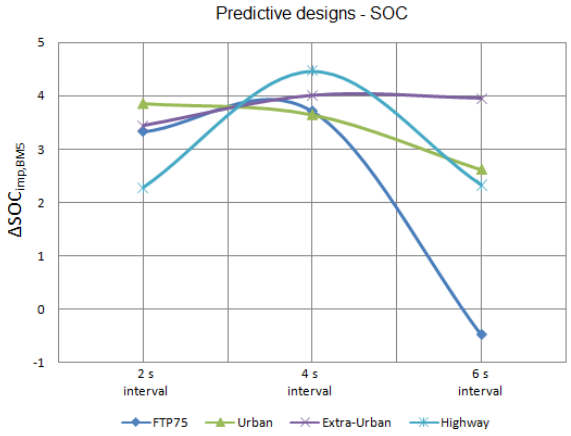

Figure 3.19: ΔSOC improvement against BM5 (const. distribution) for different prediction designs to select the optimum time interval.

From the logical point of view it does not make sense to choose very short prediction intervals since possible benefits resulting from avoided synchronizations would clearly diminish if upcoming switches are close to appear. On the other hand too long intervals would compromise the overall energy efficiency especially for wrong predictions, because the system would have to stay at non-optimal operating points for a rather long time.

Cf. fig. 3.19 which shows the energy improvement results for AVL CRUISE simulations on real-life DC with different prediction intervals. It was found out that a time span of 4 s could achieve the best results.

Simulation results

	Subopt. (BM2)	Prediction (4s)	Perfect prediction
ΔSOC (%)	7.771	7.749	7.735
$\Delta SOC_{imp,BM5}$ (%)	3.74	4.01	4.18
# switches	336	82	60
min ΔT (s)	0.025	4	4

Table 3.6: Simulation results for Extraurban-DC considering the suboptimal and predictive design (with ≈ 75 % classification rate) compared with a theoretical design where all predictions are defined to be correct.

In general, all expectations regarding the implementation of a predictive torque split control algorithm have been widely fulfilled, both in terms of energy consumption and the reduction of the number of unnecessary driving mode switches. It is clear that a forecasting algorithm always relies on its capability of producing accurate and correct predictions, and especially in case of the presented task consecutively wrong decisions would lead to a relatively high decline of energy efficiency.

As a consequence the improvement parameter $\Delta SOC_{imp,BM5}$ deviates around the values for the suboptimal design approach. If the majority of all predictions were correct, this algorithm was able to achieve quite large benefits due to the avoidance of unnecessary switches.
Consider table 3.6 which shows the simulation results for the Extraurban DC. All designs can reach large improvements compared to BM5 (constant distribution). Also $min\ \Delta T$ increases for the prediction case due to the correspondent implementation of the algorithm.
Given the fact that synchronization effects are added to the simulation model, the suboptimal design achieves rather moderate results. In contrast, a (theoretical) perfect prediction algorithm with 100 % classification rate is able to avoid every unnecessary change of the driving mode. Thus, the corresponding

number reduces from 336 to 60 essential switches and $\Delta\text{SOC}_{imp,BM5}$ improves up to 12 % compared to the suboptimal design.
In the best case results of the implemented prediction design with a 70 % classification rate typically lie somewhere in the middle, which means that energy benefits are reached due to the improved control strategy, however, possible wrong decisions also compromise the overall optimality. Cf. Appendix B for the complete table of results.

3.4 Conclusion

In the first part of this chapter two approaches as a possibility to deal with the torque split calibration problem have been introduced, each of them using hysteresis effects to improve the overall control strategy. It turned out that a simple time restrictive algorithm clearly reduces the number of switches and also increases the interval between them, however, those benefits are largely reached at the expense of energy efficiency which prevents a further application of this method.

In the threshold-based hysteresis approach one particular design was selected which achieved the best simulation results for the considered metrics. Both in terms of energy improvement as well as for the reduce of driving mode switches this design could fulfill most of the expectations towards potential practical applicability and a more safe and comfort driving style.

The last method tried to investigate if a possible prediction of the future driving behavior was able to prevent unnecessary changes of the torque split factor in an efficient way. The basis for the development of this algorithm was not primarily formed by decreasing the number of switches, rather by the avoidance of inefficient EM-synchronization phases at certain times.

Energy imp. (%)	FTP75	Urban	Extraurban	Highway
Suboptimal design	3.49	3.77	3.74	5.19
Predictive design	3.71	3.64	4.01	4.46
Hysteresis design	3.41	4.00	4.27	2.25

Table 3.7: Energy improvement results ($\Delta SOC_{imp,BM5}$) over real-life DC as a comparison between the best fitted designs selected for each calibration methodology versus the ordinary suboptimal design approach.

# Switches	FTP75	Urban	Extraurban	Highway
Suboptimal design	156	172	336	234
Predictive design	78	83	82	15
Hysteresis design	72	37	49	75

Table 3.8: Number of driving mode switches over real-life DC for major tq split designs.

Consider table 3.7 and 3.8 illustrating final comparisons of the best fitted designs together with the ordinary suboptimal version for the most crucial parameters - energy efficiency and driving mode switches. Note that these results are based on the model with EM speed synchronization, and thus, they are not directly comparable to the results presented in the Direct Calibration section.
For all conducted simulations a 50:50 constant torque split distribution leads to the worst results in terms of energy consumption so that all investigated designs can reach improvements up to 5 % for various driving cycles. BM2, the suboptimal design, forces the system to stay at the theoretical optimum operating point for every time, implicating the compromise of energy efficiency due to several synchronization phases realized in the updated model.

BM13 with the hysteresis parameters $\alpha = 40\,\mathrm{Nm}$, $\beta = 10\,\mathrm{Nm}$ generally achieves convincing results for different driving profiles. Compared to other benchmarks the number of switches also reaches a minimum which has a positive impact on comfort, safety and durability issues. Only the minimum time between two switches probably needs to be enhanced by further restrictions (e.g. time hysteresis approach), since this aspect is not considered explicitly within the threshold based calibration technique.

The predictive design with an interval of 4 s turned out to become a real alternative to BM13, if appropriate training data for the control algorithm is provided and a possible risk to wrong decisions can be overcome. Trying to generate more precise and adequate data (e.g. also from other sensors) which could be linked to the driving behavior, would most likely increase the prediction accuracy and help to improve the overall efficiency. The presented algorithm produces results deviating around the suboptimal design, also the number of switches stays at an acceptably low amount.

In summary, an application of the presented threshold-based hysteresis design can be generally recommended in order to calibrate the discussed suboptimal torque split algorithm in a sufficient way. However, predictive approaches can truly produce comparable or even better results if appropriate implementations of the algorithm are provided with meaningful and consistent input data.

4 Temperature optimization

"Nothing at all takes place in the universe in which some rule of maximum or minimum does not appear."
Leonhard Euler (1707-1783)

Temperature as a fundamental physical quantity plays an important role in all fields of natural science and needs to be carefully observed to gain optimum control of the system itself. For instance, many physical properties rely on temperature very strongly, chemical processes do not start to react until certain heat levels are reached and also metabolic processes of human beings are broadly influenced by temperature.

The proposed OpEneR prototype vehicle introduced in chapter 2 is equipped with two permanent magnet synchronous machines (PMSM) to drive each axle of the car separately. According to the literature, PMSM are widely used for hybrid and electric vehicles since they provide good efficiencies including high power density and fast dynamics. However, due to ambient thermal conditions and the process of heating during operation the temperature of an EM can vary between $-40\,°\mathrm{C}$ and $150\,°\mathrm{C}$ which puts the motor under excessive thermal stress and can even cause a total failure in the worst case. More detailed, as the temperature is rising a successive demagnetization of the permanent magnet happens inside the EM causing irreversible damages to the motor if certain limits are exceeded. Especially this brings up the need for an adequate design of motors and a suitable observation of the EM operating temperature. [GK]

With respect to the upcoming study in this chapter the most interesting aspect is the fact that the efficiency drops the more an EM is heating up, which means a lower torque output for the same consumed electric energy. It

is evident that there is a need to design an adequate controller which takes into account the current EM temperature to generate a more accurate and realistic system model in order to develop further optimum energy strategies. The objective for an efficient torque split algorithm obtaining the optimum temperature behavior for both EM used in the proposed EV will be introduced in the following section.

4.1 Problem definition

Based on a specified road and speed profile the correspondent torque demands for the proposed electric vehicle can be precalculated using a simulation environment like AVL CRUISE. The idea is to develop an algorithm which generates the optimum temperature trajectories for both EM over a predefined time span with respect to the listed conditions concerning the application of the correct torque and speed demands. In this case, optimality refers again to the minimization of energy consumption for the specified running time. The objective is to gain the optimum rotor and stator temperature behavior for front and rear EM considering the optimum load distribution over time in order to maximize the overall efficiency and to obtain a minimum of consumed electric energy in the end. Cf. fig. 4.1 illustrating an example for optimum temperature trajectories of the front EM between time t_0 and t_{end}.

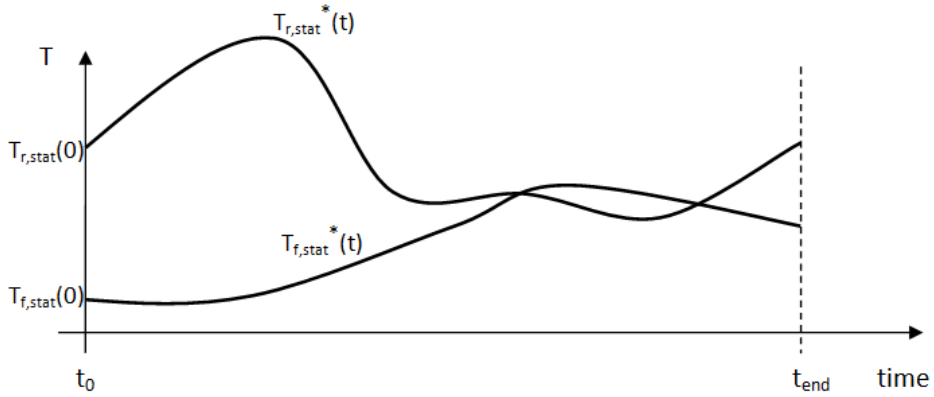

Figure 4.1: Exemplary optimum stator temperature trajectories of both EM during a certain time frame with initial temperature T(0) at time t_0.

From a mathematical point of view the optimization task is described as:

$$\int_{t_0}^{t_{end}} P_{el} \cdot dt \;\; \rightarrow \min \tag{4.1}$$

Since $P_{el} = f(\eta_{2EM})$ and η_{2EM} is a nonlinear function $f(\eta_f, \eta_r, u_f)$ as it was already derived in chapter 3, it has to be pointed out that the problem defined in (4.1) becomes a highly nonlinear optimization task. Given the fact that the efficiencies of front and rear EM also represent a nonlinear relation of the correspondent inputs and need to be estimated *online* using the Matlab/Simulink-model during optimization, the problem becomes even more difficult to solve. As a consequence the time span $(t_{end} - t_0)$ has to be limited adequately in order to prevent too long computation times.

Since the temperature cannot be influenced directly from outside and all other variables are predetermined due to the specifications, obviously the torque split factor u_f represents the only control variable being left. Based on the definition in (3.1) u_f has to satisfy the condition for all times:

$$0 \leq u_f \leq 1 \tag{4.2}$$

Furthermore, the integral in problem 4.1 must be replaced by a summation since in general, simulations require a system to be discretized both in the time domain as well as in value domain in order to be executed on a digital computer. The claim of an adequate time frame for the specified problem can now be changed to an appropriate division of the time span into a sufficiently large amount of n time steps with length ΔT.

In summary the presented nonlinear optimization task with linear inequality constraints can be expressed as follows:

$$\boxed{\begin{aligned} &\min_{\boldsymbol{u_f} \in \mathbb{R}^n} \quad J := \sum_{k=0}^{n-1} P_{el,k}(u_{f,k}) \cdot \Delta T \\ &\text{subject to:} \\ &\qquad 0 \leq \boldsymbol{u_f} \leq 1 \end{aligned}} \tag{4.3}$$

In this case $J : \mathbb{R}^n \rightarrow \mathbb{R}$ denotes the objective function which has to be minimized, $\boldsymbol{u_f}$ represents the vector of n optimization variables, and the constraints $0 \leq \boldsymbol{u_f} \leq 1$ define a set of lower and upper bounds on $\boldsymbol{u_f}$.

The optimum solution of the problem will yield to a minimum energy consumption J^* which directly results from the optimum torque split factor $\boldsymbol{u_f}^*$ obtained through the minimization algorithm. As a consequence, having knowledge about the load distribution $\boldsymbol{M_f}^*$ and $\boldsymbol{M_r}^*$ will automatically lead to the optimum temperature behavior of front and rear EM over time, such as $\boldsymbol{T_{f,stat}}^*$, $\boldsymbol{T_{f,rot}}^*$, $\boldsymbol{T_{r,stat}}^*$ and $\boldsymbol{T_{r,rot}}^*$. Note that all variables except J have been grouped to vectors with n elements each.

The main part of the proposed nonlinear optimization problem is formed by a deep analysis and subsequent modeling process regarding the thermal behavior of both EM integrated in the vehicle's dedicated cooling system. It is important to note that the specific logic for controlling the cooling system itself is subject to particular conditions as well, however, cannot be controlled independently and integrated into the optimization problem. The approach for modeling the thermal behavior of front and rear EM during the drive is described in the following section.

4.2 Thermal model

A suitable thermal model for the electric machines integrated in the proposed EV with respect to the correspondent EM cooling circle forms the basis for the upcoming optimization process introduced in the beginning of this chapter. It is helpful to recall fig. 2.6 in section 2.3, which describes the relationship of the EM-efficiency as a function of several inputs: $f(M_{EM}, \omega_{EM}, T_{rot}, T_{stat}, V_{DC})$. In contrast to the developed energy strategies in chapter 3, now the EM-temperature will not be kept constant over time and needs to be considered in further calculations. Regarding the difference between rotor and stator of an EM implicating the basic mode of operation for PMSM it is hereby referred to section 2.3 in chapter "Vehicle model".

Thermal model requirements:

- Representing a control oriented functional model of the EM temperature behavior
- Easy integration into the selected optimization method
- Preferably fast to facilitate computations
- Successful validation with different measurement data in the best case

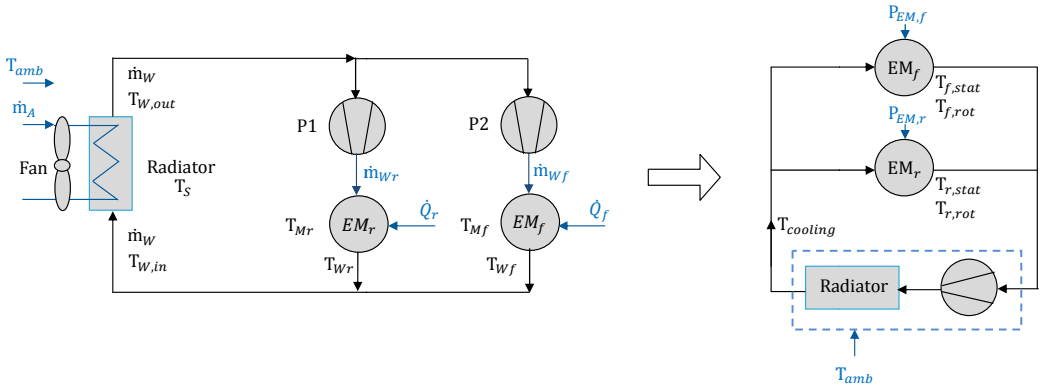

Figure 4.2: EM cooling system layout for the OpEneR prototype vehicle (left), simplified version for deriving a suitable thermal model for optimization strategies (right)

To start from scratch it is essential to analyze both the heating of the EM as well as the underlying cooling system since an adequate model of inducing and dissipating of heat will reflect the final temperature behavior in a good

way. Consider the left system in fig. 4.2 representing the actual EM cooling circuit used in the OpEneR prototype vehicle. Two pumps control the mass flow of the coolant for each EM separately, resulting in a combined mass flow $\dot{m}_w$ at the conjunction point. Heat exchange happens at the radiator which is supported by a fan running at different stages (different $\dot{m}_a$) and giving the impact for cooling down the radiator itself. The heat flow $\dot{Q}_f$ and $\dot{Q}_r$ caused by the correspondent power loss of each EM is responsible for a positive change of temperature. Note, that this model would consist of a relatively large amount of inputs ($\dot{Q}_f$, $\dot{Q}_r$, $\dot{m}_{Wf}$, $\dot{m}_{Wr}$, $\dot{m}_a$, T_{amb}) and system states representing the temperature of each subsystem itself.

With respect to the modeling purpose the presented layout was reduced to the system visualized on the right side of fig. 4.2. It consists of rear EM, front EM, radiator, one pump and the correspondent connections reflecting the flow of the coolant. Considerably good approximations to a set of real measurement data finally justify the performed simplification of the cooling structure. Note, that in this model the EM temperature needs to be split onto stator and rotor temperature to serve the proposed EM model in Matlab/Simulink. Furthermore, besides T_{amb} the inputs to the system comprise torque, speed and supplied battery voltage to approximate the correspondent power loss being mainly responsible for the heating process in the EM.
Summarized, the presented EM cooling system can be described with states:

$T_{f,stat}$ ..... stator temperature of front EM (°C)
$T_{f,rot}$ ..... rotor temperature of front EM (°C)
$T_{r,stat}$ ..... stator temperature of rear EM (°C)
$T_{r,rot}$ ..... rotor temperature of rear EM (°C)
$T_{cooling}$ ..... temperature of coolant (°C)

and inputs:

M_f, M_r ..... desired torque at front/rear EM (Nm)
ω_f, ω_r ..... desired speed at front/rear EM (rad/s)
V_{DC} ..... battery voltage (V)
T_{amb} ..... ambient temperature (°C)

4.2.1 Stator temperature model

Basically a simple thermal model needs to consider factors for heating up and cooling down the EM, respectively. Furthermore it is important to check stability issues (implicating eigenvalues < 0) and other aspects concerning the time discretization of the system which is described later on. The presented model in this section has been adapted from [Kit10].
Generally it can be said that the stator of an electric machine heats up according to the power loss an EM is exposed during its operation over time. Especially with respect to modeling issues, this power loss can be approximated by the mechanical power P_{EM} due to proportionality of both terms. For instance, if an EM is running in motor mode the correspondent power loss yields to:

$$P_{loss} = P_{el} - P_{EM} = P_{EM} \cdot \frac{1 - \eta_{EM}}{\eta_{EM}} \tag{4.4}$$

$$P_{loss} \propto \omega\, M \tag{4.5}$$

The thermal interaction between EM and coolant surrounding the correspondent stator (which forms the chassis of the EM) results in a continuing heat exchange between both structures. For simplicity it can be assumed that the coolant heats up according to the mean temperature of both EM and cools down with respect to the ambient temperature. To some degree this behavior reflects the impact of the fan in the original circuit in fig. 4.2, which is responsible for the heat release of the radiator. Finally, a simplified thermal state space model can be derived according to (4.6):

$$\begin{aligned}
\frac{dT_{f,stat}}{dt} &= -\alpha\,(T_{f,stat} - T_{cooling}) + \beta\ |\omega_f(t)\, M_f(t)| \\
\frac{dT_{r,stat}}{dt} &= -\alpha\,(T_{r,stat} - T_{cooling}) + \beta\ |\omega_r(t)\, M_r(t)| \\
\frac{dT_{cooling}}{dt} &= \gamma \left(\frac{T_{f,stat} + T_{r,stat}}{2} - T_{cooling} \right) - \delta\,(T_{cooling} - T_{amb})
\end{aligned} \tag{4.6}$$

The parameters $(\alpha, \beta, \gamma, \delta)$ have been adjusted accordingly with respect to validation against the provided measurement data.
$(\alpha = 3,85 \cdot 10^{-3}\,\frac{1}{s},\ \beta = 1,375 \cdot 10^{-5}\,\frac{^{\circ}\mathrm{C}}{Nm},\ \gamma = 0.03\,\frac{1}{s},\ \delta = 0.07\,\frac{1}{s})$

Stability

Since real temperature trajectories would always follow a good-natured behavior if no external forces were applied (system is called stable), it is important to check if certain conditions are satisfied for the derived model as well. In order to guarantee stability for the proposed dynamical system, real parts of the eigenvalues calculated for the system-matrix A have to be less than zero. For the state space model described in (4.6) A equates to:

$$A = \begin{bmatrix} -\alpha & 0 & \alpha \\ 0 & -\alpha & \alpha \\ \gamma/2 & \gamma/2 & -\gamma - \delta \end{bmatrix} \tag{4.7}$$

For the specified data following eigenvalues can be retrieved:

$$\lambda_1 = -0.1012,\ \lambda_2 = -0.0039,\ \lambda_3 = -0.0027 \tag{4.8}$$

4.2.2 Rotor temperature model

Several studies have been conducted to analyze rotor losses of PM synchronous motors used in high speed applications. Generally, the role of rotor losses have to be carefully investigated since they are responsible for heating up the internal permanent magnet resulting in a decrease of torque and total efficiency. Furthermore there is a danger of non-reversible demagnetization of the PM caused by exceedingly high rotor temperatures [ZAS; SB]. According to the literature the main heat source is formed by iron losses p_{Fe} which can be divided into hysteresis and eddy current losses at the rotor [Acq12; SB]:

$$p_{Fe} = p_{hyst} + p_{ec} = c_{hyst} f \hat{B}^2 + c_{ec} f^2 \hat{B}^2, \tag{4.9}$$

where f and $\hat{B}$ denote frequency and peak value of the flux density. The constant coefficients c_{hyst} and c_{ec} are used to model the losses.
Assuming that inverters for PMSM usually implement Pulse Width Modulation (PWM) to generate AC out of the provided battery current, high frequencies due to significant harmonics in the output current would consequently lead to high iron losses in the rotor [BJ]. Furthermore, changes of the

reluctances in the air-gap between stator and rotor as well as the geometrical shape of the rotor have an additional effect on the losses [ZAS].

Since the attempt of developing a suitable mathematical model based on the speed dependency of the iron losses does not fit to the provided measurement data, also because further models were found out to correlate hardly with the complex physical thermal behavior, it was suggested to analyze a secondary approach with respect to an appropriate implementation for the optimization task. Thus, in order to study the impact of temperature variation with respect to the resulting EM-efficiency a sensitivity analysis was conducted. Consider fig. 4.3 for the results of this analysis.

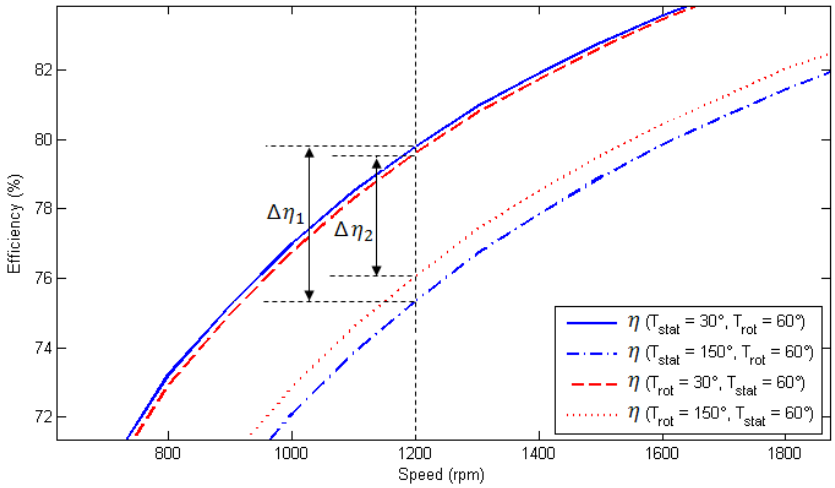

Figure 4.3: Sensitivity analysis of rotor/stator temperature variation against EM-efficiency. Torque and speed values have been chosen so that an average power of 30 kW was applied for high speeds.

$\Delta\eta_1$ ... max. efficiency deviation caused by variation of T_{stat} ($T_{rot} = 60\,°C$)
$\Delta\eta_2$ ... max. efficiency deviation caused by variation of T_{rot} ($T_{stat} = 60\,°C$)

Different torque and speed values close to the maximum EM operating curve (see fig. 2.5) have been used as inputs to the EM model together with constant stator and varying rotor temperatures (and vice versa) covering the whole range of operation. For clarity in fig. 4.3 only the curves for 30 °C and 150 °C are visualized since they represent the outer boundaries for all investigated

temperatures.
Based on the given relation it was found out that the sensitivity of the rotor temperature $\frac{\partial \eta}{\partial T_{rot}}$ is about 20 % lower compared to the stator's, meaning:

$$\frac{\Delta\eta_1 - \Delta\eta_2}{\Delta\eta_1} \approx 20\,\%, \text{ and sensitivity in general:} \tag{4.10}$$

$$\frac{\partial \eta}{\partial T_{rot}} < \frac{\partial \eta}{\partial T_{stat}} \tag{4.11}$$

Hence, a variation of the rotor temperature has generally less influence on the overall EM efficiency compared to the impact of the stator's temperature. As a consequence, with respect to the thermal EM-model used in the proposed optimization task the rotor temperature of both EM $T_{f,rot}$ and $T_{r,rot}$ is assumed to stay constant within the typical range obtained in the measurement data. As it will be proved in the validation section 4.5 later on, this assumption can truly be justified since the error of the complete thermal model is kept in an acceptable range.

4.2.3 Time discretization

For any continuous system with differential equations of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}), \tag{4.12}$$

the easiest way to transform the relation into the time-discrete domain with step ΔT is to use the approximation of the differential quotient:

$$\frac{dx}{dt} \approx \frac{x((k+1)\,\Delta T) - x((k)\,\Delta T)}{\Delta T} \tag{4.13}$$

Perceiving the derived thermal models of stator and rotor temperature as a system like described in (4.12) with states $\boldsymbol{x} = (T_{f,stat}, T_{r,stat}, T_{f,rot}, T_{r,rot})$ and input $\boldsymbol{u} = (M_f, M_r, \omega_f, \omega_r, V_{DC})$, the correspondent temperature at time $(k+1)\,\Delta T$ can be calculated using the previous state values of time $k\,\Delta T$:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) \cdot \Delta T \tag{4.14}$$

Since this approximation reflects the Euler method for the numerical solution of ordinary differential equations, further conditions concerning the continuous state space model have to be satisfied. In order to guarantee "absolute stability" of the algorithm, all eigenvalues λ_i of the system-matrix A need to meet following equation [Hof04]:

$$\begin{aligned} &|1 + \Delta T\, \lambda_i| < 1 \\ &-2 < \Delta T\, \lambda_i < 0 \quad \text{for real valued } \lambda_i \end{aligned} \tag{4.15}$$

Given the fact that eigenvalues are determined by the system itself, the constraint defined in (4.15) especially restricts the choice of ΔT to a set of admissible time discretizations in order to fullfil stability requirements of the algorithm. For the proposed model with eigenvalues obtained in eq. (4.8), ΔT is subject to following boundary condition:

$$\Delta T < 20\,\mathrm{s} \tag{4.16}$$

4.3 NLP formulation

After all necessary aspects regarding the nonlinear optimization problem stated in 4.3 have been analyzed, the procedure for solving the NLP for a given torque and speed profile can be derived in the following.

For a specified set of torque splits defined for all steps in the considered time interval, the correspondent torque demands of front and rear EM can be calculated. The previously developed thermal model is used to simulate a realistic temperature behavior based on the current load distribution. Recall the EM model implemented in Matlab/Simulink (cf. section 2.3), which implies the need for simulating the EM efficiencies for a given set of inputs online during the optimization process. After calculating the overall electrical energy consumption it can be decided if the current result leads to a solution of the optimization problem - or, if the process has to be repeated using a modified torque split configuration.

In summary the process of solving the NLP can be formulated as follows:

Algorithm 1: Formal description for solving the proposed NLP

Data: torque demand $\boldsymbol{M_{2EM}}$, speed $\boldsymbol{\omega}$ over complete time interval

Result: J^*, $\boldsymbol{u_f}^*$, $\boldsymbol{T_{r,stat}}^*$, $\boldsymbol{T_{f,stat}}^*$

repeat

1.) set $\boldsymbol{u_f}$ accordingly
    check boundary constraints

2.) calculate torque demand for both EM based on torque split
    obtain: $\boldsymbol{M_f}$, $\boldsymbol{M_r}$

3.) simulate temperature behavior for both EM
    use the developed thermal model
    obtain: $\boldsymbol{T_{r,stat}}$, $\boldsymbol{T_{f,stat}}$

4.) provide necessary inputs to EM model in Simulink
    obtain: efficiencies of front and rear EM over time

5.) calculate Energy consumption J over complete time interval

until *J is a minimum*;

Difficulties emerge due to various nonlinearities caused by obtaining the efficiencies through the simulation of the EM model in Matlab/Simulink. Furthermore calculating the overall energy consumption is done by using eq. (3.5), which implicates additional nonlinearities included in the optimization problem. In general locating the global optimum for such a complex NLP is regarded as a rather challenging task and thus, needs the application of a powerful and efficient methodology at least.

According to the literature "Sequential Quadratic Programming" is likely to be used for solving nonlinear optimization problem with constraints and was therefore selected as a suitable method for the proposed optimization task. General aspects and the mathematical approach towards SQP will be introduced in the following section.

4.4 Sequential Quadratic Programming

Sequential Quadratic Programming (SQP) refers to an effective method for solving nonlinear optimization problems with constraints. The basic idea is to approximate the nonlinear objective function locally by a quadratic problem which is easier to solve, and finally to iterate subsequently until an optimum solution is obtained by checking predefined tolerance limits. With respect to SQP it is necessary that all appearing functions are continuously differentiable around the neighborhood of the correspondent optimum point. The quadratic problem during each iteration is derived using a related 2nd order Taylor approximation of the Lagrange functional together with linearized constraints. The solution of this intermediate optimization task produces a step size which is used within the main iteration to converge successively towards the local minimum [Kal]. Consider fig. 4.4 illustrating an exemplary quadratic approximation for several iterations.

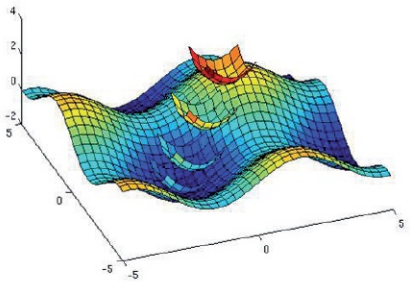

Figure 4.4: SQP visualization [Kal]

4.4.1 Introductory definitions

The standard form for a nonlinear optimization problem (NLP) is expressed as follows:

$$\begin{aligned} &\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad f(\boldsymbol{x}) \\ &\text{subject to:} \\ &\qquad g_i(\boldsymbol{x}) = 0, \quad i = 1,..,p \\ &\qquad h_j(\boldsymbol{x}) \leq 0, \quad j = 1,..,q \end{aligned} \tag{4.17}$$

with $f : \mathbb{R}^n \to \mathbb{R}$ (objective function), $g_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1,..,p$ (equality constraints), $h_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1,..,q$ (inequality constraints).

Definition 4.4.1. ***Feasible set***
The set of points where all equality and inequality constraints are satisfied

$$\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^n \,|\, g_i(\boldsymbol{x}) = 0, i = 1, ..p,\ h_j(\boldsymbol{x}) \leq 0, j = 1, ..q\}$$

is called "feasible set" and contains all admissible points for $\boldsymbol{x}$*.*

Definition 4.4.2. ***Linear Independent Constraint Qualification (LICQ)***
Let $\hat{\boldsymbol{x}}$ *be a feasible point* $\hat{\boldsymbol{x}} \in \mathcal{X}$*.*
The set of indexes $A(\hat{\boldsymbol{x}}) := \{j \in \{1, 2, .., m\} \,|\, h_j(\hat{\boldsymbol{x}}) = 0\}$ *is referred to as the set of* ***active inequality constraints****.*
It is supposed that g_i *and* h_j *are differentiable, then LICQ is satisfied in* $\hat{\boldsymbol{x}}$ *if the gradients*

$$\begin{aligned} &\nabla g_i(\hat{\boldsymbol{x}}) \quad i = 1, .., p, \\ &\nabla h_j(\hat{\boldsymbol{x}}) \ \ j \in A(\hat{\boldsymbol{x}}) \end{aligned}$$

are linearly independent for all active constraints in $\hat{\boldsymbol{x}}$*. According to the literature* $\hat{\boldsymbol{x}}$ *is also said to be* ***regular****.*

Definition 4.4.3. ***Lagrange functional***
Let $\boldsymbol{\lambda} \in \mathbb{R}^p$, $\boldsymbol{\mu} \in \mathbb{R}^q$*. Then the Lagrange functional* L *corresponding to the NLP* (4.17) *is defined as:*

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) := f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{\mu}^T \boldsymbol{h}(\boldsymbol{x})$$

with $\boldsymbol{g} = [g_1\, g_2\, ..\, g_p]$ *and* $\boldsymbol{h} = [h_1\, h_2\, ..\, h_q]$*.* $\boldsymbol{\lambda}$ *and* $\boldsymbol{\mu}$ *are called the Lagrangian multipliers.*

Throughout the following it is assumed that $f(\boldsymbol{x})$, $\boldsymbol{g}(\boldsymbol{x})$ and $\boldsymbol{h}(\boldsymbol{x})$ are at least two times differentiable. The definition of first order necessary optimality conditions are referred to as the "Karush-Kuhn-Tucker" (KKT)-conditions and are satisfied for any critical point in the NLP stated in (4.17).

Theorem 4.4.1. ***KKT-conditions of first order***
Let $\boldsymbol{x}^*$ be a local minimum of NLP (4.17). Furthermore it is assumed that LICQ holds at $\boldsymbol{x}^*$ ($\boldsymbol{x}^*$ is regular). Then, there exists a Lagrangian multiplier $((\boldsymbol{\lambda}^*)^T, (\boldsymbol{\mu}^*)^T)$ which is unique and satisfies following conditions:

$$\nabla_x L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \nabla f(\boldsymbol{x}^*) + \nabla \boldsymbol{g}(\boldsymbol{x}^*)\boldsymbol{\lambda}^* + \nabla \boldsymbol{h}(\boldsymbol{x}^*)\boldsymbol{\mu}^* = \mathbf{0} \quad (4.18)$$

$$\boldsymbol{\mu}^* \geq \mathbf{0} \quad (4.19)$$

$$\boldsymbol{h}^T(\boldsymbol{x}^*)\boldsymbol{\mu}^* = 0 \quad (4.20)$$

$$\boldsymbol{h}(\boldsymbol{x}^*) \leq \mathbf{0} \quad (4.21)$$

with $\nabla \boldsymbol{g} = [\nabla g_1 \, \nabla g_2 \, .. \, \nabla g_p]$ and $\nabla \boldsymbol{h} = [\nabla h_1 \, \nabla h_2 \, .. \, \nabla h_q]$.

Cf. [Kug, p. 62] for proof of theorem 4.2.1. □
Note, that a critical point does not necessarily need to be a local minimum. Sufficient conditions have to be checked to confirm the actual characteristic. Although many optimization methods rely on the calculation of KKT-points, this approach is still sufficient if it can be assumed that local maximums do not occur or initial estimates for $\boldsymbol{x}$ are close to the local minimum. Otherwise several strategies can be applied which try to avoid this problem by globalizing the SQP-algorithm [Kal, p. 7].

4.4.2 Lagrange-Newton method

Newton-Raphson iteration

The Newton-Raphson iteration refers to a common way for numerical calculation of local extrema solving equations of the form $\nabla f(x) = 0$.
Let $F(x) := \nabla f(x)$, then the Taylor approximation of $F(x)$ at a certain point $x^{(k)}$ is valid for a sufficient small area $\Delta x^{(k)} = (x - x^{(k)})$ around $x^{(k)}$.

$$F(x) = F(x^{(k)}) + F'(x^{(k)})\,(x - x^{(k)}) + O((x - x^{(k)})^2) \quad (4.22)$$

Setting $F(x) = 0$ and neglecting terms of order ≥ 2 leads to a converging iteration with initial value $x^{(0)}$:

$$F'(x^{(k)})\,\Delta x^{(k)} = -F(x^{(k)}) \tag{4.23}$$

$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)} \quad k = 0, 1, 2, ... \tag{4.24}$$

NLP optimization

Since all definitions and prerequisites have been clarified in the previous sections, the approach for finding a local minimum of the NLP stated in (4.17) can now be derived. Based on the KKT-conditions holding for a regular local minimum $\boldsymbol{x}^*$, following nonlinear system of equations can be expressed:

$$\begin{bmatrix} \nabla_x L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \\ \nabla_\lambda L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \\ \nabla_{(\mu_j)_{j\in A}} L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \end{bmatrix} = \begin{bmatrix} \nabla f(\boldsymbol{x}^*) + \nabla \boldsymbol{g}(\boldsymbol{x}^*)\boldsymbol{\lambda}^* + \nabla \boldsymbol{h}(\boldsymbol{x}^*)\boldsymbol{\mu}^* \\ \boldsymbol{g}(\boldsymbol{x}^*) \\ \bar{\boldsymbol{h}}(\boldsymbol{x}^*) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

with $\bar{\boldsymbol{h}}$ including all active inequality constraints defined by the correspondent set of indexes A. Note, that inactive constraints can be neglected locally [Höm, p. 131] and $\mu_j = 0$ for $h_j(\boldsymbol{x}^*) < 0$ (KKT-conditions 4.6-4.8).

Obviously the KKT conditions require the gradient of the Lagrangian functional to be zero with respect to all active contraints: $\nabla L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \boldsymbol{0}$. One option is to solve the proposed system of equations by a Newton iteration which was previously introduced, so that:

$$\begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{\lambda}_{k+1} \\ \boldsymbol{\mu}_{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{\lambda}_k \\ \boldsymbol{\mu}_k \end{bmatrix} + \begin{bmatrix} \boldsymbol{d}_{x_k} \\ \boldsymbol{d}_{\lambda_k} \\ \boldsymbol{d}_{\mu_k} \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} L_{xx}(\boldsymbol{x}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k) & \nabla \boldsymbol{g}(\boldsymbol{x}_k) & \nabla \bar{\boldsymbol{h}}(\boldsymbol{x}_k) \\ (\nabla \boldsymbol{g})^T(\boldsymbol{x}_k) & 0 & 0 \\ (\nabla \bar{\boldsymbol{h}})^T(\boldsymbol{x}_k) & 0 & 0 \end{bmatrix}}_{\boldsymbol{M}_k} \begin{bmatrix} \boldsymbol{d}_{x_k} \\ \boldsymbol{d}_{\lambda_k} \\ \boldsymbol{d}_{\mu_k} \end{bmatrix} = -\begin{bmatrix} \nabla_x L(\boldsymbol{x}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k) \\ \boldsymbol{g}(\boldsymbol{x}_k) \\ \bar{\boldsymbol{h}}(\boldsymbol{x}_k) \end{bmatrix}$$

with $\nabla \boldsymbol{g} = [\nabla g_1\, \nabla g_2\, ..\, \nabla g_p]$ and $\nabla \bar{\boldsymbol{h}} = [\nabla h_j]\ j \in A$.

$L_{xx}(\boldsymbol{x}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k)$ denotes the Hessian matrix $(\frac{\partial^2}{\partial \boldsymbol{x}^2} L)(\boldsymbol{x}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k)$.

The matrix $\boldsymbol{M}_k$ is invertible if it has full rank, which is the case when:

- LICQ holds, so that the gradients $\nabla \boldsymbol{g}$ and $\nabla \bar{\boldsymbol{h}}$ are linearly independent
- and $L_{xx}(\boldsymbol{x}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k)$ is positive definite, which means for any $\boldsymbol{c} \neq 0$ with $(\nabla \boldsymbol{g})^T \boldsymbol{c} = \boldsymbol{0}$: $\boldsymbol{c}^T L_{xx}(\boldsymbol{x}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k)\boldsymbol{c} > 0$ is satisfied (true for strictly local minimum, cf. [Kug, p. 78]).

4.4.3 Derivation of SQP

It is interesting to note that the iteration method can be replaced by an successive calculation of the quadratic optimization problem stated in (4.25), which directly leads to the SQP algorithm.

$$\begin{aligned} &\min_{\boldsymbol{d}_k \in \mathbb{R}^n} \quad f(\boldsymbol{x}_k) + (\nabla f)^T(\boldsymbol{x}_k)\boldsymbol{d}_k + \frac{1}{2}\, \boldsymbol{d}_k^T L_{xx}(\boldsymbol{x}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k)\boldsymbol{d}_k \\ &\text{subject to:} \\ &\qquad (\nabla \boldsymbol{g})^T(\boldsymbol{x}_k)\boldsymbol{d}_k + \boldsymbol{g}(\boldsymbol{x}_k) \;= \boldsymbol{0} \\ &\qquad (\nabla \boldsymbol{h})^T(\boldsymbol{x}_k)\boldsymbol{d}_k + \boldsymbol{h}(\boldsymbol{x}_k) \leq \boldsymbol{0} \end{aligned} \tag{4.25}$$

Here, $\boldsymbol{d}_k$ is stated as *descending direction* and is applied at the k^{th} iteration in order to proceed to the local minimum. Equivalence between the Lagrange-Newton method and SQP is shown directly by the following derivation.
The KKT-conditions for problem (4.25) with Lagrange multiplicators $(\bar{\boldsymbol{\lambda}}^T, \bar{\boldsymbol{\mu}}^T)$ can be expressed as:

$$\underbrace{\begin{bmatrix} L_{xx}(\boldsymbol{x}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k) & \nabla \boldsymbol{g}(\boldsymbol{x}_k) & \nabla \bar{\boldsymbol{h}}(\boldsymbol{x}_k) \\ (\nabla \boldsymbol{g})^T(\boldsymbol{x}_k) & 0 & 0 \\ (\nabla \bar{\boldsymbol{h}})^T(\boldsymbol{x}_k) & 0 & 0 \end{bmatrix}}_{\boldsymbol{M}_k} \begin{bmatrix} \boldsymbol{d}_k^* \\ \bar{\boldsymbol{\lambda}}_k^* \\ \bar{\boldsymbol{\mu}}_k^* \end{bmatrix} = - \begin{bmatrix} \nabla f(\boldsymbol{x}_k) \\ \boldsymbol{g}(\boldsymbol{x}_k) \\ \bar{\boldsymbol{h}}(\boldsymbol{x}_k) \end{bmatrix} \tag{4.26}$$

Comparing (4.26) with the system of equations stated in the previously introduced Lagrange-Newton iteration, it can be found out that both systems are equal except for a single term in the right vector. After replacing $\boldsymbol{d}_{\lambda_k} = \boldsymbol{\lambda}_{k+1} - \boldsymbol{\lambda}_k$ equivalence is established.

Thus, Newton's iteration can be conducted using the optimum values gained by minimizing the quadratic problem at each step, so that: $\boldsymbol{\lambda}_{k+1} = \bar{\boldsymbol{\lambda}}_k^*$ and $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{d}_k^*$. In other words the descending direction $\boldsymbol{d}_k^*$, which is obtained by solving the QP, equates to the step $\boldsymbol{d}_{x_k}$ in the Lagrange-Newton iteration, both of them used for calculating the next state vector $\boldsymbol{x}_{k+1}$. Also the Lagrange multipliers are equivalent for both methods.

Remarks to SQP

- It is important to note that for SQP solving nonlinear problems with inequality constraints the method of *"active sets"* is applied, so that $\bar{\boldsymbol{h}}$ contains all active inequality constraints at each iteration [Kug, p. 80].
- The convergence rate is proved to be quadratic in a sufficient small area around the local minimum $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ [Kug, p. 79].
- Since the matrix $\boldsymbol{M}_k$ of size $((n + p + |A|)\,\mathrm{x}\,(n + p + |A|))$, where $|A|$ denotes the number of elements in A (active constraints), can get huge for large-scale optimization problems, an efficient calculation for the proposed method is prohibited. According to literature especially the determination of the Hessian matrix $L_{xx}(\boldsymbol{x}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k)$ by using finite differences is rather complex and inaccurate. Furthermore algorithms can get in trouble if L_{xx} is not positive definite. As a consequence approximations are applied using a symmetric matrix $H^{(k)}$ at each step, which is positive definite also when it is not the case for L_{xx}. On the one hand this characteristic possibly leads to a degradation of the convergence rate, on the other hand superlinear convergence can still be proved for initial values close to the local minimum [Kug, p. 82].
- The objective functional in QP (4.25) is related to a 2nd order Taylor approximation of the Lagrangian functional L.
- Global convergence of SQP can be achieved by using certain strategies regarding the choice of step size for the main iteration. Thus, a new factor α_k is introduced in order to update the next value according to the equation:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\, \boldsymbol{d}_k^* \tag{4.27}$$

α_k is selected based on so called "merit functions", which try to validate if a certain value $\boldsymbol{x}_{k+1}$ is "better" than $\boldsymbol{x}_k$. Within SQP it is possible that inadmissible values $\boldsymbol{x} \notin \mathcal{X}$ can occur, thus, special functions have to represent a weighted behavior to penalize possible breaking of constraints with respect to the minimization of the objective function.
E.g. the "l_1"-merit function is known to be very popular and can be referred in [Kug, p. 83]. The choice of an optimum step size is now derived based on a line-search problem:

$$\alpha_k = \arg\min_{\alpha} \; l_1(\boldsymbol{x}_k + \alpha\, \boldsymbol{d}_k^*)$$

According to the literature several strategies exist which try to facilitate the calculation efficiency for this intermediate optimization problem. The most common algorithms are based on the "Armijo-rule" and stronger "Wolfe-Powell" conditions (cf. [Kug, p. 83] and [Kal, p. 14]).

Finally consider fig. 4.5 visualizing the structure of the globalized SQP algorithm. The stopping point is usually reached when differences of function values or step sizes do not exceed a certain threshold ϵ.

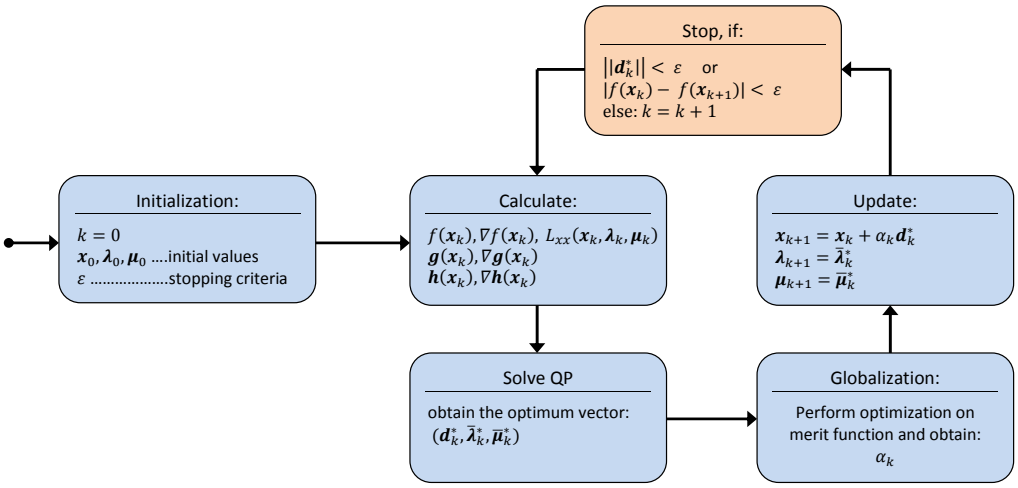

Figure 4.5: Flow chart of the globalized SQP algorithm

4.5 Validation

After a design process has been conducted it is essential to analyze the developed system with respect to design criteria, fulfillment of specifications and plausibility of results. This so called verification/validation process shall now be performed with respect to the previously developed thermal model for PMSM to estimate the temperature behavior. For this purpose the OpEneR prototype vehicle was equipped with different sensors to measure various states of the powertrain as well as the current rotor/stator temperatures of front and rear EM. During several test runs conducted at Boxberg/Germany, data series of different maneuvers have been collected and can now be used for a comparison of model results with respect to real measurement data.

4.5.1 Thermal model validation

Consider fig. 4.6 illustrating temperature trajectories for two exemplary test runs with ambient temperature of $-8\,°\mathrm{C}$. In each diagram the stator temperature of front and rear EM is visualized to compare the modeled behavior with real measurements taken from the correspondent sensor in the vehicle. Note that the input to the model comprises torque and speed values extracted from the collected data. A time discretization with $\Delta T = 1s$ satisfies condition (4.15) for eigenvalues derived in eq. (4.8) and also ensures acceptable accuracy with respect to low computation times especially for the optimization algorithm.

It was found out that the resulting temperature behavior obtained by the thermal model approximates the given measurement trajectories remarkably well. In both cases the tendency of heating and cooling is clearly evident and matches the real behavior to a great extend. Also comparing the results with further test data confirms the assumption that the developed model in (4.6) is sufficiently precise to satisfy all expectations and to serve as a basis for the proposed temperature optimization task.

In order to provide a clear measure for the accuracy of the model the error with respect to EM-efficiency was calculated. More specifically, inputs to the

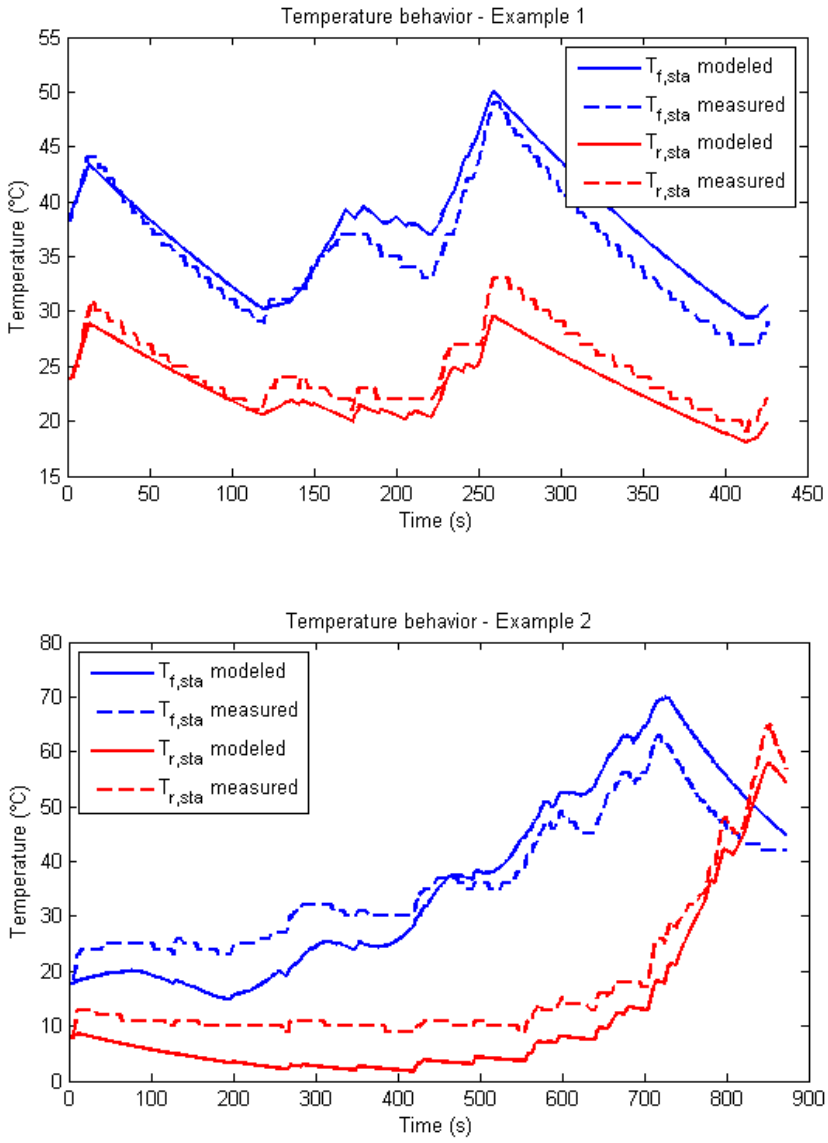

Figure 4.6: Comparison between modeled and measured EM stator temperature for two different real-vehicle test runs ($T_{amb} = -8\,°\mathrm{C}, T_{cooling}(0) = 5\,°\mathrm{C}$).

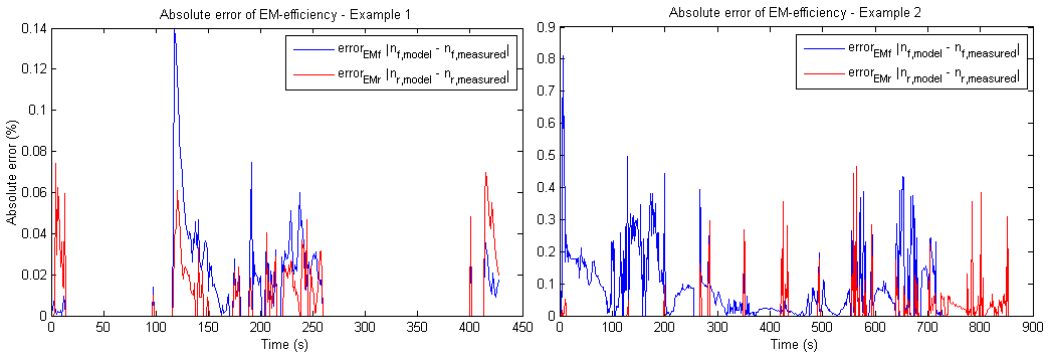

Figure 4.7: Efficiency error for thermal model compared to measurement data. Example 1: $T_{f,rot} = 40\,°\mathrm{C}$, $T_{r,rot} = 50\,°\mathrm{C}$; Example 2: $T_{f,rot} = 40\,°\mathrm{C}$, $T_{r,rot} = 60\,°\mathrm{C}$

EM model in Matlab/Simulink were formed by the given torque/speed profile of the driving maneuver and by using temperature values, which could be obtained by the thermal model and the measurement data, respectively. Afterwards, efficiency outputs were compared to each other to figure out whether the error was in an acceptable range. Consider fig. 4.7 illustrating the correspondent efficiency error between model and measurement data for front/rear EM and both exemplary test runs.

As expected the error $|\eta_{modeled} - \eta_{measured}|$ is highly dependent on the quality of correlation between measured and modeled EM temperature. In case of example 1, the maximum error is about 0.14 % compared to relatively higher values for example 2. Comparing again with fig. 4.6 it can be seen, that the deviation of temperature differs for both examples which naturally has an impact on the efficiency error. However, according to the results of several (other) test runs it could be found out that the deviation never exceeded limits of $+/-10\,°\mathrm{C}$ resulting in a maximum error of efficiency of about 0.8%. Thus, the validation of the thermal model could be successfully confirmed especially with respect to the specifications listed in section 4.2.

4.5.2 Evaluation of SQP

Given the fact that simulation results concerning the optimum torque distribution are already present from chapter 3, it is reasonable to evaluate SQP in terms of the optimized torque split behavior and resulting energy consumption. For this special analysis it is important to request the EM-temperature staying constant over the optimization interval since all simulations from chapter 3 were conducted with $T_{rot} = T_{stat} = 60\,°\mathrm{C}$. Furthermore the NLP stated in (4.3) has to be adapted so that $0 \leq \boldsymbol{u_f} \leq 0.5$ is satisfied. After defining initial values $\boldsymbol{u_{f,0}}$ for the optimization vector, SQP can be performed[1] with respect to the torque and speed demands for a certain driving cycle. Consider fig. 4.8 where the torque split behavior is compared for the first section of NEDC (0 - 200 s).

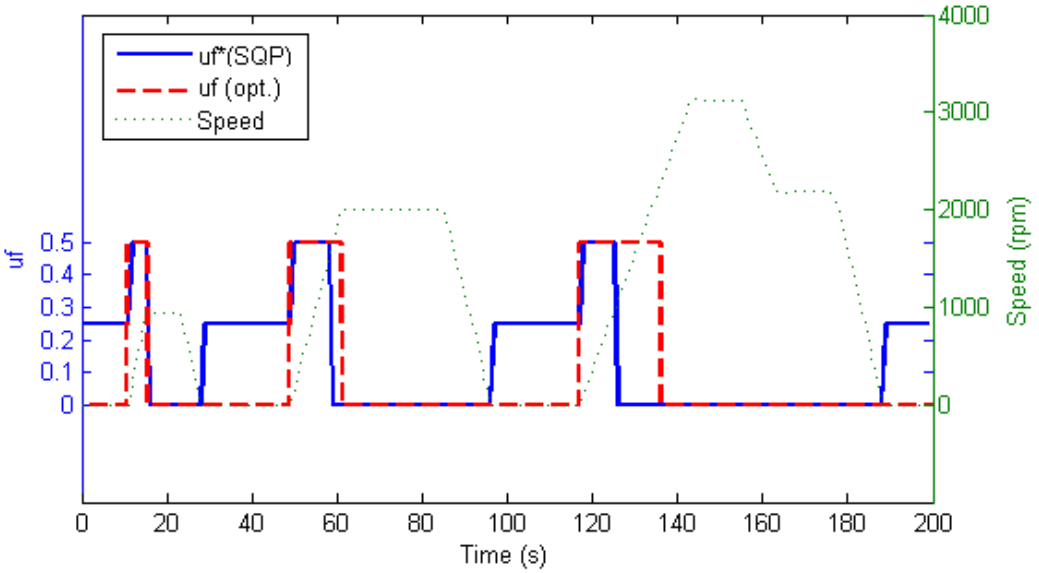

Figure 4.8: Torque split factor u_f obtained by SQP and optimal design for the first section within NEDC ($0-200\,s$ with $\Delta T = 1\,s$, $\boldsymbol{u_{f,0}} = [0.25]$). The dotted line indicates the corresponding speed profile during this interval.

Analysis:

Obviously a good match for the torque split behavior can be achieved neglecting trajectories for $u_f^* = 0.25$ at certain intermediate intervals where the

[1] In Matlab the correspondent command for performing nonlinear optimization with SQP is called $fmincon(\cdot)$.

EM-speed is zero. This does not imply any restrictions since standstill-phases of the vehicle do not contribute to the energy consumption at all (no auxiliaries, only traction is taken into account). Note that the EM speed directly corresponds to the vehicle speed since both clutches are defined to stay closed all the time. Given the fact that $\boldsymbol{u_{f,0}} = [0.25]$ the behavior for those intervals becomes self-evident.

The remaining differences appearing at around $60\,s$ and $130\,s$ may give clues about a possible non-perfect optimization of SQP, meaning that the "global optimum" which links to the minimum energy consumption during $0 - 200\,s$ has not been found. This problem is well known for SQP since the algorithm is designed to search for local minimums only which are determined by the characteristics of a critical point (see also section 4.4 for further details). SQP would most likely find the best solution in that case if the initial values were close enough to the global optimum. Taking into account the extended strategies for globalizing the algorithm, it may be expected that the local minimum obtained through SQP is still satisfactory and fulfills the expectations towards a minimum energy consumption.

Considering example fig. 4.8, the corresponding energy consumption is:

$$\begin{aligned} \Delta E_{SQP} &= 0.11455\,\mathrm{kWh} \\ \Delta E_{opt} &= 0.11416\,\mathrm{kWh} \end{aligned}$$

Since the energy consumption induced by SQP is only $0.3\,\%$ higher compared to the optimal design, the nonlinear optimization can be regarded as acceptable and is confirmed for a further application on the combined temperature optimization task.

4.6 Simulation results

In this section various simulation results regarding the nonlinear temperature optimization through SQP shall be illustrated and compared. For this purpose different torque and speed profiles have been extracted from specified driving cycles, such as FTP75, AVL Urban DC, AVL Extraurban DC and AVL Highway DC. For simplification, but not for limitation to generality, the ability of recuperation was also admitted to the front EM, which means that negative torque values during a braking phase could also be distributed onto both EM.

Following settings have been applied to all simulations:

- Time discretization: $\Delta T = 1\,s$
- Initial optimization vector: $\boldsymbol{u_{f,0}} = [0.5]$
- Ambient temperature: $T_{amb} = 25\,°\mathrm{C}$

Consider fig. 4.9-4.12 each visualizing the optimal torque split factor for a given speed profile and on the right side the resulting optimum temperature trajectories for the stator of front and rear EM.

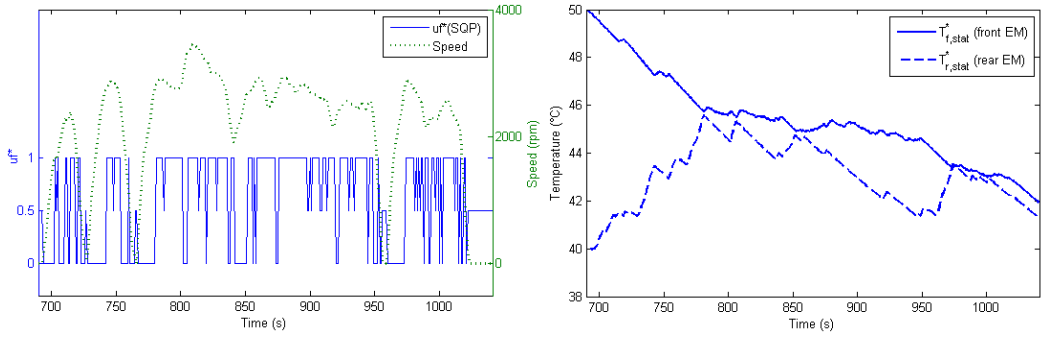

Figure 4.9: Nonlinear optimization through SQP performed on FTP75-DC between 690 - 1040 s with initial and constant temperature values (in °C): $[T_{f,stat,0},\ T_{r,stat,0},\ T_{cooling,0},\ T_{f,rot},\ T_{r,rot}] = [50, 40, 40, 60, 50]$.
Left: speed profile and optimized torque split during this time interval
Right: optimized front/rear stator temperature trajectories.

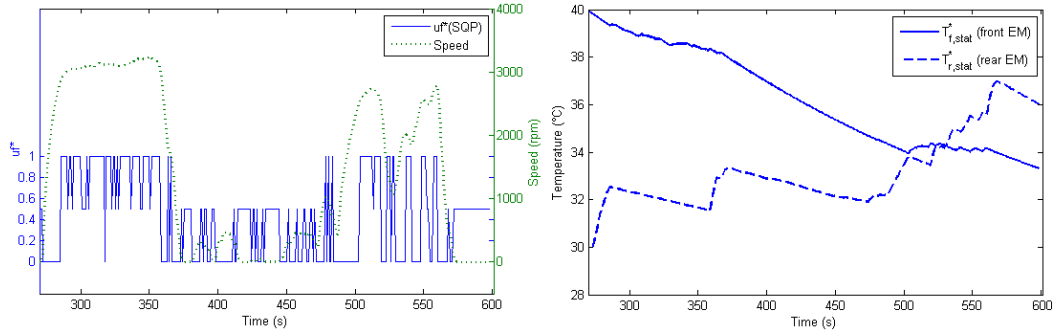

Figure 4.10: Nonlinear optimization through SQP performed on Urban-DC between 270 - 600 s with initial and constant temperature values (in °C): $[T_{f,stat,0},\ T_{r,stat,0},\ T_{cooling,0},\ T_{f,rot},\ T_{r,rot}] = [40, 30, 30, 35, 30]$.

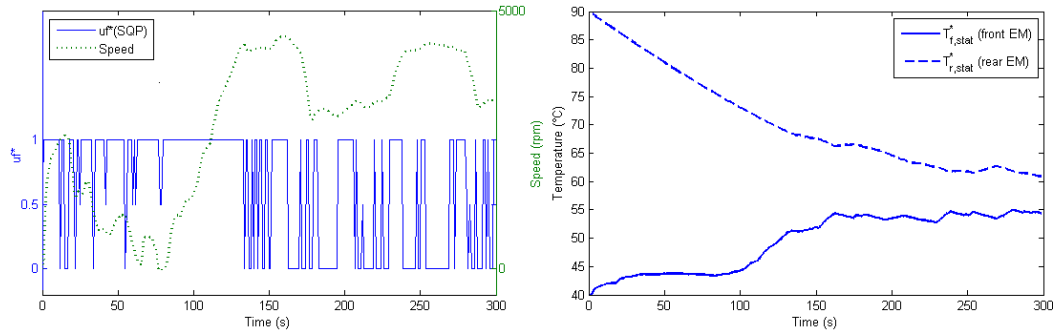

Figure 4.11: Nonlinear optimization through SQP performed on Extraurban-DC between 0 - 300 s with initial and constant temperature values (in °C): $[T_{f,stat,0},\ T_{r,stat,0},\ T_{cooling,0},\ T_{f,rot},\ T_{r,rot}] = [40, 90, 40, 40, 50]$.

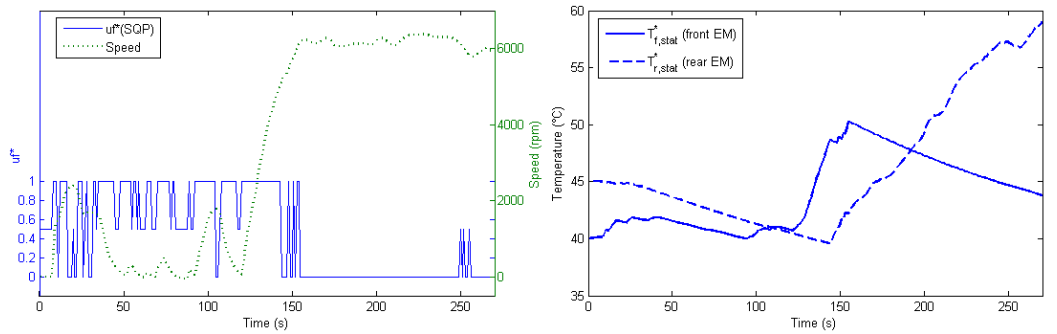

Figure 4.12: Nonlinear optimization through SQP performed on Highway-DC between 0 - 270 s with initial and constant temperature values (in °C): $[T_{f,stat,0},\ T_{r,stat,0},\ T_{cooling,0},\ T_{f,rot},\ T_{r,rot}] = [40, 45, 50, 50, 60]$.

The correspondent energy consumption J^* according to NLP (4.3) was calculated for all examples:

FTP75 (690 - 1040 s) ..... 0.403 kWh
Urban-DC (270 - 600 s) ..... 0.128 kWh
Extraurban-DC (0 - 300 s) .... 0.767 kWh
Highway-DC (0 - 270 s) ..... 0.948 kWh

A careful consideration of the temperature behavior in each example might reveal that during the nonlinear optimization process both trajectories of front and rear EM try to get closer and approach a common temperature level after some time. This observation can be supported by the fact that both EM implemented in the vehicle are build up identically providing the same power and efficiency for specified inputs at any time. Furthermore it can be noted that this process needs to be carried out sufficiently slowly since the attempt of using one EM exclusively at the beginning would cool down the unused EM quite rapidly, however, it would also cause a non marginal negative impact on the energy consumption due to possible inefficiencies of using one EM during this time.

Especially consider the examples of FTP75, Urban- and Extraurban-DC, where the initial stator temperature of front and rear EM differs significantly. Here, the harmonization of both trajectories can be figured out clearly. A closer investigation of fig. 4.11 on the left reveals, that up to 150 s the torque split factor u_f truly tends to stay mainly at 1 in order to cool down the rear EM, but also to ensure that the front EM is active when it is necessary.

After analyzing the results for Highway-DC in fig. 4.12 it can be observed, that both temperature trajectories of front and rear EM start to drift at around 150 s. This behavior is regarded as disadvantageous since additional energy consumption could be detected based on further simulations. As it was already mentioned in section 4.4, SQP implies the tracing of local minimums which usually cannot compensate the solution of a global optimum. Obviously the integration of a thermal behavior to the complex task of minimizing highly nonlinear energy functionals does not generate a good-natured optimization

problem, for which the objective of locating *good* local minimums is naturally a nontrivial task.

In summary, taking into account the thermal behavior of EM used to drive electric vehicles can still be regarded as effective in terms of improving the overall energy consumption. Given the fact that the efficiency of a PMSM generally decreases with temperature, non negligible impacts can be observed for system controllers which do not integrate thermal considerations in their algorithms.

A closer look on fig. 4.3 (sensitivity analysis) reveals, that a variation of the EM temperature in a broad range causes efficiency deviations of more than 5 %. Several simulations have shown that for the presented sections of driving cycles which have been visualized previously, energy improvements of usually 0.5 - 5 % could be measured comparing the results of ordinary torque split strategies where the impact of the EM thermal behavior is neglected. Naturally those improvements highly depend on the given road and driving profile causing different speed and torque demands over time which finally results in a variation of the EM's internal temperature.

5 Test track measurements

In this chapter a concluding review regarding the practical applicability of the proposed torque split strategies shall demonstrate the simulated benefits based on different test tracks in the real world. For this purpose the OpEneR prototype vehicle was used to collect measurement data of several test runs performed on dedicated courses of the Bosch proving ground in Boxberg/Germany. Especially for the evaluation of the applied torque split algorithms both handling courses on the test track have been selected to serve for further measurement analysis since they are provided with a variety of different curve radiuses, curve slopes, climbing gradients and downhill slopes. Consider fig. 5.1, where course No. 1 and 2 have been highlighted with dashed and solid lines.

Figure 5.1: Two different test tracks in Boxberg/Germany used for practical evaluation of the proposed torque split strategies.

The outer course No. 1 is designed for higher speeds and comprises curve radiuses between 54 m and 100 m. In contrast to this rather less demanding

test track, course No. 2 contains much tighter radiuses and in some cases the curves even feature an inward or an outward slope. It also includes sections representing mountain-pass driving with different gradients to show the effects of consecutive acceleration and deceleration phases.[1]

Test setup

Mainly to prove the efficiency benefits for the best suited hysteresis design in practice, following torque split strategies have been implemented in the vehicle's energy management software and tested on each course independently:

- 50:50 constant torque split (BM5)
- Suboptimal torque split (BM2)
- Threshold based hysteresis with $\alpha = 40$, $\beta = 10$ (BM13)

It is important to note, that clutch control with synchronization as mentioned in section 3.1 and 3.3 was active during all test runs.
For each strategy performed on both test tracks, measurement data of three repeating runs has been collected and saved for later analysis. In order to ensure maximum fairness between all conducted test runs which, it is necessary to require that vehicle and powertrain states as well as environmental conditions are formed by approximately the same initial values. For instance, these states include EM-temperature, battery SOC, battery temperature, weight of vehicle, weather conditions, ambient temperature, geographical starting point of the course, etc. Any deviation of initial values would have a different impact on the overall vehicle's behavior and thus, the total energy consumption.

It can be assumed that most of the previously mentioned initial states have been assigned accordingly for all test runs, so that possible deviations of energy efficiency are small enough to be neglected in the following. However, there is still a high impact on variation of energy consumption coming from the natural variance of the driving profile when all test runs are considered. During some sections the driver may speed up to slowly, in some cases braking is applied to hard, or even the starting time does not fit compared to

[1] cf. http://www.bosch-boxberg.de/en/30/index.asp (05/10/2013)

other test runs. In order to ensure *reproducibility* with respect to a fair evaluation of the proposed torque split modes as far as possible, it is necessary to exclude potential misleading driving sections from the further measurement analysis.

Due to these considerations, a time based evaluation of the speed profile is rather difficult in general and thus is replaced by a more useful and practical distance based analysis.

5.1 Distance based analysis

In order to compare velocity profiles of different runs in an efficient way, all provided measurements with time discretization of $T_d = 0.01$ s need to be converted to distance based values with a suitable discretization, e.g. $D_d = 0.5$ m. Besides the advantages of an easier comparison between measurements, many other benefits can be obtained when a converted analysis is applied. For instance, any arbitrary signal can be assigned to distance values which makes it possible to evaluate the driving behavior for dedicated sections on the road, and thus, giving clues about the energy consumption compared to other torque split strategies depending on the current distance from the starting point.

In the following a possible approach to eliminate misleading driving sections is described, which basically excludes outlying velocity trajectories from the measurement data and assigns mean power and energy values to this section instead. Consider fig. 5.2 illustrating all recorded velocity trajectories over distance for the investigated torque split modes and each run.

In the next step mean values and standard deviation for each distance can be computed and visualized in the same figure. Thereby, the constructed corridor formed by the mean trajectory and bounded by the estimated standard deviation of +/- 2 σ represents an acceptable area for the recorded measurement data. All outliers indicated by crosses in fig. 5.2 can now be excluded from further analysis.

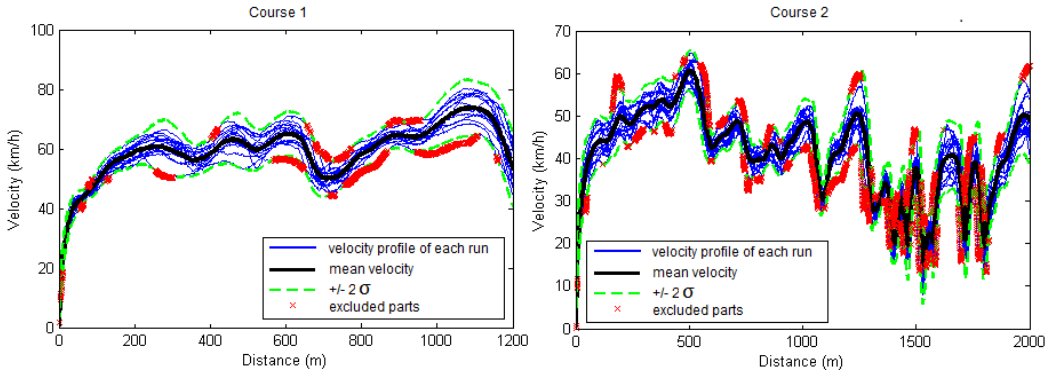

Figure 5.2: Velocity profiles for every run on both tracks including mean trajectory and estimated +/- 2 σ deviation. The crosses indicate sections which have to be excluded since they do not fit to the majority of all measurements.

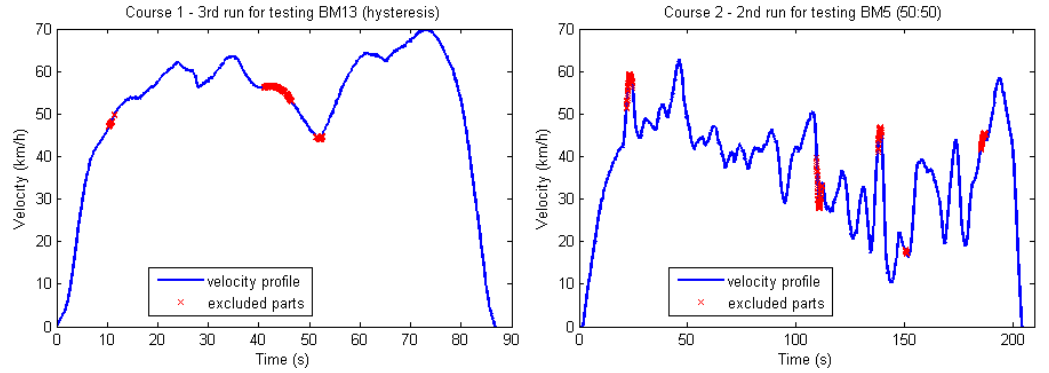

Figure 5.3: left: original velocity profile and excluded parts for the 3^{rd} run in hysteresis mode (BM13) on course 1.
right: 2^{nd} run with 50:50-constant split configuration (BM5) on course 2.

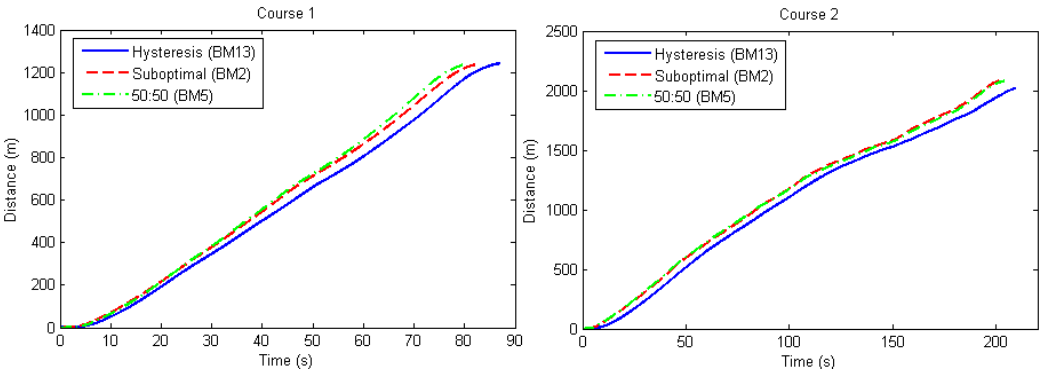

Figure 5.4: Relation between time and distance, averaged for each torque split mode

In fig. 5.3 exemplary trajectories for both courses are visualized in time domain, showing the excluded parts for a single measurement run.
The process of explicitly calculating the relationship between distance and time, again determined for the average values out of 3 runs for each torque split mode, can be regarded as helpful for further observations. Cf. fig 5.4 illustrating distance as a function of time for both courses and proposed modes.

5.2 Results

After removing outlying sections of the measurement data, all remaining parts can be regarded as acceptable since they do not deviate much from the average velocity. In order to guarantee a proper analysis in the end, power and energy behavior for all excluded sections need to be assigned to average values occurring within this calculated time and distance interval. If those parts had been simply neglected, the average energy consumption would have been distorted and a fair evaluation against other results would be simply prevented.

Finally consider fig. 5.5 showing the total electrical energy consumption for both courses and all three proposed torque split strategies. The distance based visualization of the physical energy consumption calculated in time domain is based on the relation of distance to time according to fig. 5.4.

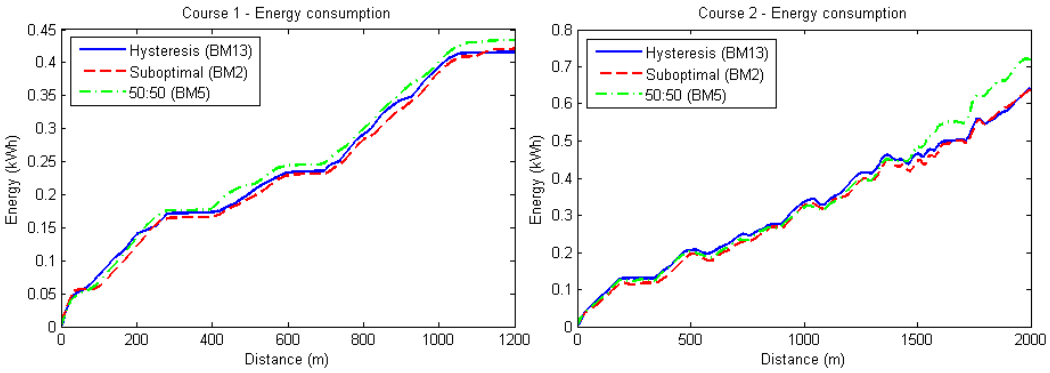

Figure 5.5: Average energy consumption over distance for both courses and different torque split modes.

	BM2 suboptimal	BM5 50:50 const.	BM13 hysteresis
Energy (kWh)	0.4182	0.4339	0.4131
$\text{Energy}_{imp,BM5}$ (%)	3.6	-	4.8
# switches	31	0	9
Duration (s)	85.3	81.9	89.7
# switches/s	0.36	0	0.1
min ΔT (s)	0.01	-	3.5

Table 5.1: Mean energy results and driving mode characteristics for major torque split strategies on course 1.

Obviously BM5 with a 50 % constant torque split distribution produced the worst results in terms of energy consumption for both courses. This observation complies with the simulation results presented in chapter 3 and confirms the application of different torque split strategies. Comparing the results in general, it can be seen that the energy improvement with respect to BM5 is much higher for course 2. Especially after 1500 m the behavior of the energy consumption for BM2 and BM13 stay close to each other whereas the trajectory of BM5 starts to drift away. According to fig. 5.2, this section contains short consecutive acceleration and braking maneuvers due to several serpentines along the road and affords to demonstrate the remarkable potential of the hysteresis design.

Finally, table 5.1 and 5.2 summarize the overall energy results and torque split characteristics separately for both courses. Note, that the energy consumption differs slightly from the ending values in diagram 5.5 since the recuperation impact caused by the final braking maneuver until vehicle standstill was included here.

In summary, applying the suboptimal or proposed threshold based hysteresis design to the vehicle's powertrain control software essentially improves the total energy consumption. The number of driving mode changes, but also the minimum time between two consecutive switches can be reduced drastically by using an appropriate hysteresis design approach which combines practical applicability with highest efficiency at any time.

As can be seen from the results presented in table 5.1 and 5.2, improvements

	BM2 suboptimal	BM5 50:50 const.	BM13 hysteresis
Energy (kWh)	0.6705	0.6906	0.6400
$\text{Energy}_{imp,BM5}$ (%)	2.9	-	7.32
# switches	87	0	26
Duration (s)	204.1	205.2	217.9
# switches/s	0.43	0	0.12
min ΔT (s)	0.01	-	2.13

Table 5.2: Mean energy results and driving mode characteristics for major torque split strategies on course 2.

of more than 7 % during 200 s of normal driving can be achieved compared to the results of a constant 50 % split distribution. Even the hysteresis design is able to beat the suboptimal approach in terms of energy consumption. The reason for this might be the wise suppress of unnecessary switches causing additional energy consumption due to the required synchronization phases of a possible unused electric machine at certain times. Note that the average velocity is slightly lower for BM13 due to the longer driving time, nevertheless all considered measurements are guaranteed to stay within the acceptable range as illustrated in fig. 5.2.

Since the promising results obtained by simulations in the previous part of this work entirely reflect the measurement results of various test runs presented in this chapter, it can be concluded that the proposed hysteresis torque split approach is regarded as absolutely suitable for a final integration into the vehicle's energy management system.

6 Conclusions and future work

In the present work several effective control strategies have been introduced to adapt the powertrain control software of a 4-WD electric vehicle in an energy efficient way. The proposed hysteresis design as well as appropriate implementations of a suitable predictive control algorithm offer sophisticated methods to distribute the current torque demand efficiently onto both EM during the drive.

For a complete understanding of the system itself the underlying model needs to integrate thermal considerations as well in order to obtain a more targeted representation of the physical system and to provide further adaptions to the developed control algorithms. Since the internal temperature of an electric machine has an additional effect on its efficiency, further improvements of the overall energy consumption can be achieved.

Within this work a control oriented approach towards the optimum utilization of both EM with respect of temperature variations has been conducted. This procedure comprises an *offline* calculation of the optimum torque profile for a predefined time interval, which can only be obtained if future torque and speed demands are available to the algorithm. For practical reasons this assumption is naturally hard to reach and requires at least smart sensors in the vehicle as well as additional telematics services from the environment. For instance combining information of in-car navigation systems with GPS data the upcoming road profile could be determined. Together with velocity regulations and/or current traffic conditions on the road, possible speed trajectories could be calculated and would finally provide the desired torque and speed profile to the presented control algorithm.

Another approach neglecting estimations of future conditions is to select the actual torque split factor based on current states of the vehicle. After obtaining the EM temperature through a suitable model or appropriate sensors, the optimum torque split factor for current conditions can be calculated according to the described procedure within this work. However, this method requires important information how to utilize both EM optimally during drive with respect to the behavior of their own internal temperature. This knowledge especially needs to be extracted out of dedicated studies e.g. as presented in chapter 4 and integrated in the correspondent control algorithm.

As a part of the vehicle's energy management system the developed control functionalities finally achieve significant improvements with respect to the overall energy consumption and consequently fulfill the presented objectives towards an increase of electrical range, safety and driving comfort.

Appendix

A Simulink model

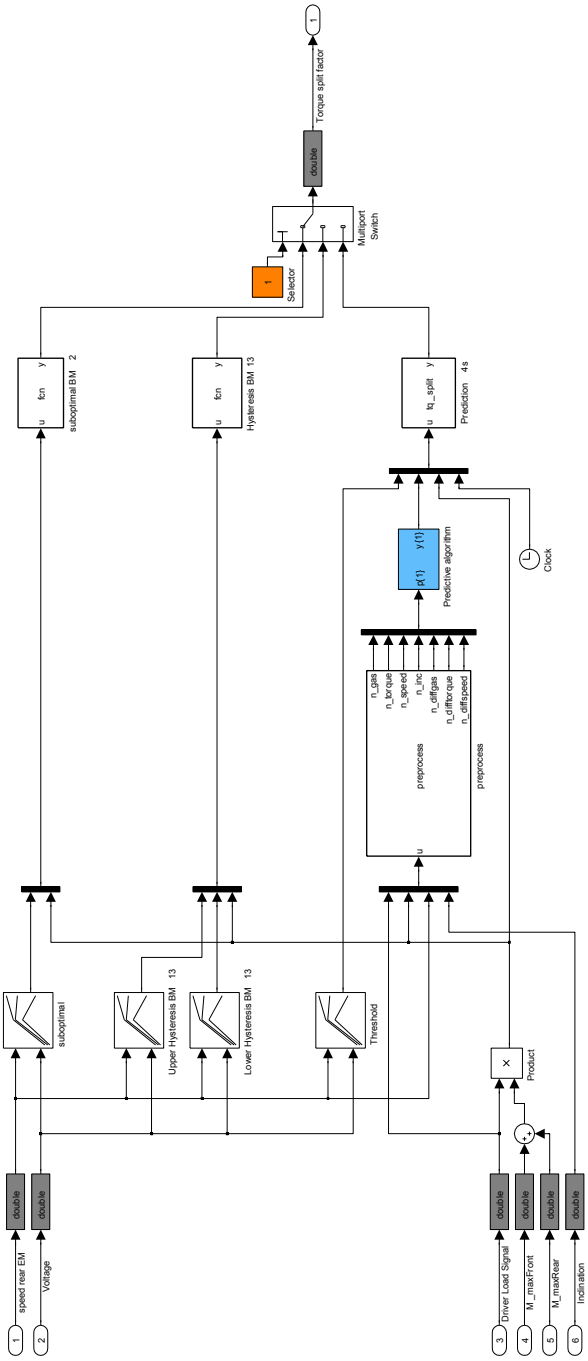

Figure A.1: Simulink model used within the torque split calibration methodology

B Benchmark results

- Basic design results
- Time hysteresis benchmark results
- Threshold hysteresis benchmark results
- Prediction results

Basic designs

NEDC: (1180s)

	BM1 (uf_opt)	BM2 (uf_subopt)	BM3 (1EM front)	BM4 (1EM rear)	BM5 (50% const)
ΔSOC [%]	3,939598894	3,940308983	3,970410204	3,969701794	4,221598828
$\Delta SOC_{imp\ [\%\ cmp.\ BM5]}$	6,679932083	6,663111682	5,95008277	5,966863357	0
#switches	39	28	0	0	0
#switches/s	0,033050847	0,023728814	0	0	0
min ΔT [s]	0.025 (=sim.step)	1,15	0	0	0

FTP75: (2477s)

	BM1 (uf_opt)	BM2 (uf_subopt)	BM3 (1EM front)	BM4 (1EM rear)	BM5 (50% const)
ΔSOC [%]	7,019788576	7,025828533	7,188973242	7,190712413	7,345751228
$\Delta SOC_{imp\ [\%\ cmp.\ BM5]}$	4,437431131	4,355207314	2,134267567	2,110591694	0
#switches	273	128	0	0	0
#switches/s	0,110213969	0,051675414	0	0	0
min ΔT [s]	0,025 (=sim.step)	0,05	0	0	0

Urban: (1073.3s)

	BM1 (uf_opt)	BM2 (uf_subopt)	BM3 (1EM front)	BM4 (1EM rear)	BM5 (50% const)
ΔSOC [%]	2,313715022	2,311676394	2,405158592	2,405370911	2,424762277
$\Delta SOC_{imp\ [\%\ cmp.\ BM5]}$	4,579717199	4,663792567	0,808478634	0,799722376	0
#switches	419	158	0	0	0
#switches/s	0,390384795	0,147209541	0	0	0
min ΔT [s]	0,025 (=sim.step)	0,025	0	0	0

Extra-Urban: (2449.3s)

	BM1 (uf_opt)	BM2 (uf_subopt)	BM3 (1EM front)	BM4 (1EM rear)	BM5 (50% const)
ΔSOC [%]	7,713499004	7,711927063	7,797244862	7,796646736	8,073032469
$\Delta SOC_{imp\ [\%\ cmp.\ BM5]}$	4,453511947	4,472983452	3,416158773	3,423567711	0
#switches	458	230	0	0	0
#switches/s	0,186992202	0,093904381	0	0	0
min ΔT [s]	0,025 (=sim.step)	0,025	0	0	0

Highway: (1069.8s)

	BM1 (uf_opt)	BM2 (uf_subopt)	BM3 (1EM front)	BM4 (1EM rear)	BM5 (50% const)
ΔSOC [%]	11,38247695	11,33107167	11,41732132	11,41747963	12,01765708
$\Delta SOC_{imp\ [\%\ cmp.\ BM5]}$	5,285390713	5,713138609	4,995447555	4,994130286	0
#switches	182	86	0	0	0
#switches/s	0,170125257	0,080388858	0	0	0
min ΔT [s]	0,025 (=sim.step)	0,025	0	0	0

Time based hysteresis designs

NEDC: (1180s)

	BM6(T1=T2=30s)	BM7 (T1=T2=20s)	BM8 (T1=T2=10s)	BM9 (T1=T2=5s)	BM10 (T1=T2=2s)	BM11(T1=T2=1s)	BM12 (T = 0s)
ΔSOC [%]	3,990376264	3,980203107	3,953210139	3,942691694	3,939946594	3,939946594	3,940308983
$\Delta SOC_{imp\ [\%\ cmp.\ BM5]}$	5,47713257	5,718111327	6,357512883	6,606670719	6,671695849	6,671695849	6,663111682
#switches	20	26	28	28	28	28	28
#switches/s	0,016949153	0,022033898	0,023728814	0,023728814	0,023728814	0,023728814	0,023728814
min ΔT [s]	30	20	10	5	3,025	3,025	1,15

FTP75: (2477s)

	BM6(T1=T2=30s)	BM7 (T1=T2=20s)	BM8 (T1=T2=10s)	BM9 (T1=T2=5s)	BM10 (T1=T2=2s)	BM11(T1=T2=1s)	BM12 (T = 0)
ΔSOC [%]	7,137669868	7,092469097	7,035129412	7,031453644	7,027690422	7,026001165	7,025828533
$\Delta SOC_{imp\ [\%\ cmp.\ BM5]}$	2,832676383	3,448008561	4,228591557	4,278630933	4,329860838	4,352857231	4,355207314
#switches	38	48	58	82	110	124	128
#switches/s	0,015341138	0,01937828	0,023415422	0,033104562	0,044408559	0,050060557	0,051675414
min ΔT [s]	30	20	10	5	2	1	0,05

Urban: (1073.3s)

	BM6(T1=T2=30s)	BM7 (T1=T2=20s)	BM8 (T1=T2=10s)	BM9 (T1=T2=5s)	BM10 (T1=T2=2s)	BM11(T1=T2=1s)	BM12 (T = 0)
ΔSOC [%]	2,37753372	2,355235919	2,352970888	2,333117728	2,323279513	2,315016095	2,311676394
$\Delta SOC_{imp\ [\%\ cmp.\ BM5]}$	1,947760315	2,867347407	2,960759911	3,779527179	4,185266525	4,52605947	4,663792567
#switches	27	37	53	79	111	129	158
#switches/s	0,025156061	0,03447312	0,049380416	0,07360477	0,103419361	0,120190068	0,147209541
min ΔT [s]	30	20	10	5	2	1	0,025

Extra-Urban: (2449.3s)

	BM6(T1=T2=30s)	BM7 (T1=T2=20s)	BM8 (T1=T2=10s)	BM9 (T1=T2=5s)	BM10 (T1=T2=2s)	BM11(T1=T2=1s)	BM12 (T = 0)
ΔSOC [%]	7,795132114	7,765958446	7,746372211	7,745019092	7,72470381	7,714444859	7,711927063
$\Delta SOC_{imp\ [\%\ cmp.\ BM5]}$	3,442329211	3,803701086	4,046314193	4,063075167	4,314718913	4,441795714	4,472983452
#switches	41	55	79	112	168	188	230
#switches/s	0,016739477	0,022455395	0,032254113	0,045727351	0,068591026	0,076756624	0,093904381
min ΔT [s]	30	20	10	5	2	1	0,025

Highway: (1069.8s)

	BM6(T1=T2=30s)	BM7 (T1=T2=20s)	BM8 (T1=T2=10s)	BM9 (T1=T2=5s)	BM10 (T1=T2=2s)	BM11(T1=T2=1s)	BM12 (T = 0)
ΔSOC [%]	11,41031537	11,50876989	11,44815876	11,4108952	11,37482809	11,3802781	11,33107167
$\Delta SOC_{imp\ [\%\ cmp.\ BM5]}$	5,053744714	4,234495845	4,738846454	5,048919856	5,349037583	5,303687543	5,713138609
#switches	15	19	25	33	51	61	86
#switches/s	0,014021312	0,017760329	0,023368854	0,030846887	0,047672462	0,057020004	0,080388858
min ΔT [s]	30	20	10	5	2	1	0,025

Threshold based hysteresis designs

NEDC: (1180s)

	BM13 (α=40, β=10)	BM14 (α=β=30)	BM15 (α=20, β=2)	BM16 (α=β=10)	BM17 (α=10, β=2)	BM18 (α=5, β=1)	BM19 (α=β=2)	BM20 (α=0, β=0)
ΔSOC [%]	3,950053216	3,943248533	3,941820782	3,941439441	3,941061877	3,940698285	3,940269049	3,940308983
$\Delta SOC_{imp\ [\%\ cmp.\ BM5]}$	6,432293146	6,593480496	6,627300632	6,636333742	6,645277348	6,653890012	6,664057643	6,663111682
#switches	18	26	26	26	26	28	28	28
#switches/s	0,015254237	0,022033898	0,022033898	0,022033898	0,022033898	0,023728814	0,023728814	0,023728814
min ΔT [s]	25,9	4,15	4,1	4,1	4,1	1,1	1,1	1,15

FTP75: (2477s)

	BM13 (α=40, β=10)	BM14 (α=β=30)	BM15 (α=20, β=2)	BM16 (α=β=10)	BM17 (α=10, β=2)	BM18 (α=5, β=1)	BM19 (α=β=2)	BM20 (α=0, β=0)
ΔSOC [%]	7,025617157	7,031487632	7,017513673	7,022670745	7,023917114	7,018591349	7,023550231	7,025828533
$\Delta SOC_{imp\ [\%\ cmp.\ BM5]}$	4,358084848	4,278168237	4,468400095	4,398195266	4,381228065	4,453729354	4,386222551	4,355207314
#switches	64	66	78	88	98	106	118	128
#switches/s	0,025837707	0,026645135	0,031489705	0,035526847	0,039563989	0,042793702	0,047638272	0,051675414
min ΔT [s]	1,4	1,675	0,85	0,5	0,175	0,2	0,3	0,05

Urban: (1073.3s)

	BM13 (α=40, β=10)	BM14 (α=β=30)	BM15 (α=20, β=2)	BM16 (α=β=10)	BM17 (α=10, β=2)	BM18 (α=5, β=1)	BM19 (α=β=2)	BM20 (α=0, β=0)
ΔSOC [%]	2,320771321	2,320202389	2,316523556	2,31290731	2,314806028	2,312954983	2,312990952	2,311676394
$\Delta SOC_{imp\ [\%\ cmp.\ BM5]}$	4,288707257	4,31217068	4,463890014	4,613028191	4,534722875	4,611062094	4,609578676	4,663792567
#switches	37	41	55	120	122	130	146	158
#switches/s	0,03447312	0,038199944	0,051243827	0,111804714	0,113668126	0,121121774	0,136029069	0,147209541
min ΔT [s]	1,4	1,35	0,325	0,1	0,1	0,11	0,05	0,0025

Extra-Urban: (2449.3s)

	BM13 (α=40, β=10)	BM14 (α=β=30)	BM15 (α=20, β=2)	BM16 (α=β=10)	BM17 (α=10, β=2)	BM18 (α=5, β=1)	BM19 (α=β=2)	BM20 (α=0, β=0)
ΔSOC [%]	7,720555412	7,720025559	7,711255546	7,712454097	7,710519517	7,712178123	7,712841725	7,711927063
$\Delta SOC_{imp\ [\%\ cmp.\ BM5]}$	4,366104786	4,372668037	4,481301472	4,46645512	4,49041861	4,469873588	4,461653605	4,472983452
#switches	49	57	83	124	130	152	186	230
#switches/s	0,020005716	0,023271955	0,033887233	0,05062671	0,053076389	0,062058547	0,075940065	0,093904381
min ΔT [s]	1,625	0,975	0,15	0,075	0,125	0,025	0,025	0,025

Highway: (1069.8s)

	BM13 (α=40, β=10)	BM14 (α=β=30)	BM15 (α=20, β=2)	BM16 (α=β=10)	BM17 (α=10, β=2)	BM18 (α=5, β=1)	BM19 (α=β=2)	BM20 (α=0, β=0)
ΔSOC [%]	11,39005651	11,381583	11,38458621	11,38150196	11,37205204	11,38033005	11,37216222	11,33107167
$\Delta SOC_{imp\ [\%\ cmp.\ BM5]}$	5,222320499	5,292829356	5,267839338	5,293503666	5,372137277	5,30325527	5,371220492	5,713138609
#switches	15	11	27	40	48	54	58	86
#switches/s	0,014021312	0,010282296	0,025238362	0,037390166	0,0448682	0,050476725	0,054215741	0,080388858
min ΔT [s]	1,95	1,875	0,475	0,175	0,175	0,175	0,05	0,025

Predictive designs based on updated model

FTP75: (2477s)

	BM2 (uf_subopt)	BM5 (50% const)	Prediction (2 s)	Prediction (4 s)	Prediction (6 s)	Perfect prediction (4 s)
ΔSOC [%]	7,088736132	7,345091374	7,100467631	7,07227357	7,380228606	7,064338848
$\Delta SOC_{imp\ [\%\ cmp.\ BM5]}$	3,490157287	0	3,330438393	3,714287414	-0,478377056	3,82231496
#switches	156	0	104	78	52	64
#switches/s	0,062979411	0	0,041986274	0,031489705	0,020993137	0,025837707
min ΔT [s]		0	2	4	6	4

Urban: (1073.3s)

	BM2 (uf_subopt)	BM5 (50% const)	Prediction (2 s)	Prediction (4 s)	Prediction (6 s)	Perfect prediction (4 s)
ΔSOC [%]	2,333325621	2,42485519	2,331427838	2,33647802	2,361444745	2,331743733
$\Delta SOC_{imp\ [\%\ cmp.\ BM5]}$	3,774640648	0	3,852904388	3,644637061	2,61501988	3,839877033
#switches	172	0	71	83	71	49
#switches/s	0,160253424	0	0,066151123	0,077331594	0,066151123	0,045653592
min ΔT [s]	0,025	0	2	4	6	4

Extra-Urban: (2449.3s)

	BM2 (uf_subopt)	BM5 (50% const)	Prediction (2 s)	Prediction (4 s)	Prediction (6 s)	Perfect prediction (4 s)
ΔSOC [%]	7,770795041	8,072430909	7,795132114	7,749090582	7,753460566	7,735028526
$\Delta SOC_{imp\ [\%\ cmp.\ BM5]}$	3,736617527	0	3,435133705	4,005488941	3,951354268	4,179687467
#switches	336	0	166	82	60	60
#switches/s	0,137182052	0	0,067774466	0,033478953	0,024496795	0,024496795
min ΔT [s]	0,025	0	2	4	6	4

Highway: (1069.8s)

	BM2 (uf_subopt)	BM5 (50% const)	Prediction (2 s)	Prediction (4 s)	Prediction (6 s)	Perfect prediction (4 s)
ΔSOC [%]	11,38137149	12,00468333	11,73096812	11,46904323	11,72563447	11,41335118
$\Delta SOC_{imp\ [\%\ cmp.\ BM5]}$	5,192238923	0	2,280070251	4,461926133	2,324499972	4,925845458
#switches	234	0	18	15	37	19
#switches/s	0,218732473	0	0,016825575	0,014021312	0,034585904	0,017760329
min ΔT [s]	0,025	0	2	4	6	4

C Driving cycles

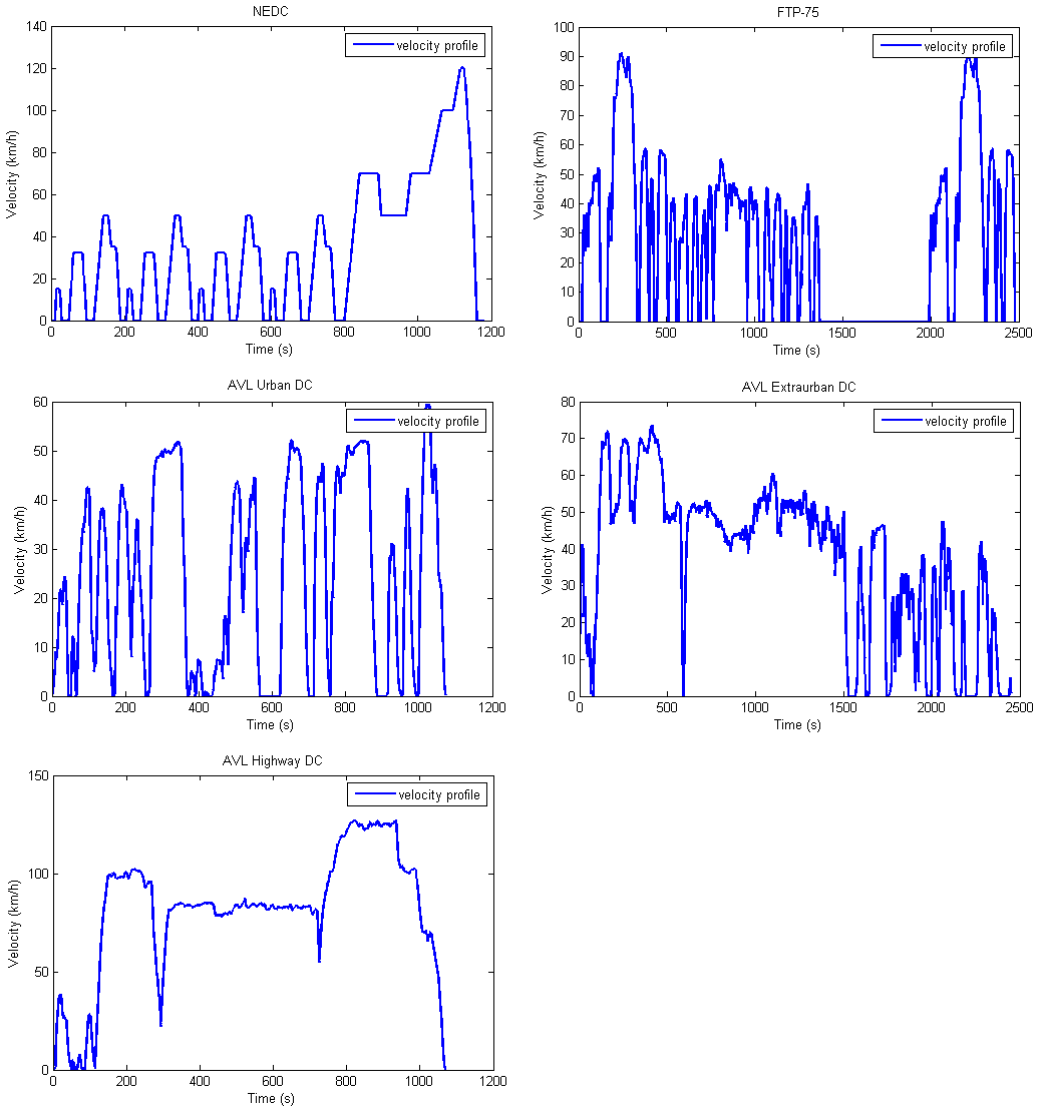

Figure C.1: Velocity profile of all investigated driving cycles

Bibliography

[Acq12] Acquaviva. *Analytical Modeling of Iron Losses for a PM Traction Machine*. KTH Stockholm, Royal Institute of Technology. Mar. 2012 (cit. on pp. 11, 65).

[Bis06] Bishop. *Pattern recognition and Machine Learning*. Springer Verlag. 2006 (cit. on pp. 39–45, 47, 48).

[BJ] Bettayeb and Jannot. *Analytical Calculation of Rotor-Magnet Eddy-Current Losses for High Speed IPMSM*. URL: `http://ieeexplore.ieee.org` (visited on 03/20/2013) (cit. on p. 65).

[Bos] Bosch. *Kraftfahrzeugtechnik*. URL: `http://www.bosch-presse.de/presseforum/details.htm?txtID=5291` (visited on 03/09/2013) (cit. on p. 13).

[Eur09] European Parliament. *Verordnung (EG) Nr. 443/2009*. Apr. 23, 2009. URL: `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2009:140:0063:008:de:PDF` (visited on 12/02/2012) (cit. on p. 1).

[GK] Ganchev and Kral. *Sensorless Rotor Temperature Estimation of PMSM*. URL: `http://publik.tuwien.ac.at/files/PubDat_203021.pdf` (visited on 03/13/2013) (cit. on p. 58).

[Gri11] Grießler. *Fahrstrategieoptimierung bei Nutzfahrzeugen mit Hilfe vorrauschauender Informationen*. AVL List GmbH / Johannes Kepler Universität Linz, Austria. Apr. 2011 (cit. on p. 10).

[GS07] Guzzella and Sciarretta. *Vehicle Propulsion Systems*. Springer Verlag. 2007 (cit. on p. 12).

[GSB11] Götz, Sunderer, and Birzle-Harder. *Attraktivität und Akzeptanz von Elektroautos*. Institut für sozial-ökologische Forschung, Frankfurt. Oct. 2011 (cit. on p. 2).

[Haj] Hajek. *Neural Networks*. URL: `http://www.cs.ukzn.ac.za/notes/NeuralNetworks2005.pdf` (visited on 01/13/2013) (cit. on pp. 37, 38, 40, 42).

[Hof04] Hofer. *Computerunterstützte Modellbildung und Simulation, lecture notes*. Institut für Regelungstechnik, TU Graz. 2004 (cit. on p. 68).

[Höm] Hömberg. *Nichtlineare Optimierung*. URL: http://www3.math.tu-berlin.de/Vorlesungen/WS05/NonLinOpt/skript/nonlin-opt.pdf (visited on 03/29/2013) (cit. on p. 73).

[Kal] Kalmbach. *Effiziente Ableitungsbestimmung bei hochdimensionaler nichtlinearer Optimierung*. URL: http://elib.suub.uni-bremen.de/edocs/00102087-1.pdf (visited on 03/29/2013) (cit. on pp. 70, 72, 76).

[Kit10] Kitanoski. *Modellbildung, Simulation und optimale Steuerung des Kühlsystems eines Hybridfahrzeugs*. Graz University of Technology, Austria. June 2010 (cit. on p. 64).

[Kug] Kugi. *Optimierung - Vorlesung u. Übung WS 2012/13*. URL: http://www.acin.tuwien.ac.at/fileadmin/cds/lehre/opt/WS2012/Optimierung_WS2012.pdf (visited on 03/29/2013) (cit. on pp. 72, 74–76).

[Maa08] Maass. *Computational Intelligence, lecture notes*. Institut für Grundlagen der Informationsverarbeitung, TU Graz. Mar. 2008 (cit. on pp. 37, 40, 42, 44, 47, 48).

[Mas12] Massoner. *Optimum Energy Management Strategies for an Electric Vehicle Integrated in an Intelligent Transport System*. AVL Graz/Institut für Regelungstechnik, TU Graz. Dec. 2012 (cit. on pp. 15, 17).

[OpE] OpEneR. *project homepage*. URL: http://www.fp7-opener.eu/ (visited on 12/05/2012) (cit. on pp. 3, 6).

[Raj06] Rajamani. *Vehicle Dynamics and Control*. Springer Verlag. 2006 (cit. on pp. 8, 9).

[SB] Specht and Böcker. *Observer for the Rotor Temperature of IPMSM*. URL: wwwlea.uni-paderborn.de/fileadmin/Elektrotechnik/AG-LEA/forschung/veroeffentlichungen/2010/Observer_for_the_Rotor_Temperature_of_IPMSM.pdf (visited on 03/20/2013) (cit. on p. 65).

[Sch09] Schwingshackl. *Simulation von elektrischen Fahrzeugkonzepten für PKW*. Graz University of Technology, Austria. May 2009 (cit. on p. 1).

[Sta] Statistik Austria. *Kraftfahrzeuge, Statistiken Neuzulassungen und Bestand*. URL: http://www.statistik.at/ (visited on 12/02/2012) (cit. on pp. 1, 2).

[Uni] United Nations. *Framework Convention on Climate Change*. URL: http://unfccc.int/resource/docs/publications/infokit_1999_ge.pdf (visited on 12/02/2011) (cit. on p. 1).

[ZAS] Zarei, Abbaszadeh, and Safari. *The Analytical Analysis of the Rotor Losses in the PMSM Motors*. URL: http://www.iaeng.org/publication/WCECS2012/WCECS2012_pp1044-1048.pdf (visited on 03/20/2013) (cit. on pp. 65, 66).

[ZHP] Zhang, Hu, and Patuwo. *Forecasting with artificial neural networks*. URL: http://www.ic.unicamp.br/~wainer/cursos/mo416/forecast-nn.pdf (visited on 01/20/2013) (cit. on pp. 44, 45, 47–49).