Real-time Energy Management of a Battery Electric Vehicle Hybridized with Supercapacitor

by

Parisa Golchoubian

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2017

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The increased interest in electric vehicles (EVs) in the recent years has intrigued numerous research, on improving efficiency and reducing ownership costs of these vehicles. As the battery in EVs is the sole energy provider, it is exposed to degradation due to high peaks and rapid fluctuations in the power demanded by the driver. Therefore, integrating a supercapacitor (SC) pack into the energy storage system of these vehicles has been proposed as a potential solution; maintaining the battery as the main energy source of the vehicle while using the SC when exposed to high power peaks and power fluctuations. However, just like any other hybrid system, the maximum benefit of this integration can only be exploited when applying a proper energy management controller. Various energy management controllers have been used for these systems through the literature; ranging from simple rule based control strategies to more complex optimal control approaches.

In this thesis, nonlinear model predictive control (NMPC) strategies have been designed as energy management controllers for battery-SC hybrid energy storage systems (HESSs) in a Toyota Rav4EV. Although traditionally used in applications dealing with slow dynamics like process control, with the rapid improvement in electric control units (ECUs) in the recent years, NMPCs have received a great deal of attention in areas with systems of faster dynamics, including the automotive sector. However, the question still needs to be addressed whether NMPC can demonstrate performance improvement over other state-of-the-art controllers, while maintaining computational efficiency necessary for automotive real-time applications.

This investigation has been conducted through Model-in-the-Loop (MIL) simulating and Hardware-in-the-Loop (HIL) testing on the NMPC energy management strategies designed in this work. The NMPC uses a control-oriented model of the system, some form of the future trip prediction, and an optimization solver to find the optimal power split between the battery and SC at each time step during the trip. The designed NMPC has been compared to other state-of-the-art controllers in the literature. A number of methods for future trip prediction have also been studied through the thesis and the NMPC shows to outperform other controllers even with no prior knowledge of the future trip whatsoever.

The results obtained through HIL testing on a dSPACE ECU indicate that upon carefully choosing the prediction and control horizon length, as well as the maximum number of iterations allowed, the execution time for NMPC falls far below the necessary sampling time of 10 milliseconds in vehicle control. The correlation between each of these parameters and turn-around time have been presented; constructing a benchmark for NMPC design.

Acknowledgments

I would like to thank my supervisor, Prof. Nasser L. Azad for his motivation and guidance throughout this research. One could not ask for a more supportive and understanding supervisor.

I would also like to thank my colleagues and friends, Mahyar Vajedi, Mohit Batra, Amer Keblawi and Joydeep Banerjee for their technical support through different stages of this work.

Special thanks to my readers Prof. Mehrdad Kazerani and Prof. Shoja'eddin Chenouri for their valuable guidance in areas that laid outside my field of expertise.

I thank Dr. Ken Butts and Dr. Josh Payne for their insightful comments and suggestions for improving the quality of this research.

I acknowledge NSERC and Toyota for Financially supporting this research.

I'm grateful to all my friends in the SHEVS lab and Motion Research Group for providing a warm, pleasant environment that made research time at the office memorable.

At last but not least, I would like to thank my wonderful husband Ali, whom I couldn't have done this without his emotional uplifting support and my loving parents who never hesitated to provide me with whatever I needed to peruse my dreams.

Dedication

To my best friend and husband, Ali, for his endless love and support and my loving family whom their distance fueled my effort to make bearing this separation worthwhile.

Table of Contents

List of Tables

List of Figures

Nomenclature

Abbreviations

BFGS	Broyden-Fletcher-Goldfarb-Shanno
CAN	Controller Area Network
D-NMPC	Deterministic Nonlinear Model Predictive Control
DP	Dynamoc Programming
ECU	Electronic Control Unit
EMC	Energy Management Controller
EMPC	Explecit Model Predictive Control
EV	Electric Vehicle
EVPI	Expected Value of Perfect Information
FT-MPC	Frozen-Time Model Predictive Control
GPS	Global Positioning System
HESS	Hybrid Energy Storage System
HIL	Hardware-in-the-Loop
ITS	Intelligent Transportation System
LMPC	Linear Model Predictive Control
MIL	Model-in-the-Loop

MPC	Model Predictive Control
NLP	Nonlinear Prgramming
NMPC	Nonlinear Model Predictive Control
P-MPC	Prescient Model Predictive Control
PC	Personal Computer
PID	Proportional-Integral-Derivative
QP	Quadratic Programming
RBC	Rule-based Control
RMPC	Robust Model Predictive Control
RMS	Root Mean Square
S-NMPC	Stochastic Nonlinear Model Predictive Control
SC	Supercapacitor
SDP	Stochastic Dynamic Programming
SOC	State of Charge
SQP	Sequential Quadratic Programming
SS	Stochastic Solution
TMMC	Toyota Motor Manufacturing Canada
TP-MPC	Trip-Planning Model Predictive Control
TPM	Transition Probability Matrix
VMS	Vehicle Measurement System
WSS	Wait and See Solution

Common Symbols

η_c	Converter Efficiency

I_b	Battery Current
$I_{b,RMS}$	Battery Current Root Mean Square
K_{max}	Maximum Number of Iterations
N_c	Control Horizon
N_p	Prediction Horizon
N_S	Number of Scenarios
P_b	Battery Power
P_d	Power Demand
P_{SC}	Supercapacitor Power
SOC_b	Battery State of Charge
SOC_{SC}	Supercapacitor State of Charge
T_c	Discretizing Time Constant

Chapter 1

Introduction

According to NASA and NOAA, 2015 has been marked as the warmest year worldwide since modern record keeping began in 1880 [2]. The transportation sector is considered the second largest contributor of U.S. greenhouse gas emission after the electricity sector, and according to the U.S. EPA [3], more than half of this sector accounts for passenger cars. As a result, electric vehicles (EVs) have received a great deal of attention as one of the most promising solutions in reducing exhaust emissions.

1.1 Motivation and Challenges

As many issues regarding EVs have been resolved and EVs have hit commercialization in the recent years, research has been mostly conducted to improve efficiency and reduce the purchase and ownership costs of these vehicles.

In current EVs, the power demanded by the driver is solely provided by the battery pack; hence, all the high power peaks and fluctuations due to drivers' behavior and route condition are handled by the battery. These frequent power surges into and out of the battery normally come at the cost of heating the battery and reducing the battery lifespan and efficiency. As a result, many have proposed the potential solution of integrating SCs into the energy storage system of EVs, in order to take advantage of the functionalities of both SCs and batteries [4, 5]. Doing so, the battery stays as the primal energy source of the vehicle, while the SC assists the battery during high power peaks and power fluctuations. While batteries can only handle a limited number of charge/discharge cycles, before their capacity falls under 80 percent of nominal value [6], SCs have a cycle-life 1000 times that of batteries [7]. SCs also have a lower internal resistance, causing less power loss for the system when using the SC as much as possible.

Although, all these characteristics of SCs make them ideal to be assigned as assistants for batteries in EVs, the maximum benefit of these HESSs could only be exploited when a proper energy management controller (EMC) is employed. The EMC defines the best power split at each time instant between the battery and SC in order to achieve maximum performance and battery health.

Various energy management strategies for the battery-SC HESSs in EVs have been previously proposed; ranging from rule-based control (RBC) to more complected optimal control approaches. However, there still exists a great gap between the potential maximum benefit of these systems and what EMCs available in the literature are able to obtain in real-time.

Model Predictive Control (MPC) is a state of the art control method that aims to optimize a current action of interest, by taking future events into account. The concept is usually compared to a driver's brain, where the driver observes a turn in the road in the near future and adjusts its current speed accordingly. MPC similarly predicts the future states and outputs of the system using a simple dynamic model of it and settles on a set of actions that optimize the objective of interest. The method has shown to be superior to other rule-based methods in many areas of automotive control [8, 9] due to the fact that it takes future road and traffic information into account for its optimal decision making. This, however, demonstrates the importance of accurate future road and traffic prediction on the performance of the model protective controller. Methods have been proposed in the literature for estimating these future conditions [10] using either external technologies such as Global Positioning System (GPS), Intelligent Transportation System (ITS), internet maps and real-time traffic data and/or using data form the past trips in the form of a Markov process or Machine learning algorithm [11]. As exact knowledge of the future road and traffic conditions a priori is practically almost impossible due to many disturbances on the system, modifications on the classic MPC method have been made to improve the control robustness and allow the system to work with stochastic knowledge of the future road and traffic conditions [12].

Other than accuracy in predicting future road and traffic conditions, accuracy in the behavior of the dynamic model can also significantly impact the controller performance. The use of a nonlinear model can obviously better capture the behavior of the system in comparison to a linear model. By using a nonlinear model inside the MPC controller, a Nonlinear MPC (NMPC) is obtained.

Despite the superiority MPC approaches hold in performance, the required computational cost for these methods makes them challenging to be implemented for systems with fast dynamics such as those in the automotive sector. This challenge gets only more critical when more complex extensions like NMPC and stochastic MPC are employed. This paves the way for state of the art, fast and efficient nonlinear programming solvers, to take over the optimization hassle in NMPCs and make real-time implementation of these methods possible.

In this thesis, a real-time NMPC method is proposed as the energy management controller of a battery-SC HESS in the Toyota Rav4EV. The practicality and performance of a stochastic NMPC method is also investigated for these systems.

1.2 Problem Statement and Proposed Approach

The goal of this research is to design an NMPC based energy management controller for the battery-SC HESS in order to maximize the efficiency and battery health. The controller needs to outperform the available controllers in the literature and be implementable in real-time. To demonstrate implementation practicality, the proposed controller has been fine-tuned for a commercial Toyota Rav4EV. The following set of actions have been taken for control design and evaluation:

Control-oriented Modeling: To improve the performance of the predicative controller from those mainly available through the literature, a nonlinear control-oriented model has been developed and used. The control-oriented model needs adequately capture the dynamic behavior of the system, yet to be fast enough for real-time response. Therefore, in this research a nonlinear dynamic model of the battery pack and SC have been used as the control-oriented model.

Estimation of Future Driving Condition: As the controller performance is highly dependent on the prediction quality of the future driving conditions, evaluations have been made for different levels of prior knowledge of the future driving conditions. Stochastic knowledge of the future traffic and road conditions have been employed in the form of a Markovian process for future condition estimation.

Control Design: An NMPC energy management controller is accordingly designed with the objective of maximizing the battery health and system efficiency with respect to the allowable performance range of the battery and SC. A nonlinear optimization problem is continuously solved over a receding horizon at each sampling time, in order to obtain the optimal controls. For implementing stochastic NMPC, the optimization problem is modified to enable employing stochastic knowledge of the future driving conditions. Finally, several weighting factors are tuned accordingly to improve system performance.

Nonlinear Programming Solver Development: In order to allow real-time optimization, a nonlinear programming solver is coded and used online. A shooting based, sequential quadratic programming (SQP) method is used as the real-time optimization solver of the NMPC. The real-time implementation capability of the coded solver is investigated through HIL testing.

Control Evaluation: After all aspects of the controller are developed, the performance and real-time implementation capability of the controller needs to be evaluated. This has been done using two main strategies; MIL simulations and HIL testing.

In MIL simulations, the performance of energy management controllers are validated using a high-fidelity model of the system. The high-fidelity model includes a more complicated and realistic model of the system with respect to what is used in the controller. Using MIL simulations, the performance of the designed controller is compared to other energy management systems.

HIL testing is afterwards performed for computational speed evaluation. The proposed controller is embedded on an electronic control unit (ECU) and the real-time implementation capability of the controller is tested. HIL testing demonstrates the implementation capability of the proposed controller on commercial hardware.

1.3 Thesis Layout

The thesis is organized in the following 6 chapters. Chapter 2 provides a brief description and literature review on some essential concepts used throughout this thesis. Chapter 3 covers high-fidelity and control-oriented modeling for the system of interest. Chapter 4 discusses the designed NMPC energy management controllers, as well as controllers used for performance comparison from the literature. In Chapter 5, the evaluation of the EMC is presented by introducing MIL simulation and HIL testing. Finally, conclusions and future work are discussed in Chapter 6.

Chapter 2

Literature Review and Background

In this chapter a literature review on different energy management controllers previously proposed is presented and brief descriptions on some essential concepts used throughout the thesis is provided. Some of the most common architectures for battery-SC HESSs are discussed and the chosen architecture is verified. The concepts of model predictive control and stochastic model predictive control are also thoroughly described.

The chapter is organized as follows: in section 2.1 different typologies used throughout the literature are introduced and discussed. Section 2.2 presents a literature survey on various energy management controllers proposed so far. In section 2.3 the concept of MPC is described and some challenges and improvements are discussed. Section 2.4 covers the idea of stochastic MPC and a literature survey on different approaches proposed.

2.1 Battery-SC HESS Topology

Different topologies of battery-SC HESSs have been studied in the literature. The SC/battery topology has been used in this work, as it maintains adequate degrees of freedom for implementing various control strategies and maintaines a fair trade-off between the circuit complexity, performance and cost. A review of the most widely used topologies has been presented here [13].

As the passive parallel hybrid configuration shown in Fig. 2.1 uses no converter/inverter for combining the battery and SC together, it is known to be the simplest hybridyzing method available. The SC in this topology acts as a low-pass filter and the SC and battery are both in parallel with the DC bus, i.e. $V_{DC} = V_{SC} = V_B$ [14]. Although this topology is the most inexpensive, it cannot effectively use the stored energy of the SC.

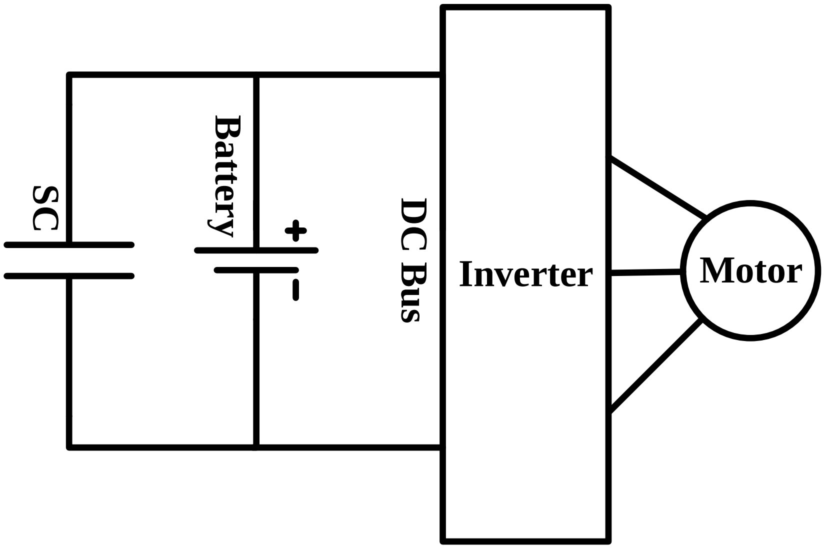

Figure 2.1: Basic passive parallel hybrid topology

By placing a bidirectional DC/DC converter for the interface of the SC with the battery/DC bus, as shown in Fig. 2.2, the SC/battery configuration is obtained. This allows for the voltage of SC to change in a wide range. However, it requires a larger converter in order to handle the power of the SC. The battery, on the other hand, is directly connected to the DC bus, causing the voltage of the DC bus to remain constant. The configuration is the most commonly used HESS in the literature [15, 13] and maintains adequate degrees of freedom for implementing various control strategies. Also, as only one converter is in use, a fair trade-off between the circuit complexity, performance and cost has been gained.

By swaping the position of the battery and SC in the topology above, the battery/SC configuration is obtained and shown in Fig. 2.3. The voltage of the battery can vary from the voltage of the SC in this configuration and the SC, connected directly to the DC bus, works as a low-pass filter. This however does not allow for the stored energy of the SC to be effectively used [14].

By adding another bidirectional DC/DC converter between the SC and the DC bus as shown in Fig. 2.4, a cascaded configuration is obtained. This allows for a wider working

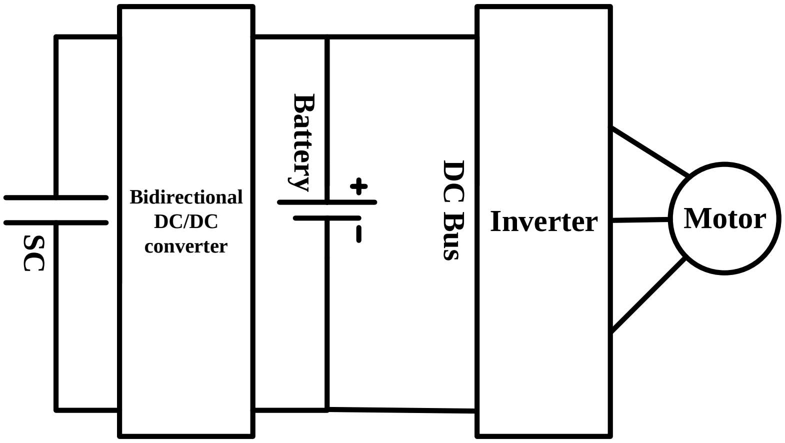

Figure 2.2: SC/battery topology

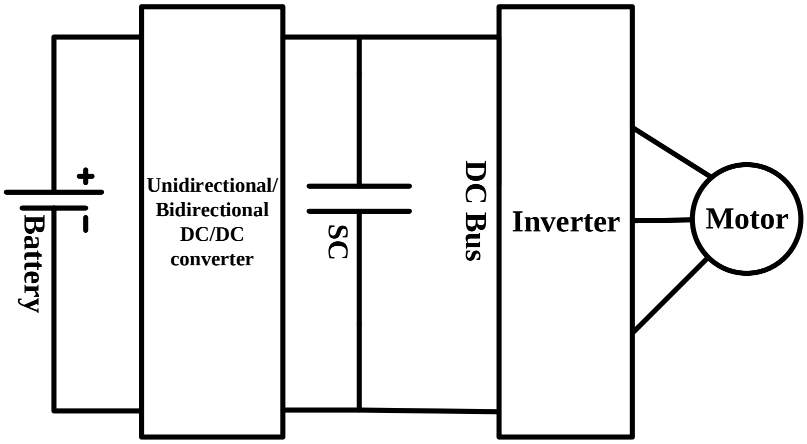

Figure 2.3: battery/SC topology

range for the SC than the SC/battery topology. However, this comes for the cost of an additional converter.

If the two converters are installed in parallel with each other, a multiple converter topology is obtained, Fig. 2.5 [16]. The voltage of both the battery and SC can be

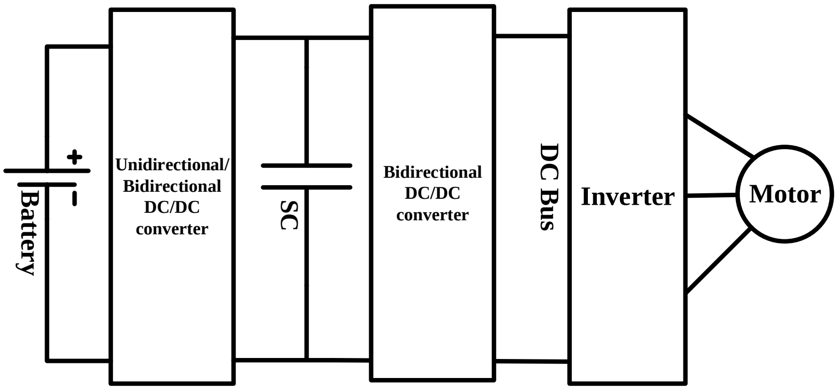

Figure 2.4: Cascaded topology

maintained to be different from the DC bus and the stored energy of the SC can be fully used. The disadvantage of this method is that two full-size converters are needed.

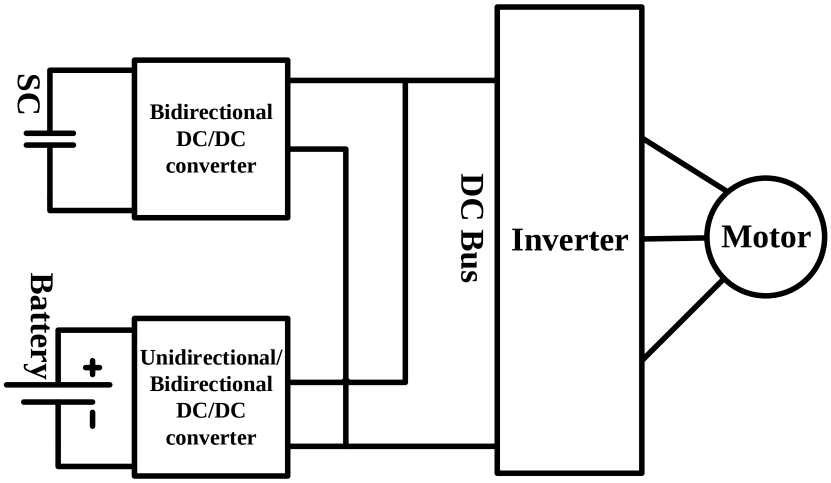

Figure 2.5: Multiple converter topology

The multiple input converter topology was introduced in order to reduce the cost of the two full-size convereters in the multiple converter system and is shown in Fig. 2.6 [16, 17].

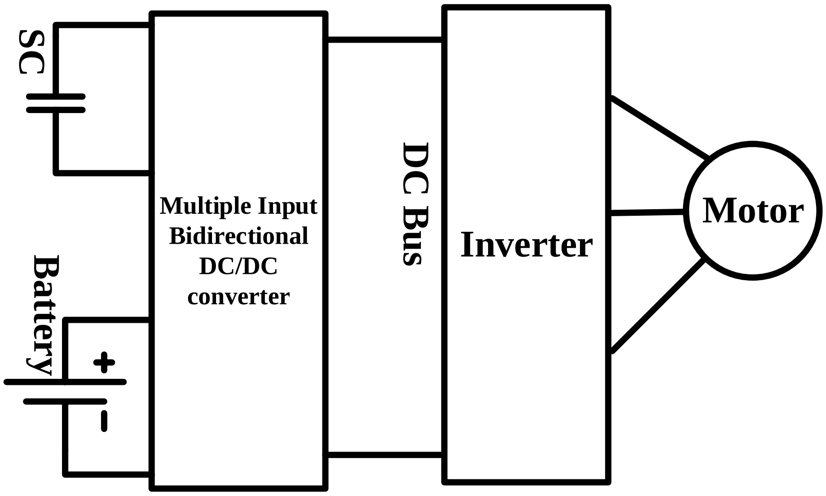

Figure 2.6: Multiple input converter topology

2.2 Energy Management Controllers

Although, hybridizing EVs with SCs seems to be a promising solution for improving system efficiency and battery health, the maximum benefit of these HESSs could only be exploited when a proper EMC is employed. The EMC defines the best power split at each time instant for maximum performance and efficiency.

Various energy management strategies for the battery-SC HESSs in EVs have been previously proposed; ranging from rule-based control (RBC) to optimal control. A number of RBC strategies have been set forth by [13, 18, 1]. RBC methods are known to be viable for real-time applications but do not ensure best performance. In [1], an RBC system has been applied that uses two adjustable parameters: The battery minimum power and the charging power from the battery to the SC. Ferreira et al. proposed a fuzzy logic controller for battery peak current reduction [19]. Offline control approaches using dynamic programming (DP) are also found in the literature [20, 21]. In [22], a DP approach has been used for component sizing and deriving a near optimal RBC. Rmoaus at al. solves the energy management problem of the HESS using stochastic dynamic programming (SDP) [21]. In [23], we have presented a potential investigation for the battery-SC HESS of a Toyota Rav4EV using DP and a very simple RBC.

Researchers in [24, 25, 26] have treated the battery-SC HESS as an optimization problem. In [25], a real-time optimization-based strategy has been experimentally validated using a dSPACE card and compared against an RBC system. The optimization-based approach is shown to have equivalent performance to that of the RBC in situations were no prior knowledge of the trip is available.

In [15], an artificial neural network has been designed as the EMC of a HESS and trained using various sets of data. Styler and Nourbakhsh also propose a data driven adaptive optimization approach for energy management of HESSs [27]. The optimality of these controllers depend on the set of data used for training the network.

Others have employed a model predictive control (MPC) strategy as the EMC of the battery-SC HESS. MPC has shown to be a promissing solution for a variety of applications in vehicle control [9, 8]. In [28], a linear model predictive control (LMPC) method has been introduced and verified via experiment, while an estimator is used for predicting the future power demand. In LMPCs, the optimization problem is linearized in order to speed-up the computational process. Next a comparative study between two RBC strategies, a fuzzy controller and a LMPC, has been conducted in [29] while no knowledge of the future power demand is assumed. The study shows that the performance of the LMPC falls behind, or at most equals the performance of the RBC approach introduced by [1], depending on the drive cycle. The performance difference between the aforementioned RBC and fuzzy controller is also shown to be minimal in the two driving cycles used by Song et al. Therefore, in this work, the RBC in [1] has been used as a fair benchmark for the performance of non-optimization based methods. In [30], a DP based approach is used to find the optimal control within a model predictive control method while the future power demand is assumed to be partially known using GPS and ITS. The paper, however, does not provide any guidelines on how this prediction is performed and how it affects the system's performance. The DP based approach is also far from real-time implementable. Laldin et al. uses a stochastic model predictive control approach to predict the future load torque [12]. The authors have presented a statistical Markov process using a finite number of states, defined from the speed/acceleration of the previously known driving cycle data. However, the proposed method shows to diverge from optimality when the actual driving cycle is different from the previous known driving cycles. The accuracy of the applied controller also drops when exposed to sharp changes in vehicle speed. The derived nonlinear model predictive control (NMPC) system is afterwards solved using Matlab's $fmincon$ solver. Experimental results are presented using two Linux PCs that are far more powerful than the actual ECU available on the vehicle. This paper therefore doesn't demonstrate real-time capability of the applied NMPC system. Meyer et al. also uses an NMPC approach using Matlab's $fmincon$ as a solver. However, the paper fails to show real-time capability of its proposed NMPC method [31].

In this work we have presented a real-time NMPC method as the EMC of a SC/battery

HESS in EVs. Comparative results with state of the art controllers are also demonstrated.

2.3 Model Predictive Control

MPC is an advanced optimal control strategy that takes knowledge of the future outputs over a prediction horizon of length N_p into account to optimize the behavior of a system with respect to a cost function. The control actions are calculated repeatedly at each time step, knowing the current state of the system and using a control-oriented model developed for fast output prediction. To calculate the optimal controls, an open-loop finite horizon optimization problem is solved at each time step to investigate the optimal controls at time intervals of a control horizon of length N_c. This has been demonstrated in Fig. 2.7. As results of the optimization problem need to be calculated before the next receding sampling time arrives, the computational time of the solver with respect to the optimization problem is of crucial importance. Therefore, there exists a trade-off between optimization complexity and computational time. The nonlinearity of the control-oriented model and cost function, as well as the length of the prediction and control horizon all correspond to the complexity of the optimization problem and hence the computational time. The use of a nonlinear or linear control-oriented model and optimization solver defines whether the controller is called nonlinear model predictive control (NMPC) or linear model predictive control (LMPC). The LMPC requires lineariztion of the control-oriented model and constraints in advance, in addition to a quadratic cost function, to allow compatibility with linear solvers. The NMPC, on the other hand, can work with almost any form of nonlinear control-oriented model and cost function. As in LMPC, finding sub-optimal control actions is equivalent to solving a convex Quadratic Programming (QP) problem at each sampling time; the real-time implementation of such controllers has shown to be feasible in the liturature [32]. The real-time implementation of NMPC solvers however are still under research. The optimizer calculated results, for the first control is afterwards applied to the system and the rest of the control sequences are discarded. The new state obtained by applying the calculated control to the system (or high-fidelity model) is used to update the initial conditions of the optimization problem.

The use of a high-fidelity model for off-line testing, prior to manufacturing the prototype vehicle is essential to the control development process. The high-fidelity model is a relatively complex model of the vehicle that predicts the vehicle behavior to the extent necessary for controller development.

The control-oriented model is a more simple, computationally efficient model, that captures sufficient accuracy in the system behavior for fast online control calculations.

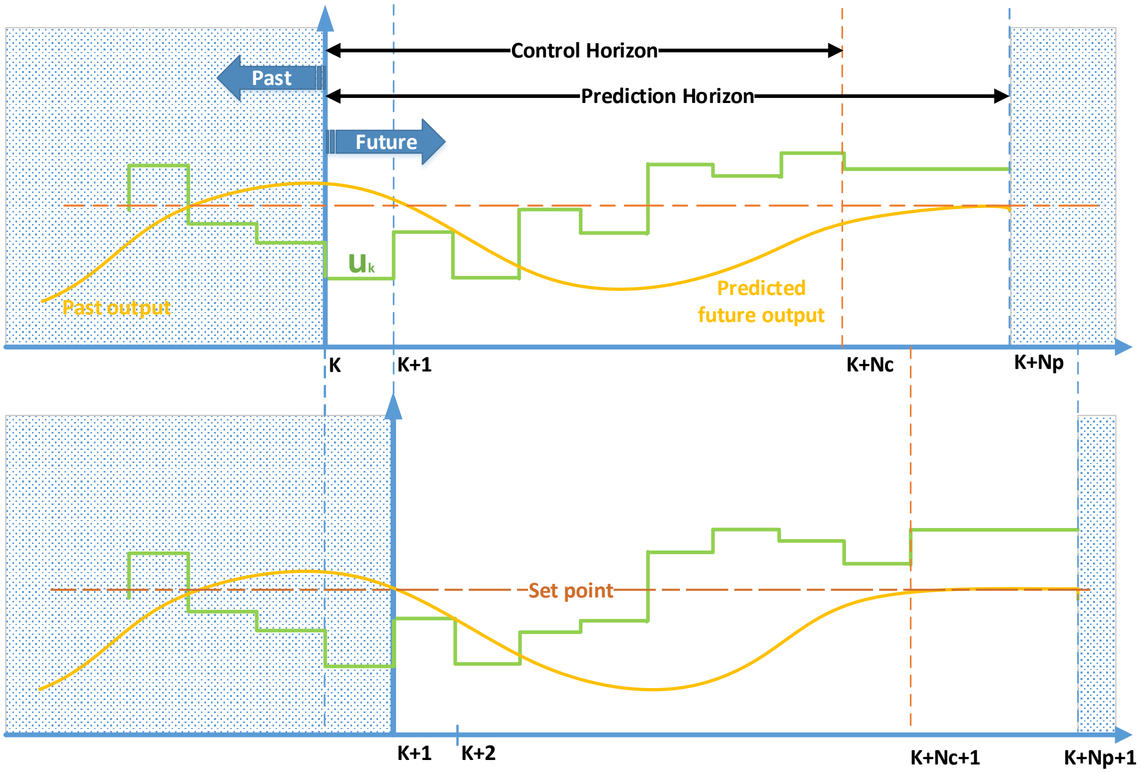

Figure 2.7: Model Predictive Control principle illustration

Obviously, the use of a linear control-oriented model (LMPC) allows for more time efficient behavior estimation, but captures less accuracy, in comparison to a nonlinear control-oriented model (NMPC).

Depending on the nature of the problem, the optimization problem might include a number of time variant parameters, that need to be independently estimated over the prediction horizon. In our problem of interest, the power demand is a time variant parameter. The value of this parameter is essential to solving the optimization problem but is unknown over the prediction horizon. As the performance of the MPC method is highly dependent on the drivers future power demand estimation [33], the future power demand needs to be carefully estimated. This is done by various ways in the literature. Some have assumed full knowledge of the future trip [30], while others have used some form of power estimator to predict the upcoming power demand [28]. In [12], a stochastic Markov process has been used to predict future torque and load currents.

The use of Explicit MPC (EMPC) has been proposed to conquer the limitation in computational speed. In this method, the nonlinear optimization is solved offline and the results are clustered to form a look-up table that could be used online. The method, however, faces storage limitation for the look-up table obtained offline [34, 35].

The improvement in computational hardware like electronic control unites (ECUs) in the recent years shows promising potential for NMPC implementation in faster applications as in automotive control [36, 37, 9]. Many fast nonlinear solvers have been proposed to exploit this potential e.g. GMRES-optimization methods [38], Shooting-based newton methods [39] and interior point methods [40]. In this work, a shooting-based newton method has been employed as the solver of NMPC.

2.4 Stochastic Model Predictive Control

Although MPC brings a certain degree of robustness to the system, due to its receding horizon implementation, it fails to provide a systematic way for dealing with system uncertainties. The development of robust MPC methods (RMPC) in the past two decades, meant to overcome these limitations. Early work on RMPC was based on a min-max approach where the optimization problem was solved with respect to the wort-case realization, rendering the calculated control outputs to be too conservative [41, 42]. The tube-based MPC approach was introduced to overcome the short comings in the Min-Max approach [42, 43]. While uncertainties are often considered to be of probabilistic nature, the tube-based MPC doesn't discuss an explicit way to consider stochastic knowledge of uncertainties in control design. Hence, stochastic MPC (S-MPC) emerged to take probabilistic knowledge of the uncertainties into account for solving a stochastic optimal control problem. The method is rooted in stochastic programming and chance-constrained optimization and has found applications in various areas including automotive control in the recent years [44, 45]. While the majority of application are reported for linear systems, Stochastic Nonlinear MPC has received little attention; with mainly applications in the area of process control [46, 47].

As a natural approach to tackle a stochastic optimal control problem would be to employ stochastic programming techniques [48, 49], ideas from multistage optimization [50] and sample-based approaches [51] have been adopted for obtaining approximate stochastic solutions. In these methods, a time-varying stochastic tree in formed based on the updated statistical information from the past and used to solve an optimal control problem in one shot, where a finite set of random realization of uncertainties are considered [50]. The large sample size required in sample-based approaches, however, prohibit the approach from having many practical applications. The *scenario approach* was therefore a significant development in the area of stochastic optimization [47]. The method employs appropriate number of samples that enable approximate feasible solution for the original problem [52]. As the method imposes a certain level of robustness to the system, some have calssified it under the robust MPC category in the literature [53].

In this work, a scenario-based multistage stochastic programming method has been adopted for solving the stochastic optimal control problem of interest. As the main uncertainty imposed to the system is single sourced (due to power demand), a two-stage optimization problem has been adopted as a subset of the multistage method. The hard constraints on inputs and states are transformed into soft constraints by using slack variables that define the possible extent of constraint violation. Although theoretical bounds on the number of scenario/samples to probabilistically ensure feasibility and constraint satisfaction are available in the literature [53], the obtained number might end up to be too conservative. Therefore in this work, the number of scenarios adopted have been chosen accordingly to enable possible real-time implementation of the SNMPC controller. Scenario-reduction methods are also available in order to significantly reduce the number of scenarios neccesary to garantee an admissble level of constraint satisfaction as well as managable comutational time for practical application [54]. Hence recent work on scenario-based MPC has mainly focused on reducing computational complexity of these methods [47, 55, 56, 57] and will be investigated in the future work.

2.5 Chapter Summary

In this chapter, the main architectures for battery-SC hybrid energy storage systems for EVs were introduced and a literature review on various energy management controllers for such powertrains was presented. The linear and nonlinear MPC methods have been briefly explained and the practical implementation issues have been discussed. Finally, the stochastic MPC method was introduced as a method for handling uncertainties in the system. A number of extensions to this method in the literature where also explored.

Chapter 3

System Modeling

In this chapter the powertrain model of the Toyota Rav4EV is presented. A complete block diagram of the controlled system, is shown in Fig. 3.1. It consists of the driver model, the high-fidelity model of the system, the power demand estimator and the controller. The high-fidelity model substitutes the actual vehicle within the control loop, to examine the actual response of the system. The controller employs a control-oriented model for approximate, fast prediction of the vehicle's response along with some form of the future power demand estimation, in order to command proper action. The driver model consists of a feed forward and feedback controller and ensures that the vehicle follows the desired reference speed.

This chapter is organized as follows: in section 3.1 the driver model is presented. Next, the high-fidelity model of the system has been introduced in section 3.2 by separately discussing the chassis model and the SC/battery HESS model. The control-oriented model of the SC/battery HESS used within the controller is afterwards presented in section 3.3 and validated in section 3.4. Finally, the power demand estimations used throughout this work are discussed in section 3.5.

3.1 Driver Model

The driver model block assures that the vehicle follows the desired reference speed. It consists of a a feedforward and feedback controller. As the feedback controller relies on feedback from the high-fidelity model, this controller is slow but accurate. Therefore, the feedforward controller is added to the system to assist the feedback controller and increase

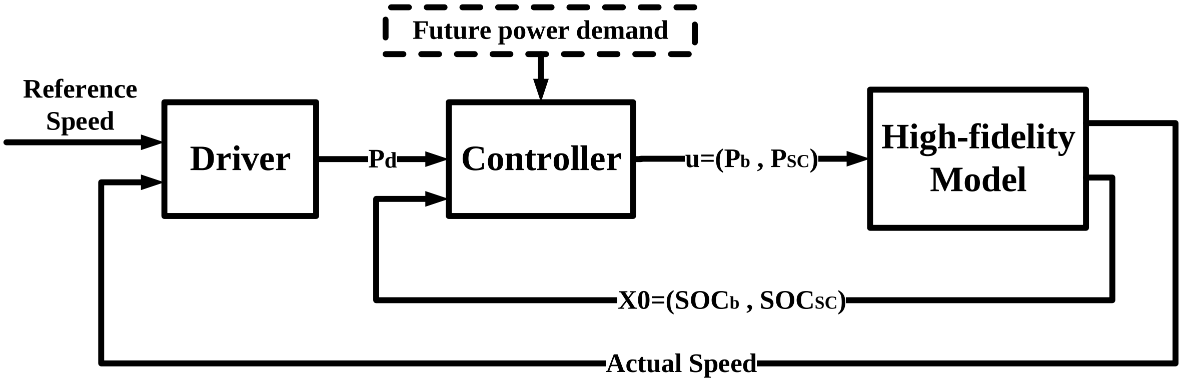

Figure 3.1: Block-diagram of the controlled System

the driver speed. While the feedback controller adjusts the desired power using a PID controller, the feedforward controller estimates the desired power based on longitudinal dynamics of the vehicle (Eq. 3.1-3.3). In order to do so, the resistance force F_r at a certain speed $v(t)$ and road slop $\theta(t)$, are calculated using the longitudinal dynamics model of the vehicle:

$$F_r(t) = C_{rr}mg + mgsin(\theta(t)) + m\dot{v}(t) + 1/2\rho_a C_d A_f v(t)^2, \tag{3.1}$$

where ρ_a, C_d, A_f, C_{rr} and m are the air density, drag coefficient, frontal area, rolling resistance coefficient and vehicle mass, respectively.

As the force at the wheels is delivered from/to the motor through the vehicle's gearbox, knowing the resistance force and vehicle speed, the required motor torque $\tau_m(t)$, can be computed as:

$$\tau_m(t) = (F_r(t) + m\dot{v}(t))(r_w/G_R), \tag{3.2}$$

where r_w and G_R are the radius of the wheels and gear ratio respectively. Finally, the power demand $P_d(t)$, from/to the motor can be determined as:

$$P_d(t) = \begin{cases} \tau_m(t)\omega_m(t)/\eta_a & \tau_m(t) \geq 0 \ , \\ \tau_m(t)\omega_m(t)\eta_d & \tau_m(t) < 0 \ , \end{cases} \tag{3.3}$$

where $\omega_m(t)$ is the motor rotational speed that equals $\omega_m(t) = v(t)G_R/r_w$. Also, $\eta_a = 0.85$ and $\eta_d = 0.35$ are the motor/inverter efficiency during acceleration and deceleration (regenerative breaking) respectively, which are assumed to be equal to the values indicated above [5].

3.2 High-fidelity Model

The high-fidelity model is used for evaluation of the controller's performance in MIL simulations and HIL testing. It represents the actual vehicle within the control loop, and examines the response of the system to a certain control signal. It can also be used for model validation of the control-oriented model. The high-fidelity model consists of a chassis model developed in MapleSim and a model of the SC/battery HESS.

3.2.1 Chassis Model

A four wheel chassis model of the front wheel drive Rav4EV has been developed using the MapleSim software to incorporate the longitudinal dynamics of the vehicle for the high-fidelity model. The symbolic computation power of MapleSim makes the model very suitable for real-time HIL testing. The model consists of a lumped car mass that is connected to the tires through four vertical suspensions. The tires have been modeled using a magic-formula tire model available in MapleSim. The applied motor torque is transmitted to the tires through a drivetrain model consisting of rotary inertia of the induction motor, reduction gear and flexible driveshaft. The model is afterwards transferred and used in the Simulink environment. The model simulates the vehicle actual speed, considering the motor torque as the input.

The parameters for the longitudinal dynamics model have been derived through experiments on the Toyota Rav4EV. Experimental tests have been conducted on the Toyota Motor Manufacturing Canada (TMMC) test tracks and are present in Table 3.1. The data have been recorded by integrating the measurements from a system of sensors using Vehicle Measurement System (VMS), GPS and the vehicle Controller Area Network (CAN). A MATLAB/Simulink based nonlinear least square parameter estimator has been used to identify the parameters of the Rav4EV model. More details on the chassis model, along with parameter identification of the Rav4EV are available in a previous publication by the authors' reseach group [58].

3.2.2 SC/Battery HESS Model

The topology of HESS used in this work is presented in Fig. 2.3. A bidirectional DC/DC converter is used for the interface of the SC with the battery/DC bus, allowing for the voltage of SC to change in a wide range. The battery, on the other hand, is directly connected to the DC bus, causing the voltage of the DC bus to remain constant. The proposed

Table 3.1: Parameters for longitudinal dynamics model of Rav4EV

Parameter	Symbol	Value	Unit
Vehicle mass	m	1970	kg
Drag Coefficient	C_d	0.3	-
Frontal Area	A_f	2.464	m^2
Rolling resistance	C_{rr}	0.015	-
Wheel radius	r_w	0.355	m
Gear Ratio	G_R	0.9.73	-

topology, maintains adequate degrees of freedom for implementing various control strategies. Also, as only one converter is in use, a fair trade-off between the circuit complexity, performance and cost has been gained. This topology is common in much of the literature on battery-SC HESSs [31]. More details on different topologyies of battery-SC HESS can be found in [13]. It has been assumed that the power net, flowing in and out of the DC bus at each time step, are equal to zero, i.e.

$$P_b(t) + \eta_c P_{SC}(t) = P_d(t), \tag{3.4}$$

where $P_b(t)$, $P_{SC}(t)$ and η_c are the battery power, SC power and converter efficiency, respectively. The DC/DC converter is modeled with an average efficiency, $\eta_c = 0.92$, which also includes losses related to the electric bus [59]. Since the structure of the DC/DC converter does not play an essential role in energy management, a detailed model of the converter has not been taken into account. As the efficiency of the converter connected to the SC drops significantly when the voltage of the SC is below $0.5V_{SCmax}$, the state of charge (SOC) of the SC, defined as $SOC_{SC} = \frac{V_{SC}}{V_{SCmax}}$, has been limited to vary between 0.5 and 1 [24]. The SOC of the battery is also limited to vary between 0.2 and 0.9 to avoid high depth of discharge that degrades the battery.

Also, the motor and motor inverter play the role of a power demanding unit. Therefore, the model of their structure is not essential to the subject of this research [59].

For the high-fidelity model of the battery, a Lithium-Ion battery developed in the Autonomie software has been used. As this study does not discuss supercapacitor sizing, by assumption the SC used in this work consists of 25 Maxwell BCAP1200 SCs in series causing the total capacity of 30 Wh [60]. The controls need to be re-tuned, if other sizes of SC are intended to be used. For the SC model a Matlab's generic supercapacitor block has been applied [61]. A third order SC model has been applied using this block. The main properties of the SC used in this study are listed in Table 3.3.

Table 3.2: Parameters of Maxwell BCAP1200 cells

Rated Voltage	Rated Capacitance	Series Resistance	Mass
$2.7V$	$1200F$	$0.58m\Omega$	$260g$

Table 3.3: Parameters of Rav4EV Lithium-ion Battery Pack

Rated Voltage	Usable Energy	Resistance	Max Power Output
$380V$	$41.8kW/h$	0.8Ω	$129kW$

3.3 Control-Oriented Model

In practice, the high-fidelity model cannot be used for model-based control purposes, due to the complexity of the high-fidelity model and the necessity for fast, real-time implementation in controllers. For such purposes a control-oriented model is developed, which is more computationally efficient, yet sufficiently accurate. Employing such control-oriented models, allows for fast, accurate controllers that are suitable for real-time implementation. The remainder of this section explores mathematical modeling for implementation of simple, yet adequately accurate, control-oriented models of the battery pack and SC. The battery and SC can be modeled by a voltage source followed by an internal resistance as shown in Fig. 3.2 and 3.3. The open circuit voltage and the internal resistance of the battery alongside with the capacitance, maximum rated voltage and the internal resistance of the SC are assumed to be constant. The state of charge is defined as the ratio between the stored charge and the maximum charge capacity of a battery or SC. The dynamics for the state of charge of the battery and SC can be determined as:

$$\begin{aligned} S\dot{O}C_b &= -\frac{V_{oc} - \sqrt{V_{oc}^2 - 4R_bP_b}}{2R_bC_b}, \\ S\dot{O}C_{SC} &= -\frac{SOC_{SC}V_{max} - \sqrt{(SOC_{SC}V_{max})^2 - 4R_{SC}P_{SC}}}{2R_{SC}C_{SC}V_{max}}, \end{aligned} \tag{3.5}$$

where SOC_b and SOC_{SC} are the state of charge of the battery and SC, respectively. Also, V_{oc}, R_b, C_b and P_b are the open circuit voltage, internal resistance, capacity and charge/discharge power of the battery while, V_{max}, R_{SC}, C_{SC} and P_{SC} are the maximum rated voltage, internal resistance, capacity and power of the SC, respectively. The positive power indicates discharge mode and the negative power indicates charge mode within the battery and SC. The battery current could be determined by:

$$S\dot{O}C_b = -\frac{I_b}{C_b}, \tag{3.6}$$

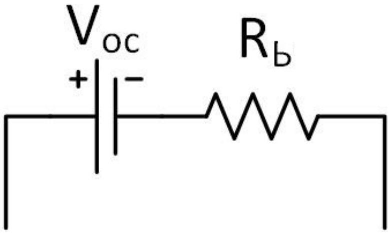

Figure 3.2: Battery control-oriented model

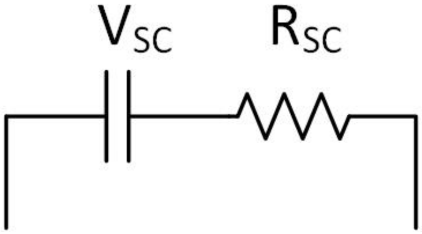

Figure 3.3: SC control-oriented model

3.4 Control-oriented Model Validation

As the control-oriented model has been employed in the controller as a fast representative of the high-fidelity model, consideration has been paid to make sure that the response of the control-oriented model remains close to that of the high-fidelity model throughout the trip.

MIL simulation results have been illustrated in Fig. 3.4 for the rule based controller introduced in section 4.1, by replacing the high-fidelity SC/battery HESS model in Fig. 3.1 with the control-oriented model presented in section 3.3. Comparative results have been demonstrated, showing the maximum error for the SOC of the battery, the SOC of the SC and battery current squared sum to be 0.02%, 2.88% and 2.60% respectively over a UDDS driving cycle. Therefore, the control-oriented model shows adequate accuracy for performance estimation in predictive controllers.

3.5 Future Power Demand Estimator

The performance of predictive controllers are highly dependent on the power demand estimation strategy. Even though full knowledge of the future trip is never available, due to uncertainties in road condition and driver behavior, simulations show that partial knowledge of the future trip can also be beneficial. Partial knowledge can be obtained by either using on-board technologies or statistical information from the past trips. The remainder of this section is dedicated to discussing the power demand estimation strategies used for performance evaluation of our controllers.

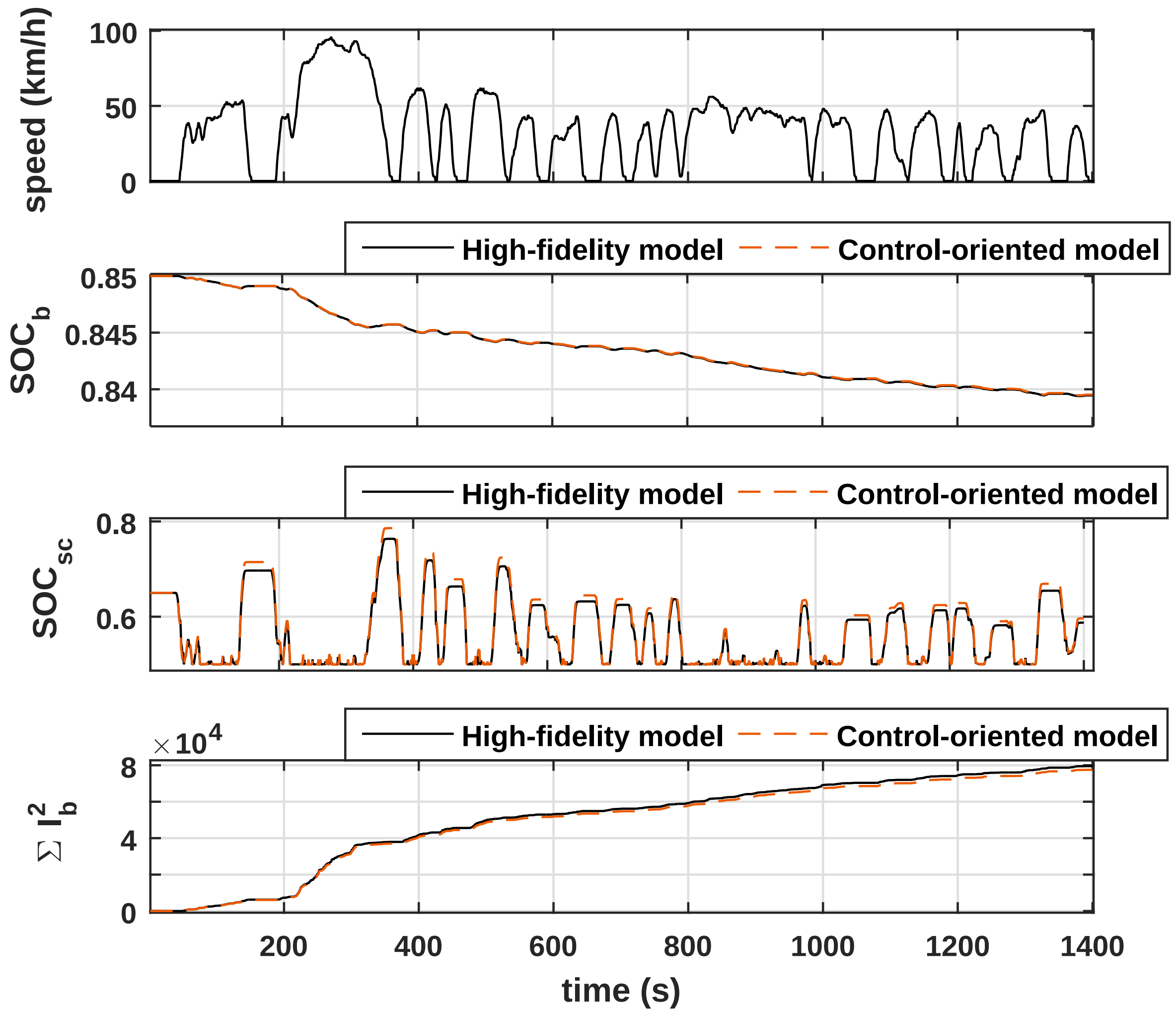

Figure 3.4: Simulation results for validating the control-oriented model

3.5.1 Frozen Time Model

Also known as frozen-time MPC (FT-MPC) [44], is a case where no priori knowledge of the future trip is used in the decision making of the predictive controller. In FT-MPC it is assumed that the measured disturbance, i.e. the power demand, is constant over the prediction horizon. The philosophy for this assumption, lies in the probability distribution of the future power demand. It is observed from real-world data analysis that the future power demand is most likely to be close to the current power demand. As further discussed in section 3.5.3, Fig. 3.6 demonstrates the transition probabilities of power demand which have been derived from a combination of standard driving cycles; UDDS, NEDC, HWFET and US06. Transition probabilities are estimated by analyzing available data to evaluate the probability of the next power demand based on the current power demand. Diagonal dominance can be observed in low power demands in the figure, which indicates that it is most likely for the next power demand to stay close to where it currently stands. These transition probabilities are further used in stochastic modeling.

3.5.2 Trip Planning Model

The advances in technologies such as the Global Positioning System (GPS), Intelligent Transportation System (ITS), internet maps and real-time traffic data, coupled with advances in information transfer to and from vehicles, have made prior knowledge of route and traffic data more accessible then ever. Having access to these technologies allows us to divide the future route into segments in which each division has constant key parameters such as traffic speed, maximum allowable speed and road grade. Obstacles such as traffic lights, stop signs and road intersections are what form the border of each segment [10]. The velocity for each segment is defined as the minimum of the maximum legal velocity of the road v_{max}, the maximum safe cornering velocity v_{curv} and the traffic velocity obtained by the vehicles ahead $v_{traffic}$ [62].

$$v_{seg} = min(v_{max}, v_{curv}, v_{traffic}). \tag{3.7}$$

Using Eq. 3.7, the velocity for each segment could be calculated in advance. Fig. 3.5 shows examples of segmented speed trajectories for standard driving cycles. Each segment has been modeled with a constant cruise speed, indicated as the average speed over each segment, shown as a dashed red line. The acceleration and deceleration mode at the beginning and the end of each segment has been neglected for simplicity. Based on the data collected from GPS, traffic data and etc, the proposed velocity segments are formed and the future power demand is estimated using equations obtained from section 3.1. Experimental research on accuracy and delay modeling of the proposed trip planning MPC (TP-MPC) is still in hand.

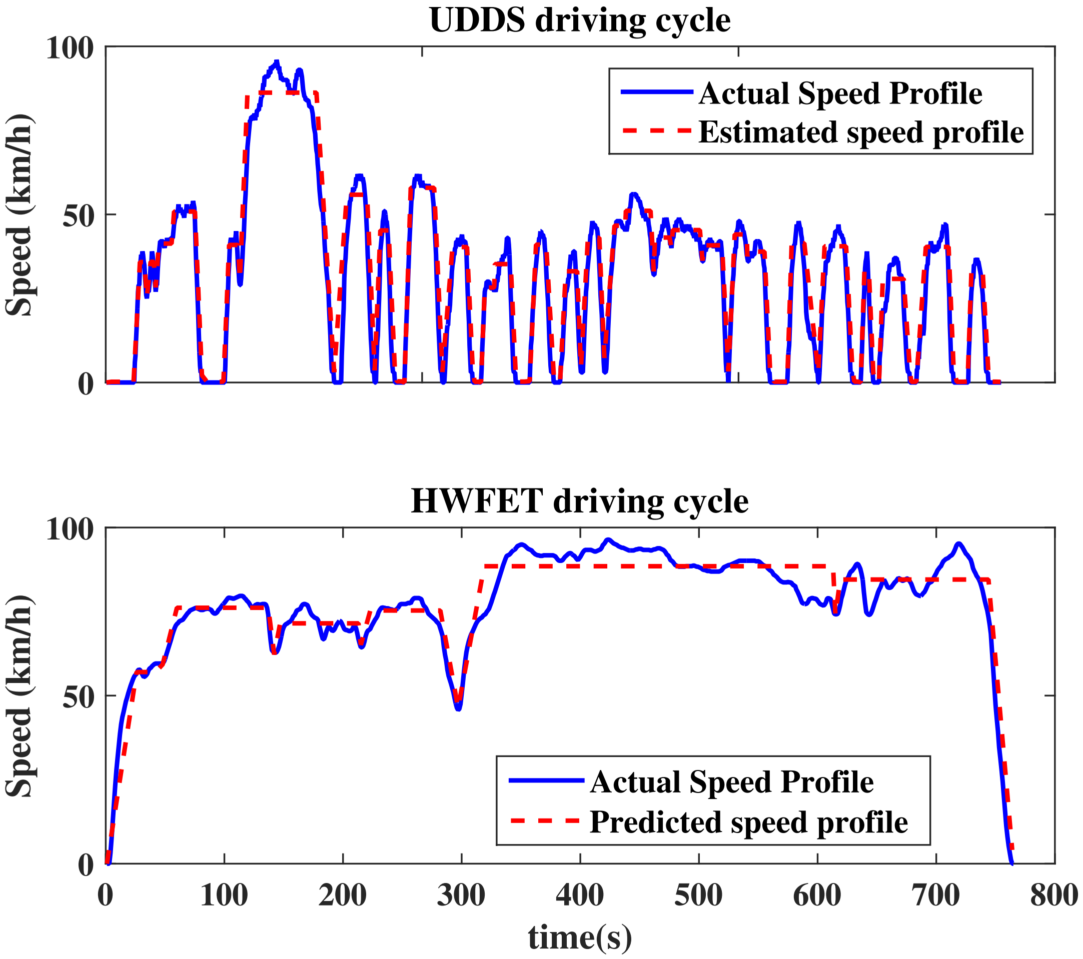

Figure 3.5: Speed trajectories corresponding to different route segments

3.5.3 Stochastic Model

In this strategy, the power demand is modeled as a discrete-time, continuous state, time invariant stochastic process. The random gereration of the power demand has been modeled using a Markov chain with power demand as the Markov state [44, 63]. This model defines the probability of the next power demand to fall in a certain range, considering the range of the power demand at the current state. In order to achieve this model, the power demand is assumed to fall in a finite number of ranges, represented with the midpoint values of $P_d \in \{p_1, p_2, ..., p_m\}$. It should be noted that, throughout this work whenever referring to p_i, reference is made to a range not the midpoint value. The number m has been chosen to maintain a fair trade-off between accuracy and complexity of the stochastic process and has been set to be 20 in this work based on trial and error tuning. The ranges have been formed by dividing the allowable P_d into equal divisions. The Markov chain is defined by a transition probability matrix $TPM \in \mathbb{R}^{m \times m}$, such that;

$$t_{ij} = Pr\{P_d(k+1) = p_i | P_d(k) = p_j\} \quad i, j \in \{1, ..., m\}, \tag{3.8}$$

where $P_d(k)$ is the state of the Markov chain at time k and t_{ij} is an element of the TPM matrix located in the ith row and jth column. To estimate these transition probabilities, training data from standard driving cycles or past driving records are needed. As the main benefit of hybridyzing EVs with SCs is noted to be in urban driving [6], in this work, estimations of the transition probabilities t_{ij} and number of finite states m have been done using a number of urban driving cycles, consisting of standard driving cycles (FTP75 and US06) and several real-world trips as training data. Power demand P_d is calculated from speed profiles using the vehicle model presented in section 3.2 while the driver is tuned to follow the desired reference speed. The transition probabilities are calculated as $t_{ij} = \frac{n_{ij}}{n_i}$, where n_{ij} is the number of times a transition from p_i to p_j has occurred and $n_i = \sum_{j=1}^{m} n_{ij}$ is the total number of times p_i has occurred. The method employed for deriving transition probabilities is relatively consistent throughout the literature [63] . When new data is collected through the trip, the TPM matrix could be easily updated. The estimated transition probabilities are shown in Fig. 3.6.

3.5.3.1 Scenario Generation

As the power demand needs to be estimated over a prediction horizon with the length of $N_p \in \mathbb{N}$, the non-zero transition probabilities are used to form a scenario tree at each time step with respect to the current power demand. The scenario tree shows the possible transition from the current state to the next N_p states while each node of the tree represents a predicted state and is used in the optimization process. An example of a scenario tree for $N_p = 5$ and a starting state of p_i (representing the current power demand) has been demonstrated in Fig. 3.7. The example shows twenty possible scenarios, when starting from p_i. The probability of each scenario can be determined from the product of the branch transition probabilities. Although Fig. 3.7 shows all the possible scenarios when starting from p_i, including all the twenty scenarios in our decision making, would lead to a relevantly large optimization problem which would be challenging to solve online. Therefore, a small number of these scenarios are taken into account at each time step. In order to select

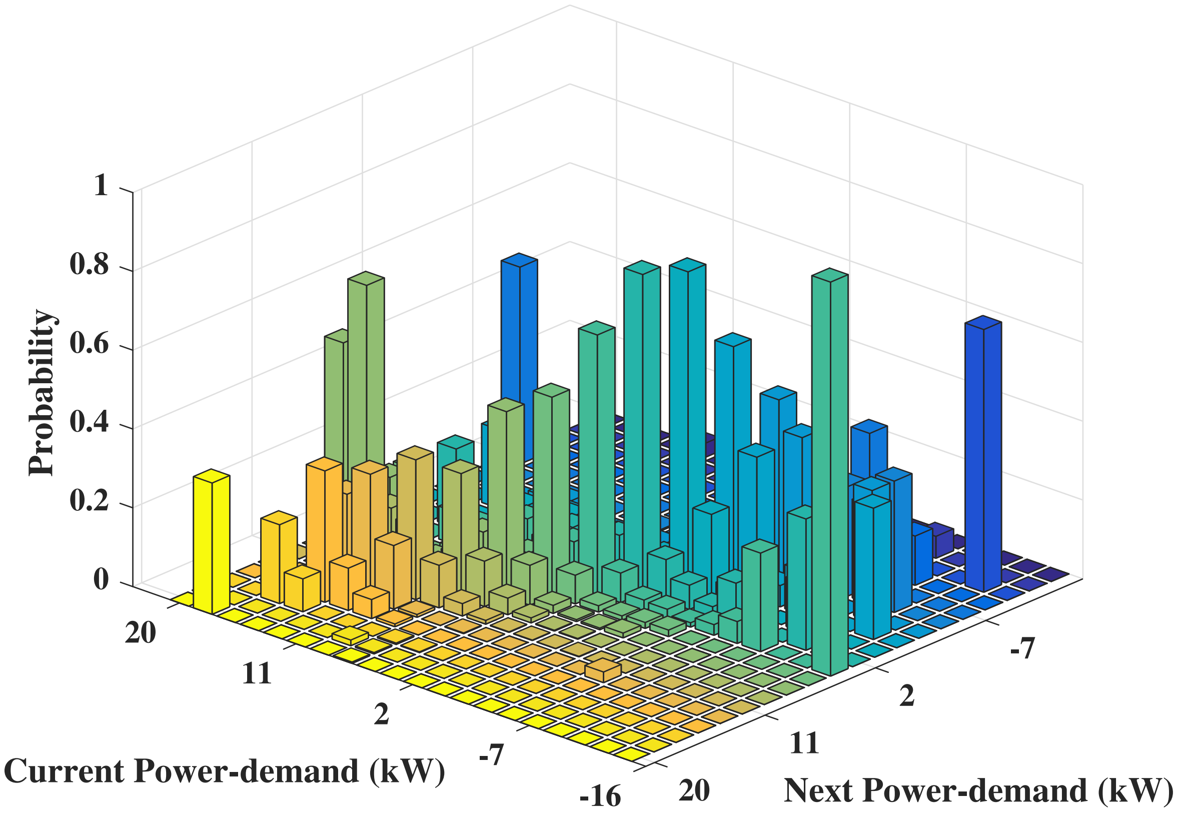

Figure 3.6: Transition Probabilities of Power Demand

the best scenarios to work with, the maximum likelihood approach is often used in the literature [64, 44]. However, throughout this work a different approach has been employed, where the scenarios are selected based on a weighted random choice; using the transition probabilities as weights. The advantage of this approach over the maximum likelihood approach is that although the scenarios with higher probabilities are more likely to be selected, there is a chance for the low probability scenarios to get involved in our decision

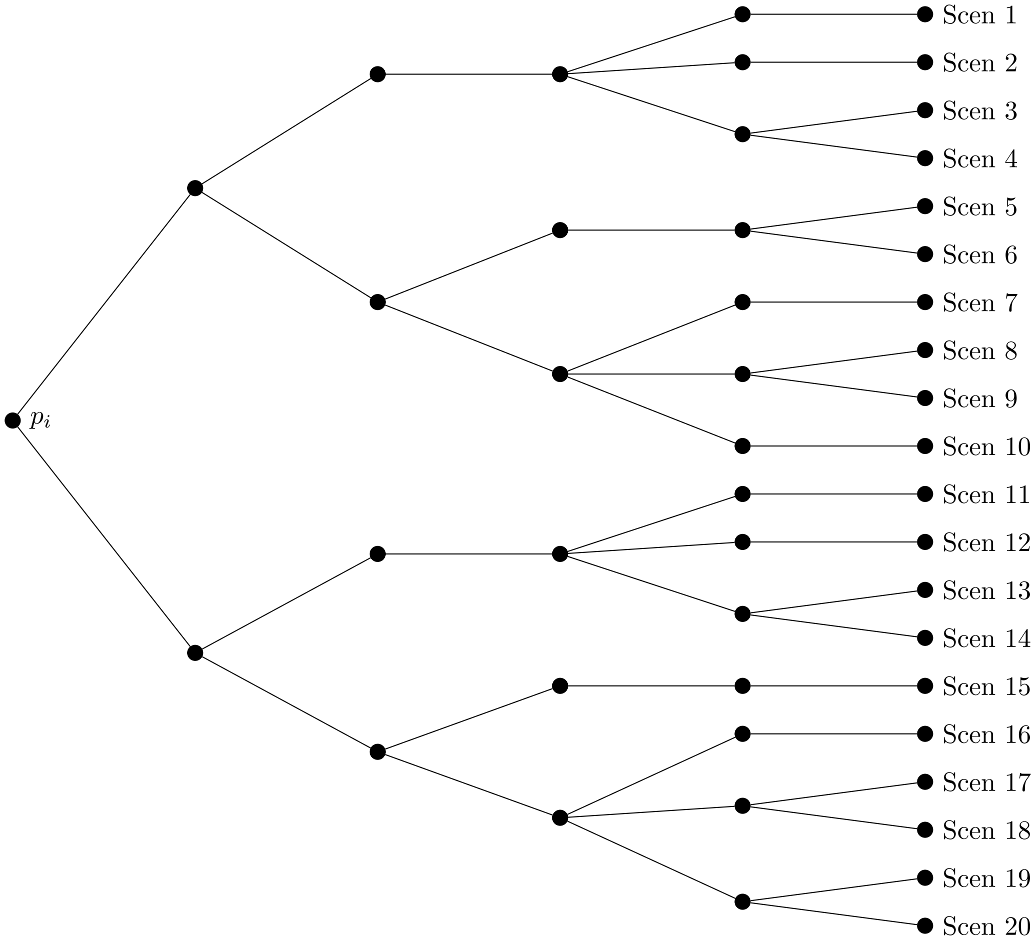

Figure 3.7: Scenario tree for $N_p = 5$

making. Therefore, although only a number of scenarios are involved in our optimization problem at each time step, unlike the maximum likelihood approach, no information from the TPM matrix is thrown away. Starting from the root node of the tree, the next node is randomly chosen using the transition probabilities as weights. This is done until all the power demand predictions over the prediction horizon of N_p time steps are generated. The procedure is repeated for as many scenarios we intend to involve in the optimization problem (N_S). As each node represents a range of power demand, the power demands have a continuous form and are chosen randomly within that range.

3.5.4 Prescient Model

Although full-knowledge of the future trip is never available a priori, simulation results have been presented as a reference for evaluating the estimators introduced above. The results achieved for full-knowledge of the future trip can depict the maximum potential benefit of the proposed NMPC strategy. Predictive control with full-knowledge of the future power demand is occasionally called prescient MPC (P-MPC) in the literature [44].

3.6 Chapter Summary

In this chapter, the model of various parts of the system are discussed. The driver is modeled using a feedforward and feedback PID controller and follows the desired reference speed. The control-oriented model consists of a battery and SC and is introduced as a more simple format of the system model that follows approximately the same behavior of the high-fidelity model. The high fidelity model is a more complex model of the system and is used for controller evaluation. The power demand estimation models are also discussed in this chapter. Power demand estimations are used by predictive controllers discussed in the later chapters. Four models have been used for predicting future power demand: frozen time model, trip planning model, stochastic model and prescient model. The models have been extensively discussed and are used in the following chapters.

Chapter 4

Energy Management Controller Design

The role of the EMC is to optimally split the power demanded by the motor between the battery and SC at each time step. Doing so, the battery remains as the main energy reservoir of the vehicle thorough the trip, while the SC takes over during high power peaks and power fluctuations that effect the battery health.

In this chapter, the various EMCs employed for the proposed system are introduced and discussed. The performance of the examined controllers are compared in the next sections.

The chapter is organized as follows: in section 4.1 the rule based controller used for comparison purposes is introduced. Section 4.2 covers the nonlinear model predictive controller designed in details; discussing the optimal control formulation in deterministic and stochastic form, the discretization methods used and the optimization solver developed. The linear form of the model predictive controller is presented in section 4.3 and finally, the dynamic programming method used for maximum potential investigation of the system is discussed in section 4.4.

4.1 Rule Based Controller

A rule based controller (RBC) has been implemented for the purpose of performance comparison. The flow-chart of the RBC applied in this work, is shown in Fig. 4.1. In the applied algorithm, during acceleration, up to a certain threshold, P_{min}, the power-demand, P_d, is provided by the SC and the access power is delivered by the battery. The battery also provides power charge to the SC, P_{ch}, whenever the SOC of the SC falls below the allowable range of 0.5. During regenerative breaking, the battery only receives regenerative power after the SC is fully charged [1, 29]. The algorithm requires tuning of the parameters P_{min} and P_{ch} for maximum benefit. The tuning values over a combination of driving cycles have been chosen to be 2.7 kW and 1 kW for P_{min} and P_{ch}, respectively.

The applied RBC is shown in [1] to be among the best non-predictive EMC of battery-SC HESSs in EVs and therefore has been considered as a performance benchmark for non-predictive controllers for SC/battery systems.

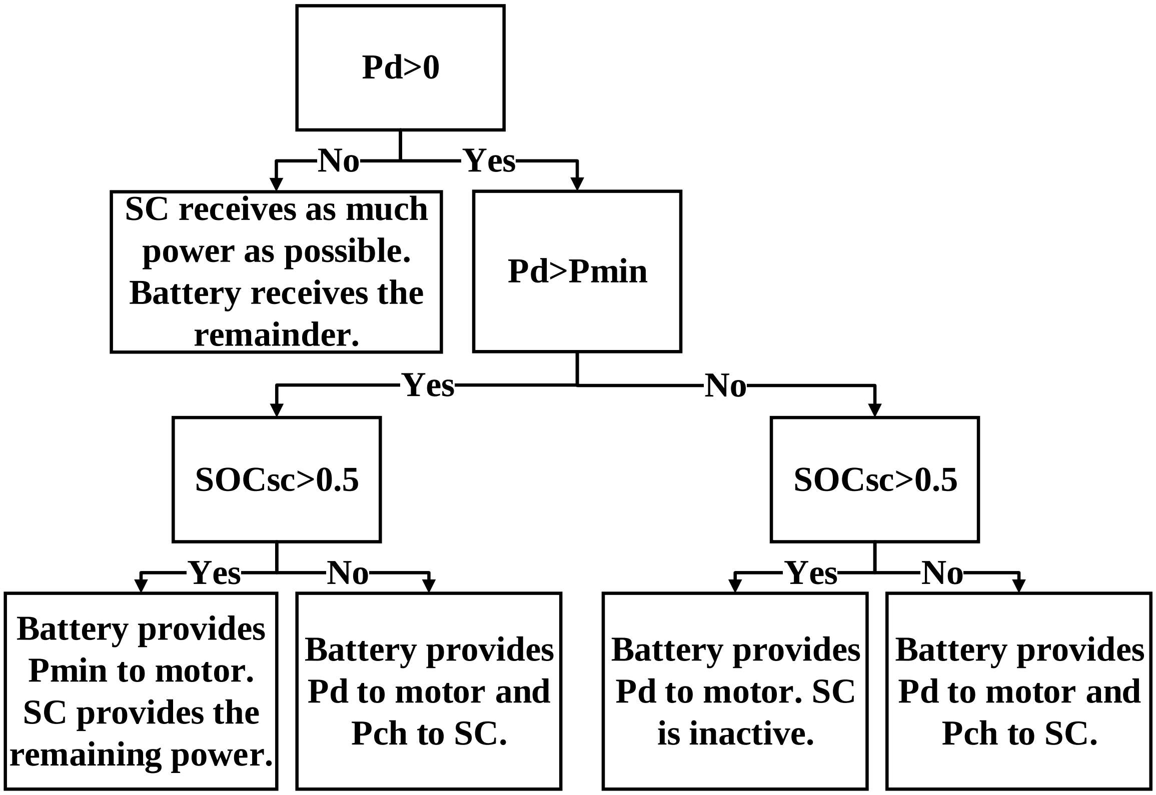
Pd>0
No
Yes
SC receives as much power as possible. Battery receives the remainder.
Pd>Pmin
Yes
No
SOCsc>0.5
SOCsc>0.5
Yes
No
Yes
No
Battery provides Pmin to motor. SC provides the remaining power.
Battery provides Pd to motor and Pch to SC.
Battery provides Pd to motor. SC is inactive.
Battery provides Pd to motor and Pch to SC.

Figure 4.1: Rule Based Controller flow-chart [1]

4.2 Nonlinear Model Predictive Control

Model predictive control (MPC) takes knowledge of the future outputs, over a prediction horizon of length N_p, into account for decision making. It does so by using some form of the predicted future power demand and the control-oriented model of the system, introduced in section 3.3. Considering the current state of the high-fidelity model and using the control-oriented model, an open-loop finite horizon optimal control problem is solved at each time step, to investigate the optimal power split at each time interval of a control horizon of the length N_c. The use of a nonlinear or linear control-oriented model and optimization solver defines whether the controller is called nonlinear model predictive control (NMPC) or linear model predictive control (LMPC). The LMPC requires lineariztion of the control oriented model and constraints in advance, to allow compatibility with the linear solvers. The optimizer calculated results for the first control is then applied to the system and the rest of the control sequences are discarded. The new state obtained by applying the calculated control to the high-fidelity model is used to update the initial condition of the optimization problem. Discussions have been made in section 5.2 on how the length of N_p and N_c have been chosen for the purpose of this work.

4.2.1 Optimal Control Problem

Specifically, we wish for the MPC to minimize the following quadratic cost function over a fixed and finite prediction horizon $[t_0, t_0 + T]$:

$$\min_{x(.),u(.)} \int_{t_0}^{t_0+T} J(x(t), u(t))dt = \min \int_{t_0}^{t_0+T} \left(w_1 I_b^2(t) + w_2 \big(SOC_{SC}(t) - SOC_{SC,ref}\big)^2 \right) dt. \tag{4.1}$$

As the battery is considered the higher resistance component of the system, by reducing the battery current square, I_b^2, the total amount of energy wasted as heat $(R_b I_b^2 + R_{SC} I_{SC}^2)$ is reduced. Furthermore, minimizing I_b^2 decreases the battery duty, and thus, increases the battery lifespan [6]. It should be noted that, as I_b is proportional to $S\dot{O}C_b$, the term I_b^2 in the cost function can be substituted by $S\dot{O}C_b^{\,2}$ for simplicity. Minimizing $S\dot{O}C_b^{\,2}$ also indicates reducing the fluctuation in battery SOC and according to Eq. 3.5 leads to a smaller $|P_b|$. We also wish for the SOC_{SC} to remain mostly half way in the allowable range ($SOC_{SC,ref} = 0.75$), in order for the system to be prepared for upcoming unknown power demands in the future [29]. Further analysis like those considered by Choi et al. [24] could be applied for finding a more efficient reference for SOC_{SC}. The positive values w_1 and w_2 establish the weights for different parts of the cost function and are determined as tuning factors with respect to deterministic or stochastic NMPC.

An optimal control problem can be written corresponding to the performance index of Eq. 4.1 with respect to the discretized form of the system dynamics indicated in Eq. 3.5, subject to the following constraints:

$$\begin{aligned} SOC_{b,min} \leq SOC_b \leq SOC_{b,max}, \\ SOC_{SC,min} \leq SOC_{SC} \leq SOC_{SC,max}, \end{aligned} \tag{4.2}$$

and

$$\begin{aligned} P_{b,min} \leq P_b \leq P_{b,max}, \\ P_{SC,min} \leq P_{SC} \leq P_{SC,max}, \end{aligned} \tag{4.3}$$

where $SOC_{.,min}$ and $SOC_{.,max}$ are the battery and SC lower and upper allowable limits for SOC, respectively. The values for these bounds have been mentioned in the previous section. Similar equations are available for battery and SC power as indicated in Eq. 4.3. However, in practice, it has been observed that power never reaches the considered bounds and therefore Eq. 4.3 has been neglected for the sake of simplicity.

4.2.2 Direct Shooting Methods

Direct methods transform the optimal control problem of interst into a finite dimensional nonlinear programming problem (NLP). The NLP is afterwards solved using a state of the art numerical optimization method e.g. SQP or interior point. In our problem, the continues-time dynamics of the system are first discretized for time intervals of $T_c = 1s$. The optimal control problem has been discretized using the direct single-shooting and multiple-shooting methods. As the size of the nonlinear programming (NLP) problem is considered to be small to medium scale (depending on the size of N_c), not much of an improvement has been observed by using the multiple-shooting approach over single-shooting. Although the direct multiple shooting method is known to decrease the number of iterations needed for convergence, with respect to the single shooting method, by incorporating information about the behavior of the state trajectory into the initial guess of the iteration process, it also increases the size of the problem by adding additional equality constraints [39]. It has been observed through simulations that the overall effect in computational time is not notable for the proposed NLP size. As a result, the single-shooting method has been adopted as the main discretizing method through this work. The optimal control problem in the form of single and multiple shooting have next been briefly presented. For more details on the shooting methods, the reader is referd to [65].

4.2.2.1 Single Shooting

The single shooting approach starts by discretizing the controls $u(t)$ over the prediction horizon $[t_0, t_0+T]$ into N_c time intervals, $t_0 < t_1 < ... < t_{N_c}$ where the controls are assumed to be piecewise constant, $u(t) = q_i$ for $t \in [t_i, t_{i+1}]$ and $i = 0, 1, ..., N_c$. Doing this, $u(t)$ only depends on a finite number of control parameters $q = (q_0, q_1, ..., q_{N_c})$ and can be denoted as $u(t; q)$. By using a numeric simulation routine for solving the initial value problem,

$$x(t_0) = x_0, \quad \dot{x(t)} = f(x(t), u(t; q)), \quad t \in [t_0, t_0 + T],$$

the states $x(t)$ on $[t_0, t_0 + T]$ can be considered as dependent variables and are denoted as $x(t; q)$. Throughout this work, a linear approximation routine has been used for solving the ODE above, in which $\dot{x}$ is substituted with $\big(x(t_{i+1}; q) - x(t_i, q)\big)/T_c$. The function $f(.)$ is defined from the control oriented model presented in Eq. 3.5. The state constraints (also known as the path constraints) are also discretized and required to be satisfied at the grid points t_i. The following integral can be numerically calculated; Thus, the considered finite-horizon NMPC problem based on Eqs. 3.5, 3.6, 4.1 and 4.2 in the single shooting format has the form of the NLP presented [65];

$$\begin{aligned}
&\min_{q} \int_{t_0}^{t_0+T} J(x(t; q), u(t; q))dt, \\
&\text{s.t.} \\
&\quad X_{min} \le x(t_i; q) \le X_{max} \qquad i = 0, 1, ..., N_c \quad , \\
&\quad U_{min} \le q_i \le U_{max} \quad , \\
&\quad v_j = \begin{bmatrix} 1 & \eta_c \end{bmatrix} q_j \qquad j = 0, 1, ..., N_p \quad ,
\end{aligned} \tag{4.4}$$

where $x(.) = \begin{bmatrix} SOC_b(.) \\ SOC_{SC}(.) \end{bmatrix}$ is the state of the system, $q = \begin{bmatrix} P_b \\ P_{SC} \end{bmatrix}$ is the input and $v = P_d$ is the measured disturbance to the system. The prediction horizon is defined such that $N_c \le N_p$. The optimal control problem also includes a linear constraint which has been discussed in Eq. 3.4 where η_c is the efficiency of the converter. The NLP problem 4.4 is solved using a Sequential Quadratic Programming (SQP) solver that is further discussed in the next section. When using single shooting, unlike multiple shooting, only initial guesses for the controls are required by the solver. Although, this causes the system to be simple to initialize, provides no margin for taking into account knowledge of the state trajectory obtained by the previous time step.

The proposed problem has 2 states and 2 controls, which entails solving an NLP problem with $2 \times N_c$ variables and $4 \times N_p$ inequality constraints, when applying a direct single shooting discretization method.

4.2.2.2 Multiple Shooting

The direct multiple shooting similarly starts with discritizing the controls piecewise;

$$u(t) = q_i \quad for \quad \in [t_i, t_{i+1}],$$

where the time intervals could be the same size as single shooting. However, the ODEs are solved independently on each time interval $[t_i, t_{i+1}]$, starting with an artificial initial value b_i;

$$\begin{aligned} &x_i(t_i) = b_i, \\ &\dot{x_i} = f(x_i(t), q_i), \qquad t \in [t_i, t_{i+1}]. \end{aligned} \tag{4.5}$$

By solving these initial value problems, trajectory pieces $x_i(t; b_i, q_i)$ are obtained, where each trajectory is dependent on the initial value and controls. Simultaneously with solving the ODE above, the following integral is numerically computed [65];

$$l_i(b_i, q_i) = \int_{t_i}^{t_i+1} J(x_i(t; b_i, q_i), q_i) dt.$$

The artificial degree of freedom imposed by b_i is afterwards constrained by adding an additional continuity constraint to the NLP formulation;

$$b_{i+1} = x_i(t_{i+1}; b_i, q_i).$$

Thus the following NLP is obtained that contains an extra variable b;

$$\min_{b,q} \sum_{i=0}^{N_c-1} l_i(b_i, q_i),$$

$$\begin{aligned}
&b_0 = x_0, && && \text{(initial value)}\\
&b_{i+1} = x_i(t_{i+1}; b_i, q_i) && i = 0, 1, ..., N_c - 1, && \text{(continuity)}\\
&X_{min} \le x_i(t_{i+1}; b_i, q_i) \le X_{max} && i = 0, 1, ..., N_c, && \text{(discretized path constraint)}\\
&U_{min} \le q_i \le U_{max}, && && \text{(control constraints)}\\
&v_j = \begin{bmatrix} 1 & \eta_c \end{bmatrix} q_j && j = 0, 1, ..., N_p, &&
\end{aligned}$$

The above NLP problem could be summarized as follows and solved using a sequential quadratic programming (SQP) method;

$$\min_{w} F(w) \qquad \text{s.t.} \qquad G(w) = 0, \qquad H(w) \le 0, \tag{4.6}$$

where $w = (s_0, q_0, s_1, q_1, ..., s_{N_c}, q_{N_c})$ contains all the optimization variables. It is important to note that although the state trajectory discretization is not constrained to match the control trajectories, it has been chosen point for point for the sake of simplicity. The NLP obtained by the multiple shooting method is known to converge in a fewer number of iterations in comparison to the single shooting method, as knowledge of the states at the previous time step are taken into account as initial guesses for the SQP solver. However in practice, the increase in the size of the optimization problem (caused by the artificial variable b_i and the added continuity constraint) compensates for the computational time caused by the fewer number of iterations. As a result, in the problem of interest, an improvement in the computational time was not observed by using multiple shooting over the single shooting approach and the single shooting approach has been dominated throughout this work for simplicity.

4.2.3 Deterministic NMPC

As mentioned before, the power demand in the NMPC problem is considered a measured disturbance and needs to be estimated. If the power demand estimation over the prediction horizon results in a single scenario, a deterministic NMPC (D-NMPC) could be solved for the power split problem of the battery/SC HESS. In the D-NMPC the optimal control problem presented in Eq. 4.4 is solved by simply substituting the vector $V = (v_0, v_i, ..., v_{N_p})$ with the estimated power demands over the prediction horizon. The FT-NMPC, TP-NMPC and P-NMPC introduced in section 3.5 are all deterministic NMPCs and produce a single vector estimation of the future power demand.

4.2.4 Two-stage Stochastic NMPC

Stochastic NMPC uses available statistical information from the past in order to minimize the expected value of the performance index. Unlike what was done in TP-NMPC, no on-board technology is used for future trip predection. As the D-NMPC formulations does not present a systematic way of including uncertainties, the formulation of Eq. 4.4 has been modified to fit the form of a stochastic programming problem. The S-NMPC adopted in this work exploits the ideas form the two-stage stochastic optimization method [66, 67]. The method relies on two sets of decision variables. The first set consists of the same decision variables presented in the deterministic optimal control problem in Eq. 4.4, while the second set of decisions are artificial variables taking into account the stochastic nature of the problem. The performance index is accordingly modified, to include both sets of decision variables.

The method captures future uncertainties based on knowledge of stochastic parameters, provided through scenarios generated in section 3.5.3.1, leading to an increase in the number of constraints that include the stochastic variables with respect to the number of scenarios involved. Applying these modifications to Eq. 4.4 results in the open-loop, two-stage stochastic optimal control problem bellow:

$$
\begin{aligned}
&\min_{q,s} \Big\{ \int_{t_0}^{t_0+T} J(x(t;q), u(t;q))dt + w_3 \sum_{k=1}^{N_S} Pr_k s_k \Big\}, \\
&\quad \text{s.t.} \\
&\qquad X_{min} \le x(t_i;q) \le X_{max} \qquad i = 0, 1, ..., N_c \\
&\qquad U_{min} \le q_i \le U_{max} \\
&\qquad v_{1,j} = \begin{bmatrix} 1 & \eta_c \end{bmatrix} q_j + s_1 \qquad j = 0, 1, ..., N_p \\
&\qquad v_{2,j} = \begin{bmatrix} 1 & \eta_c \end{bmatrix} q_j + s_2 \\
&\qquad \vdots \\
&\qquad v_{N_S,j} = \begin{bmatrix} 1 & \eta_c \end{bmatrix} q_j + s_{N_S}
\end{aligned}
\tag{4.7}
$$

where w_3 is a weighting factor, s_k, are the second stage variables, N_S is the number of scenarios involved in the stochastic method, Pr_k is the occurrence probability of each scenario and $v_{k,j}$ indicates the measured disturbance estimated by the kth scenario at the jth time step of the prediction horizon. As only one power split could be adopted at each time instant, but N_S scenarios have been predicted, the second stage variables, s_k, are used to compensate for the difference associated with the linear equality constraints. These compensating variables are minimized in the performance index with respect to the occurrence probability of their associated scenario. As the transition probabilities have been already adopted in the scenario generation process, the probability of $\frac{1}{N_S}$ has been used at this point for each scenario within the performance index.

4.2.5 Numerical Solver

The proposed NLP for the D-NMPC has been solved using a self-coded sequential quadratic (SQP) solver in Matlab. The stated solver has been validated using Matlab's $fmincon$. Since Matlab's SQP in $fmincon$ cannot be easily used in code generation for HIL purposes, a self-coded SQP has been developed to make HIL implementations possible. Further explanation on this subject is available in section 5.2. Coding our own SQP also allows for algorithm modification for potential improvement in the future e.g. modifying the constraint handling method within the algorithm. As of now, the author has tried to faithfully follow $fmincon$'s SQP algorithm as stated in [68]. In Fig. 4.2, the results of the optimal control problem, indicated in Eq. 4.4, using the author's coded SQP has been presented and compared to the results obtained by Matlab's $fmincon$, assuming perfect power prediction over the UDDS driving cycle, while $N_p = N_c = 10$. The results are shown to be identical, up to three decimal digits. In order to obtain real-time results, the solver has been limited to perform only two iterations per sampling time i.e. the optimal control problem is only approximately solved during each iteration. More details on this are provided in the next sections.

The results in Fig. 4.2 are obtained using an open-loop controller and assuming full-knowledge of the future trip. As model uncertainties coming from the high-fidelity model are not incorporated, the maximum potential of the NMPC strategy can be observed. It can be noticed how the battery remains the primary power reservoir, outlining the power needed through the trip, while the SC takes care of all the fluctuations in power demand. The procedure followed by the self-coded SQP solver is next discussed.

Due to the size of the stochastic NLP, neither Matlab's SQP solver nor the self-coded SQP have been able to solve the S-NMPC optimization problem of interest. The results demonstrated for S-NMPC have been obtained using Matlab's interior point solver in

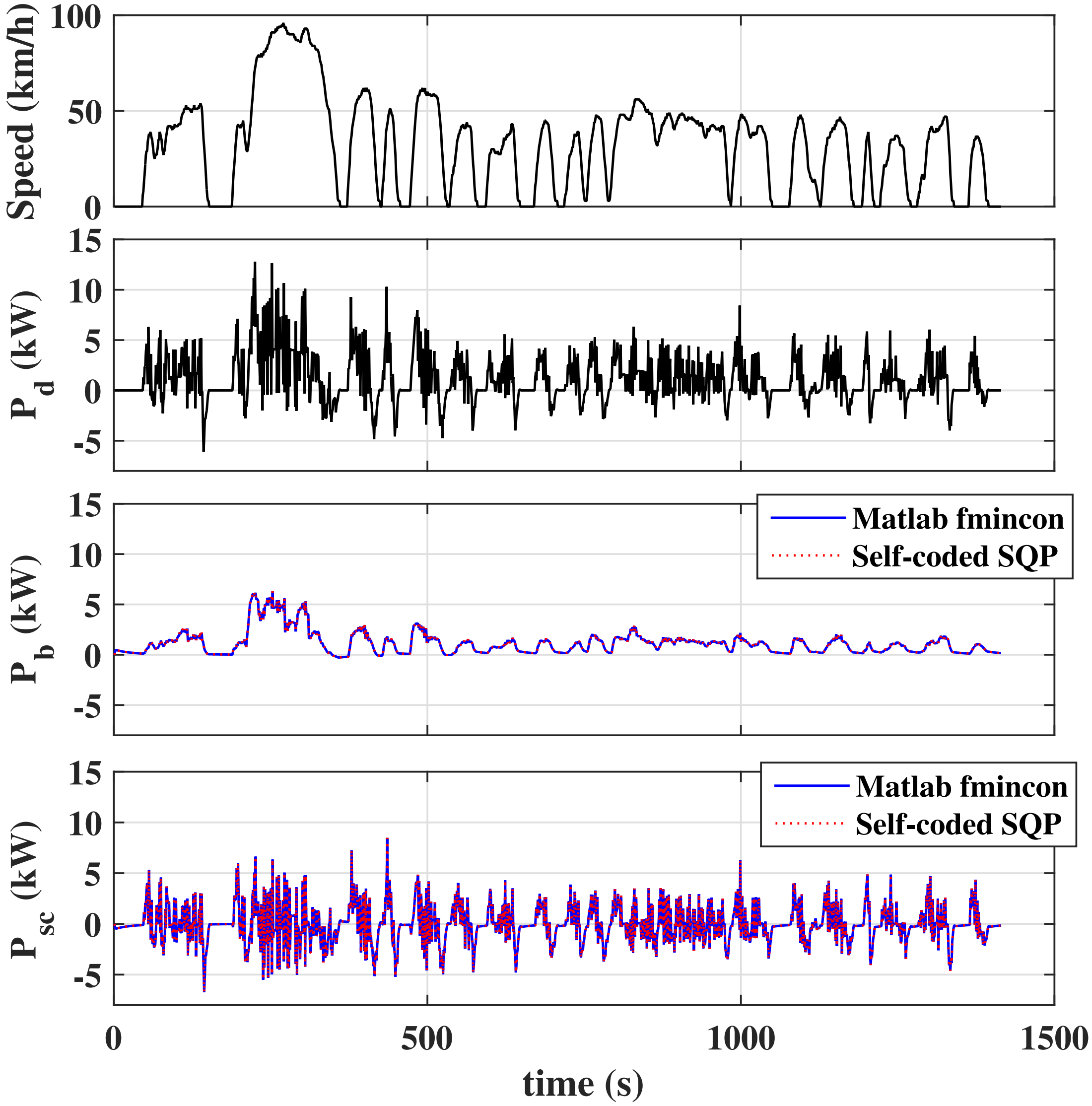

Figure 4.2: Matlab *fmincon* vs. self-coded SQP power-split results for UDDS driving cycle

fmincon. The interior point solver is known to be a better solution for larger size NLPs. The interior point solver however, was observed to be slower than SQP for the D-NMPC.

4.2.5.1 Sequential Quadratic Programming

The NLP presented in Eq. 4.4 is solved using self-coded Sequential Quadratic Programming (SQP), also known as a Newton-type solver. The author has tried to faithfully follow *fmincon*'s SQP algorithm as stated in [68]. The outline of the algorithm is however presented here.

Having an NLP problem of the general form :

$$\begin{aligned} &\min_x f(x), \\ h_i(x) = 0 \qquad & i = 1, 2, ..., p \quad , \\ g_j(x) \leq 0 \qquad & j = 1, 2, ..., m \quad . \end{aligned} \tag{4.8}$$

the "Lagrangian function" can be written in the form of:

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

where λ and μ are the Lagrangian multipliers. $h(x)$ and $g(x)$ are vectors of the equality and inequality constraints respectively. In order for the necessary conditions of a local optimum to be satisfied at a point x^*, the following must hold:

$$\begin{aligned} \nabla L_x(x^*, \lambda^*, \mu^*) = 0 \quad &, \\ h(x^*) = 0 \quad &, \\ g(x^*) \leq 0 \quad &, \\ \mu^* \geq 0 \quad &, \\ g(x^*)^T \mu^* = 0 \quad &. \end{aligned} \tag{4.9}$$

SQP uses an iterative procedure to approximately find (x^*, λ^*, μ^*). Thus, the NLP problem in Eq. 4.8 is transformed into an Quadratic Programming (QP) problem of the form of:

$$\begin{aligned} &\min_d \frac{1}{2} d^T H_k d + \nabla f(x_k)^T d, \\ \nabla h_i(x_k)^T d + h_i(x_k) = 0, \qquad & i = 1, 2, ..., p \quad , \\ \nabla g_j(x_k)^T d + g_j(x_k) \leq 0, \qquad & j \subset A\{1, 2, ..., p\} \quad , \end{aligned} \tag{4.10}$$

where H_k is the Hessian of the Lagrangian or a Quasi-Newton approximation of it; $H_k \simeq \nabla_x^2 L(x_k, \lambda_k, \mu_k)$ and $\nabla h_i(x_k)^T$ and $\nabla g_j(x_k)^T$ are the constraint Jacobians and A is the active set at the current iterate. Starting with an initial guess (w_0, λ_0, μ_0), by solving the QP in Eq. 4.10, a standard full SQP iteration could be performed by:

$$\begin{aligned} x_{k+1} &= x_k + d_k, \\ \lambda_k + 1 &= \lambda_k^{QP}, \\ \mu_{k+1} &= \mu_k^{QP}, \end{aligned} \tag{4.11}$$

where $(d_k, \lambda_k^{QP}, \mu_k^{QP})$ is the solution of the discussed QP. Practical SQP methods mostly vary in the QP solver used, the way the Hessian is approximated and the initial guess of the starting iterate. The Hessian approximation method and QP solver are next discussed.

Hessian Approximation As it might be computationally expensive to calculate the exact Hessian and Jacobian at each main iterate, an approximation of the Hessian and/or Jacobian is taken into account in a wide range of SQP variants. The quality of the Hessian approximation usually defines whether the rate of convergence is linear, super-linear or quadratic. The most widely used Hessian update is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method. As the solver implementation in this work is meant to allow real-time optimization and the use of Symbolic variables in Matlab or numerical methods for updating the Hessian has shown not to yield that purpose, a BFGS method has been used instead. The BFGS uses the exact Jacobians of the constraints, that are numerically calculated at each main iterate, but approximates the Hessian matrix $H_{k+1} \simeq \nabla_x^2 L(x_{k+1}, \lambda_{k+1}, \mu_{k+1})$ at each main iterate, using the approximation of the Hessian H_k [68] obtained at the previous iterate;

$$\begin{aligned} &H_{k+1} = H_k + \frac{q_k q_k^T}{q_k^T s_k} - \frac{H_k s_k s_k^T H_k^T}{s_k^T H_k s_k}, \\ &s_k = x_{k+1} - x_k, \\ &q_k = \Big(\nabla f(x_{k+1}) + \sum_{i=1}^{m} \lambda_i . \nabla g_i(x_{k+1})\Big) - \Big(\nabla f(x_k) + \sum_{i=1}^{m} \lambda_i . \nabla g_i(x_k)\Big), \end{aligned} \tag{4.12}$$

where λ_i is an estimate of the Lagrangian multipliers obtained from the previous iterate. While the simulation is initialized by setting the Hessian to an arbitrarily positive definate matrix, the approximation of the Hessian is improved through the iterations. The Hessian is maintained to stay positive definite [69], although it might not be positive definite at the solution point. This is done by keeping $q_k^T s_k$ positive at each update. When $q_k^T s_k$ is not positive, q_k is modified on an element-by-element bases until $q_k^T s_k > 0$ is obtained. To do so, the most negative element of $q_k^T s_k$ is repeatedly divided in half, until $q_k^T s_k$ is greater than a small negative tolerance. If after applying this modification, $q_k^T s_k$ still remains negative, q_k is modified as follows:

$$\begin{aligned} q_k =& q_k + av, \\ v_i =& \nabla g_i(x_{k+1}).g_i(x_{k+1}) - \nabla g_i(x_k).g_i(x_k), \\ & \text{if}(q_k)_i.a < 0 \quad \text{and} \quad (q_k)_i.(s_k)_i < 0, \quad i = 1, ..., m, \\ v_i =& 0 \qquad \text{otherwise}, \end{aligned} \tag{4.13}$$

where a is a constant scalar and is increased until $q_k^T s_k$ becomes positive. The purpose of this procedure is to distort the elements of q_k that correspond to a positive definite update as little as possible. Many packages have implemented BFGS or similar Quasi-Newton methods, including NPSOL, SNOPT [70] and MUSCOD-II [71].

QP Solver As mentioned previously, at each major iterate of the SQP method, the following inequality QP problem is formed and solved;

$$\begin{aligned} \min_x & \frac{1}{2} x^T Q x + q^T x, \\ & Ax = a, \\ & Bx \leq b, \end{aligned} \tag{4.14}$$

where $Q \in \mathbb{R}^{n \times n}$, is symmetric. Also, $A \in \mathbb{R}^{m_1 \times n}$, $B \in \mathbb{R}^{m_2 \times n}$, $a \in \mathbb{R}^{m_1 \times 1}$, $b \in \mathbb{R}^{m_2 \times 1}$ and $m_1 \leq n$. An iterative active set method has been used for solving this QP[72]. The method starts from a feasible initial point x_0 and the next iterate is formed by setting $x_{k+1} = x_k + \alpha_k d_k$ where α_k is a step length and d_k is the search direction. An active set $\mathcal{A}_k$ is formed from the constraints that are active at the current point x_k, i.e.;

$$\mathcal{A}_k = \{j | B_j x_k - b_j = 0, \quad j = 1, ..., m_2\}, \tag{4.15}$$

where B_j and b_j are the jth row of B and b and A_j and a_j are the jth row of A, respectively. The equality constrained QP is afterwards solved to obtain the search direction d_k;

$$\begin{aligned} \min_d & (x_k + d_k)^T Q (x_k + d_k) + q^T (x_k + d_k), \\ & a_i^T (x_k + d_k) = a_i, \quad i = 1, ..., m_1 \quad , \\ & b_j^T (x_k + d_k) = b_j, \quad j \in \mathcal{A}_k \quad , \end{aligned} \tag{4.16}$$

The equations above could be simplified to obtain;

$$\begin{aligned} \min_{d} \Big\{ \frac{1}{2} d_k^T Q d_k {+} g_k^T d_k \Big\}&, \\ A d_k &= 0, \\ \tilde{B} d_k &= 0, \end{aligned} \tag{4.17}$$

where, $\tilde{B} = [...b_j...]^T, j \in \mathcal{A}_k$ and $g_k = Qx_k{+}q$. Therefore, the active set method transforms the inequality constrained QP to an equality constrained QP [72]. The KKT optimality conditions lead to the following system of equations;

$$\begin{aligned} Qd_k + g_k + A^T\lambda + \tilde{B}^T\tilde{\mu} &= 0, \\ A d_k &= 0, \\ \tilde{B} d_k &= 0. \end{aligned} \tag{4.18}$$

Therefore, the optimization problem is eventually reduced to the solution of a (possibly large-scale) system of linear equation:

$$\begin{bmatrix} Q & A^T & \tilde{B}^T \\ A & \mathbb{O} & \mathbb{O} \\ \tilde{B} & \mathbb{O} & \mathbb{O} \end{bmatrix} \begin{bmatrix} d_k \\ \lambda_k \\ \tilde{\mu_k} \end{bmatrix} = \begin{bmatrix} -g_k \\ \mathbb{O} \\ \mathbb{O} \end{bmatrix}, \tag{4.19}$$

The system of linear equations is afterwards solved using a null-space approach. The d_k found from solving this equation is used to update the QP iterate, i.e. $x_{k+1} = x_k + \alpha_k d_k$. The step size α_k is computed as:

$$\alpha_k = \min \Big\{ 1, \frac{b_j - b_j^T d_k}{b_j^T d_k} | j \notin \mathcal{A}_k \quad \text{and} \quad b_j^T d_k > 0 \Big\}.$$

If the d_k obtained from solving this equation equals zero and $\tilde{\mu} \geq 0$ then $x_{k+1} = x_k$ is a KKT point and the QP algorithm stops. However, if some components of $\tilde{\mu}$ are negative, x_k is not an optimal solution. Therefore, the active set is updated as follows and the equality constrained QP is solved again:

$$\begin{aligned} &\mu_{j_0} = \min\{\tilde{\mu}_j | \tilde{\mu}_j < 0, j \in \mathcal{A}_k\}, \\ &\mathcal{A}_k = \mathcal{A}_k \backslash \{j_0\}. \end{aligned} \tag{4.20}$$

Therefore, other than the main iterations of the SQP solver, the QP solver performs its own sequence of sub-problem iterations.

4.3 Linear Model Predictive Control (LMPC)

The design of a LMPC as the EMC of the HESS uses the same objective function. The format of the objective function in Eq. 4.1 does not need to be modified, as it is already in quadratic form. However, the values w_1 and w_2 are re-tuned for maximum performance. Moreover, the nonlinear control-oriented model of the system is linearized around the current operating conditions to form a linear system of the form:

$$\begin{aligned} \dot{x} &= Ax + B_u u + B_v v + F, \\ y &= Cx + D_u u + D_v v + G, \end{aligned} \tag{4.21}$$

where

$$\begin{aligned} A &= (\frac{\partial f}{\partial x}), & B_u &= (\frac{\partial f}{\partial u}), & B_v &= (\frac{\partial f}{\partial v}), \\ C &= (\frac{\partial g}{\partial x}), & D_u &= (\frac{\partial g}{\partial u}), & D_v &= (\frac{\partial g}{\partial v}), \end{aligned}$$

$$\begin{aligned} F &= f(x_k, u_k, v_k) = Ax_k - B_u u_k - B_v v_k, \\ G &= g(x_k, u_k, v_k) = Cx_k - D_u u_k - D_v v_k, \end{aligned} \tag{4.22}$$

and x_k, u_k and v_k are the current states, inputs and measured disturbances to the system, respectively. The Matlab MPC toolbox has been used for LMPC implementation [73].

4.4 Dynamic Programming

An offline Deterministic Dynamic Programming (DP) optimal control strategy has been performed for maximum potential investigation. As DP is capable of providing the global solution to an optimal control problem, the best power split between the battery and SC could be found for drive cycles known a priory. Although far from real-time implementation, DP can also be performed online as the nonlinear solver for model predictive controllers. As DP requires a discrete model, the controls, u, and control-oriented state model 3.5 are discretized with the time constant of $T_c = 1s$;

$$x_{k+1} = \tilde{f}(x_k, u_k); \qquad k = 0, 1, ..., N-1 \quad , \tag{4.23}$$

were N is the length of the known driving cycle while x_k and u_k present the state and control at time k respectively. We aim to minimize;

$$\begin{aligned} J = \min \sum_{k=0}^{N-1} g(x_k, u_k) = \min \sum_{k=0}^{N-1} w_1 (I_b)_k^2 + w_2 \big((SOC_{SC})_k - SOC_{SC,ref}\big)^2, \\ (I_b)_k = \frac{(SOC_b)_k - (SOC_b)_{k-1}}{T_c}, \end{aligned} \tag{4.24}$$

where $x_k = \begin{bmatrix} (SOC_b)_k \\ (SOC_{SC})_k \end{bmatrix}$ and $u_k = \begin{bmatrix} (P_b)_k \\ (P_{SC})_k \end{bmatrix}$. By applying the principle of optimality, and moving backwards in time, the cost function can be written in the following from:

$$\begin{aligned} J^*_{N-k,N}(x_{N-k}) = \min_{u_{N-k}} \{ g(x_{N-k}, u_{N-k}) + J^*_{N-(k-1),N}(\tilde{f}(x_{N-k}, u_{N-k}) \}, \\ k = 1, 2, ..., N \quad , \end{aligned} \tag{4.25}$$

In the equation above, the optimal cost for the K-stage policy of an N-stage process, $J_{N-k,N}$ can be calculated by knowing the optimal cost for the (K-1)-stage policy. The algorithm starts at the zero-stage (at the terminal state N) where $J^*_{NN} = 0$. The solution of recurrence equation stated in Eq. 4.25 is the global optimal control policy $u^*(N-k)$, $k = 1, 2, ..., N$, which is obtained by trying all feasible control values at each feasible state value. This is done by dividing the feasible state and control values into finite grids. As our system of interest has two states, the grid of the state values would be two-dimensional. More details on the DP method could be found in [74].

4.5 Chapter Summary

The various energy management controllers used for finding the optimal power split between the battery and SC in the proposed HESS are introduced in this chapter. The RBC, NMPC, LMPC and DP energy management approaches are discussed. The optimal control format of the deterministic and stochastic NMPC methods, consisting of an objective function, control-oriented model and constraints are presented and the SQP solver developed for solving the optimal control problem is discussed. Additionally, the shooting method used for discritizing the optimal control problem is thoroughly presented.

Chapter 5

Energy Management Controller Evaluation

In this chapter, the tests performed for controller evaluation are introduced and test results are presented. Comparative results have been demonstrated for the performance of various controllers discussed in chapter 4. Simulation results on a closed-loop high-fidelity model have been presented over a combination of different driving cycles while the controllers attempt to obtain the optimal power split at each time step. Results have been discussed in two main categories of deterministic and stochastic NMPC. As real-time implementation of nonlinear predictive controllers is still under question in the literature, HIL testing has been performed for D-NMPC to address this issue. It has been observed that upon choosing the prediction horizon and control horizon length as well as the maximum number of iterations, the turnaround time for the control update is shown to fall far below the necessary sampling time in vehicle control. The correlation between each of these parameters and turn-around time has been presented in this chapter. Further investigation on real-time implementation of S-NMPC is still in hand.

The chapter is organized as follows: section 5.1 and 5.2 respectively introduce MIL simulating and HIL testing for control performance evaluation. In section 5.3 and 5.4 evaluation and comparison results are presented and discussed for deterministic and stochastic NMPC, respectively.

5.1 MIL Simulation

The off-line simulations used within the early phases of the development process are often called MIL. A complete block diagram of the system is shown in Fig. 3.1. It consists of a high-fidelity model of the system, as proposed in section 3.2, in order to evaluate the performance of the employed controller. The high-fidelity model substitutes the actual vehicle within the control loop to examine the actual response of the vehicle, while the control-oriented model is used in the controller for approximate, fast prediction of the vehicle's response in order to command proper action. The driver consists of a PID controller and follows the desired reference speed. In MIL simulations the controller runs on the same computer system as the high-fidelity model. This is important to point out as it is not the case in HIL simulations.

5.2 HIL Testing

Hardware-in-the-Loop (HIL) testing has been applied to evaluate real-time computational speed of the Electronic Control Unit (ECU); embedded with the proposed controller. HIL testing takes into account the communication issues and controller computational limitation and therefore the results obtained are considered to be more practical. As HIL testing is more cost effective and faster than field/laboratory experiments, it is often performed prior to manufacturing the prototype vehicle. In this work, dSPACE systems have been used for HIL testing. Generally, the HIL testing setup consists of (1) an independent processing unit to run the controller; namely the ECU, (2) a powerful simulator to run the plant model in real-time and (3) a communication channel between the ECU and the plant (real-time target) namely, a Control Area Network (CAN) bus. The system is also attached to a personal computer (PC) in order to monitor the results. During each time step, the real-time simulator sends the state variables to the ECU through the CAN bus. The ECU calculates the optimal control command and sends it to the real-time simulator.

As the development of an implementable ECU for the vehicle requires proper hardware design and code optimization, it could be considered a challenging task. Rapid Control Prototyping (RCP) systems have been developed to reduce the difficulties and enhance this processing time. These RCP systems provide both hardware and code generation software. In this work, the dSPACE MicroAutoBox II has been used as the ECU for testing real-time performance of the EMC.

The high-fidelity model of the system in Simulink, including the driver, power estimator,

Table 5.1: Specification of the DSPACE HIL components

Component	Part	Specification
Real-time Simulator	Processor	DS-1006 Quad-Core AMD, 2.8 GHz
	Memory	1 GB local, 4x128 MB global memory
	HIL I/O Board	DS-2202
ECU: MircoAutoBox II	Processor	DS-1401 PowerPC 750GL 900 MHz
	Memory	16 MB main, 16 MB nonvolatile memory
	I/O interface	DS-1511
Interface	Processor	Core i7, 3.4 GHz
	Memory	16 GB

powertrain and HESS high-fidelity model, has been converted into a C-code using the dSPACE Real Time Workshop code generator and uploaded to the DS1006 processor board (real-time simulator). The designed controller in Matlab/Simulink is also converted into a C-code separately. Each hardware platform corresponds with its own compiler and code generator (rti1401.tlc and rti1006.tlc for MicrroAutoBox (DS1041) and Simulator (DS1006), respectively). These modules are able to communicate over the CAN bus. More details on the described componenets are available in Table 5.1 [75]. An architecture of the HIL setup is presented in Fig. 5.1.

As code generation is only allowed in Simulink using a Matlab function block [76] many predefined Matlab functions, including $fmincon$, are not compatible to be used within this block. Efforts have been made to code a fast SQP solver in Matlab that is also compatible for code generation. This has been done by following $fmincon$ guidelines indicated in [68]. As explained in section 4.2.5, the coded SQP obtains identical results to the $fmincon$'s SQP, upto three decimal digits, within the second iteration. The calculation time of the proposed SQP is also shown to be comparable to $fmincon$'s SQP on a PC with an Intel Core i7, 3.60 GHz processor, 16 GB RAM memory and a Windows 10 operating system with the NMPC average simulation time being 22.62s and 14.59s for the $fmincon$'s SQP and author's SQP respectively, over an FTP75 driving cycle with $N_p = N_c = 10$, and two iteration per sampling time in place. The maximum turn-around time for different prediction and control horizons and maximum number of iterations are reported in this section. The turn-around time is the time taken for executing the controller code and

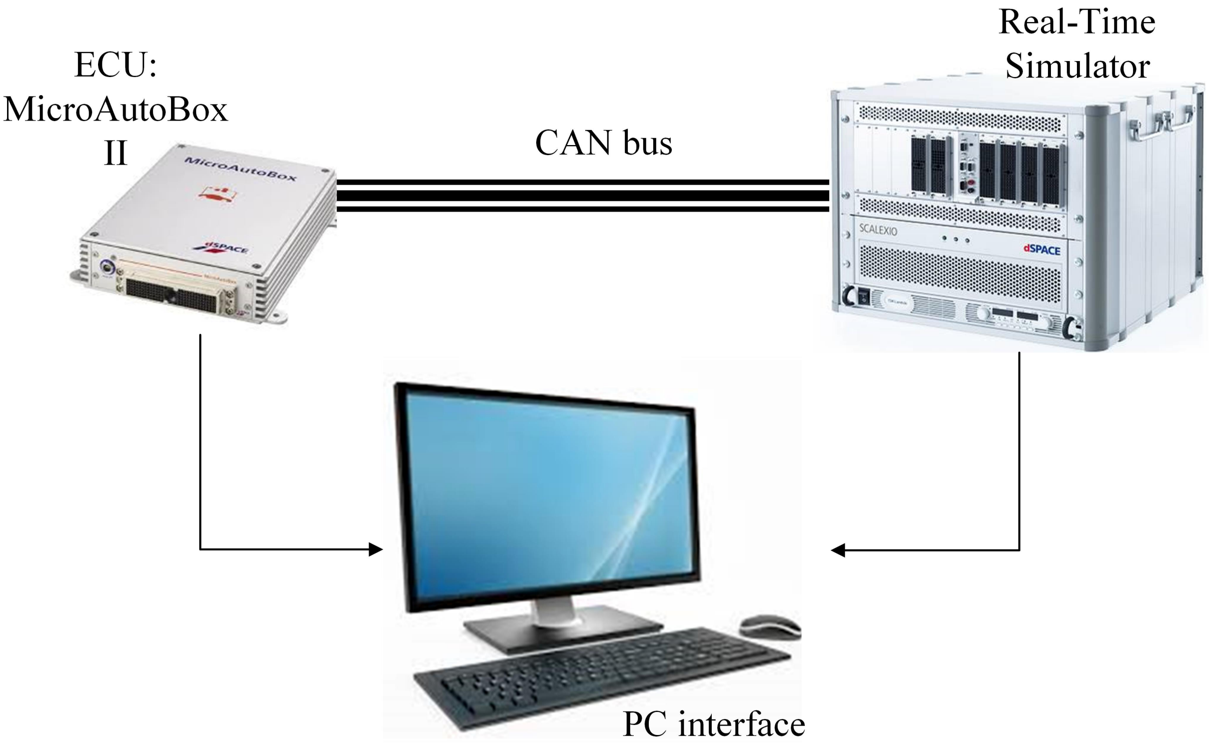

Figure 5.1: Schematic of the HIL setup

providing ECU output. As the turn-around time has to be less than 10 milliseconds for real-time applications in vehicle control [77], we have set our upper bound for the turn-around time of the MicroAutoBox to be less than 1 millisecond to compensate for the slightly slower computational time of the actual ECU on the vehicle. The maximum turn-around time is of our interest as it is necessary for the turn-around time of all sample times to fall bellow the allowable range.

The initial control value for each time step is set to the solution obtained from solving the last step for a warm-start. The warm-start technique can effectively reduce the number of iterations necessary for the SQP convergence and allows the algorithm to start from a feasible initial guess [78, 39].

5.3 Deterministic NMPC Results

In order to enable real-time results, the number of main iterates of the SQP solver has been fixed to be less than a certain small number ($K_{max} = 1 - 5$). This prevents the NLP from fully converging to the solution that lies within the tolerance distance from the local minimum. However, as the SQP is moving towards the local minimum, the optimal control problem is approximately solved. It has been observed through simulations that although increasing K_{max} linearly increases the computational time, for the first 5 iterations, the improvement in the performance is minor (Fig. 5.2 and Fig. 5.3). The norm of the residual has been set to be smaller than 10^{-5} in order for the stopping criteria to be met. It can be deduced that within the first 5 main iterations, the order of improvement in the solution is minor for most time steps. The stopping criteria, however, has not been met, as the computational time still increases by increasing K_{max}. The constraints are always satisfied since the proposed SQP uses an active set method for solving the derived QP and the algorithm always starts with an initial feasible guess. The QP sub-problem is permitted to perform a larger number of iterations. This is allowed because the computational time shows to already fall in the permissible range by merely limiting the main iteration of SQP, K_{max}.

In Fig. 5.4, the energy management results of the HESS are shown over a combination of the driving cycles, FTP75, US06 and HWFET respectively, for the proposed NMPC SQP solver using the three aforementioned power demand estimators. The prediction horizon and control horizon length have been chosen to be 10 while the maximum number of iterations per time-step have been limited to 3. More discussions on choosing the proper prediction horizon and control horizon length and the maximum number of iterations are presented in the next section. It can be noticed from Fig. 5.4 that through the trip, the regenerative braking is completely handled by the SC and the SOC_{SC} stays well in the bounds defined.

The current-squared of the battery can be an indicator of the system's efficiency, as it is related to the system's heat loss and battery lifespan [5, 79]. Fig. 5.5 compares the battery current-squared sum of the NMPC with three power demand estimations, to that of the RBC, the LMPC with no prior trip information (FT-LMPC) and the case with no SC in use. The lengths of N_p and N_c as well as K_{max} have been tuned for optimal performance within the allowable computational time of 1 milliseconds for vehicle control applications. This has been discussed in details in section 5.2 while the tuning parameters are introduced to be $N_p = N_c = 10$ and $K_{max} = 3$ for the P-NMPC and TP-NMPC strategy and $N_p = N_c = 2$ and $K_{max} = 6$ for the FT-NMPC system. It can be noticed that the NMPC solver has a lower battery current-squared sum in comparison to the RBC, even

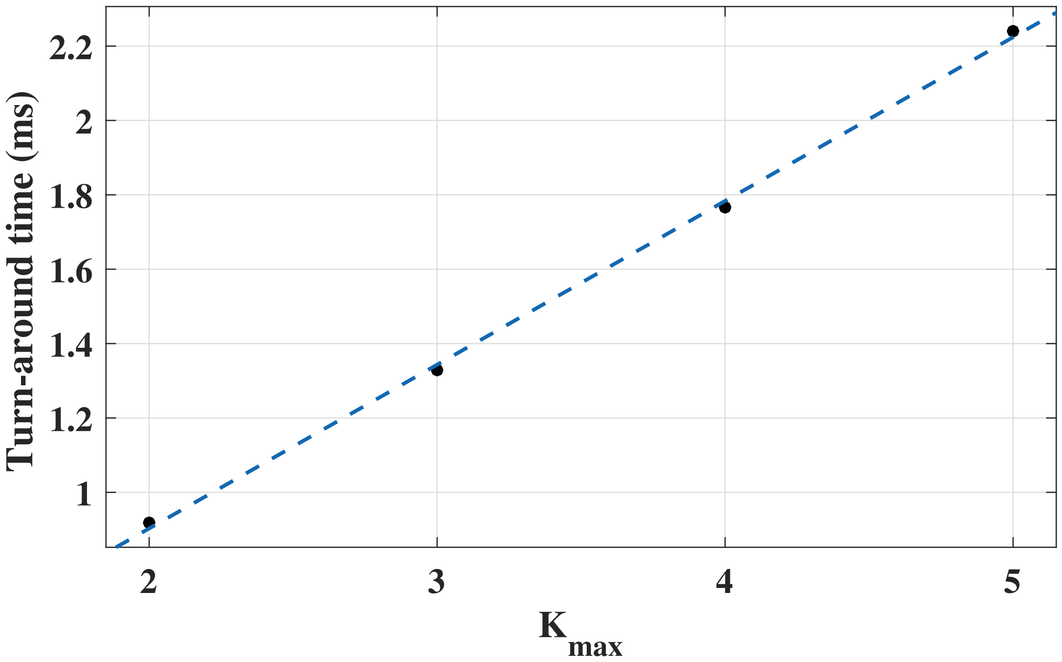

Figure 5.2: ECU maximum turn-around time for different maximum number of iterations, $N_p = 15$, $N_c = 5$

with no prior knowledge of the future trip (FT-NMPC), while this is not the case for the LMPC. Prior knowledge, however, does improve the NMPC performance. It is observed that while using trip-planing (TP-NMPC), the performance is closer to that of the P-NMPC where full-knowledge of the future trip is assumed. Another interesting observation is that the FT-NMPC shows to outperform the FT-LMPC. It is observed that, even though the SQP solver in the NMPC is stopped prematurely (limited to 3 iterations) and only an approximate optimal solution is obtained, as the NMPC compensates for the errors caused by linearization in LMPC, the aggregated result shows an improvement in the NMPC over the LMPC. The range of the vehicle (indicated by the change in the battery SOC) however, does not show significant improvement for any of the EMCs used in comparison to the case with no SC in use. Many others have used the battery current root mean square (RMS) as an indicator of the battery lifespan [80, 25]. While many stress factors, as in high fluctuations in battery SOC, high rates of power demand and operation in low and high temperatures have shown to be effective in battery aging [79, 81], the battery current

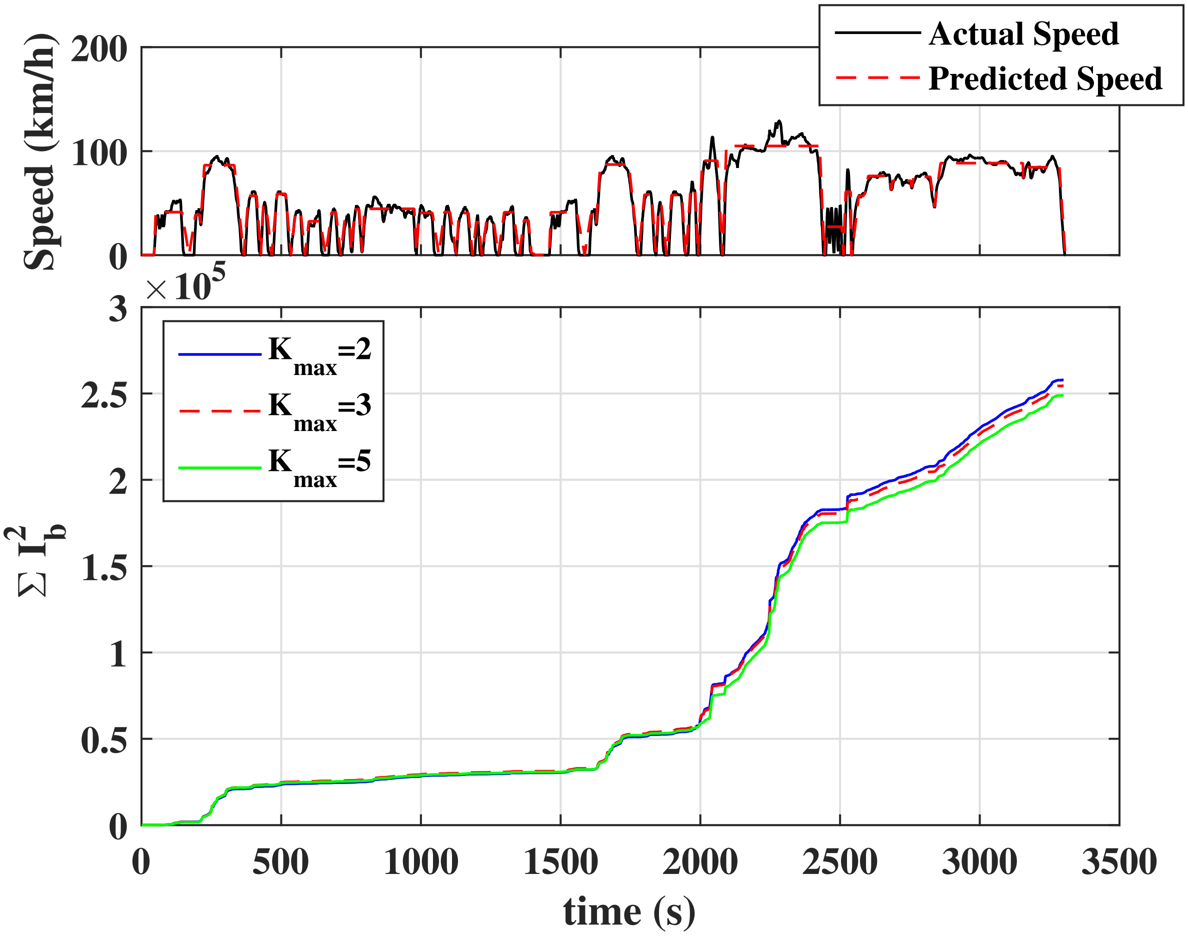

Figure 5.3: Comparison of the battery current-squared sum for different maximum number of iterations in TP-NMPC, $N_p = 15$, $N_c = 5$

RMS has been used as an indicator of these aging parameters and is defined as [80, 25]:

$$I_{b,RMS} = \sqrt{\frac{1}{T_f}\sum_{i=1}^{T_f}(I_b)_i^2}, \tag{5.1}$$

where $I_{b,RMS}$ is the battery current RMS, $(I_b)_i$ is the battery current at the ith time step and T_f is the trip duration with the discretizing time constant of $T_c = 1s$. The battery

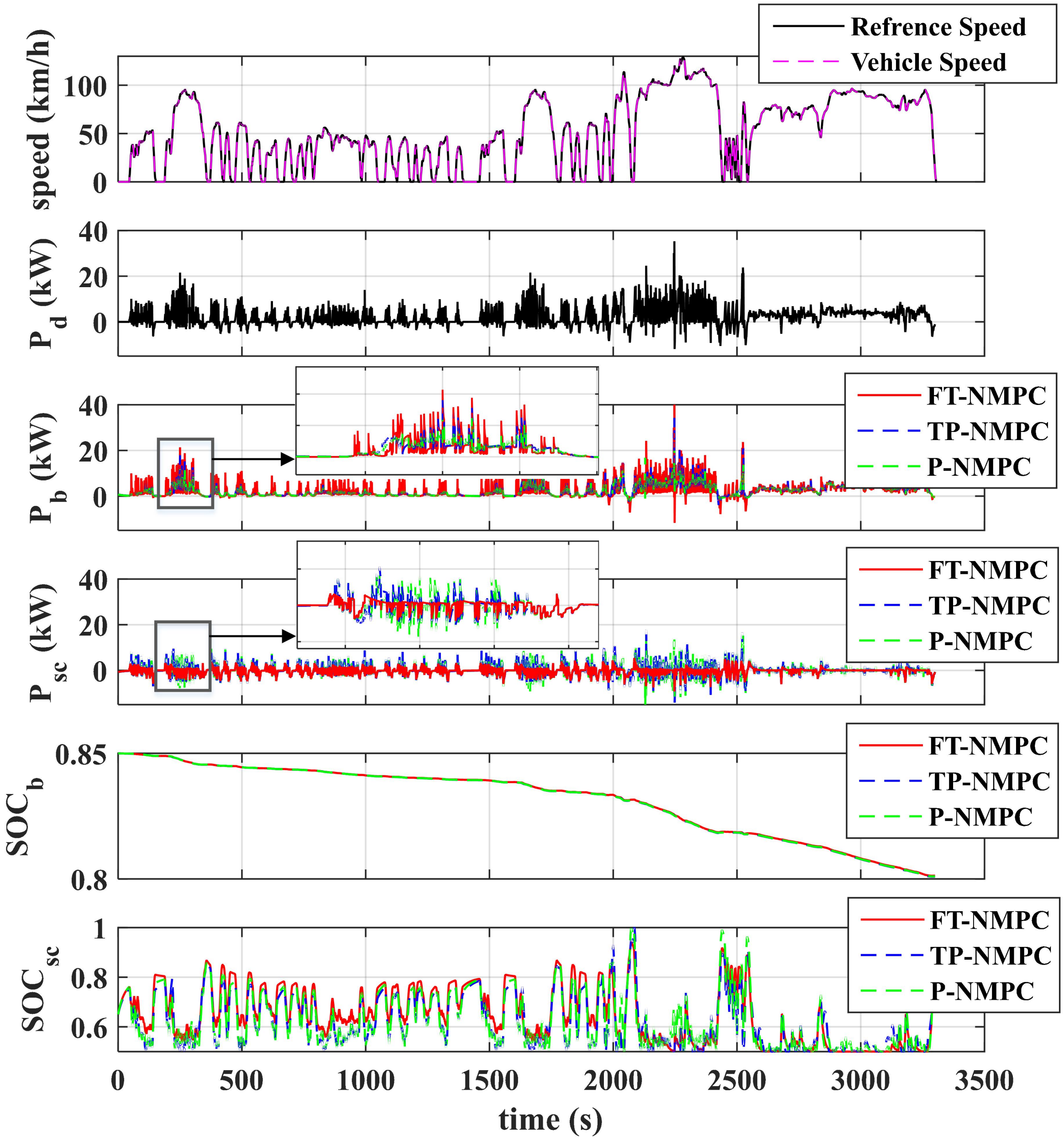

Figure 5.4: NMPC MIL results over combined FTP75, US06 and HWFET driving cycles with $N_p = N_c = 10$ and $K_{max} = 3$

current RMS is also related to the heat-loss of the system and therefore can be a measure for efficiency. By looking at Fig. 5.6, it can be noted that the battery current RMS over these three driving cycles has a deduction of 27%, 22% and 17% with respect to when no SC is in use, for the proposed NMPC SQP solver with respectively full, partial and no prior knowledge of the future trip. The RBC and FT-LMPC both show 12% deduction.

Choosing the proper prediction and control horizon is a trade-off between performance and computational time. As the length of the prediction horizon is associated with the number of constraints in the NLP, the turnaround time increases by increasing N_p. However, the performance may increase or decrease depending on the quality of predicted power demand. As a longer prediction horizon conveys more knowledge of the future trip, it might be expected for the performance to improve by increasing N_p. Although this is shown to be true in the case of a perfect power prediction (P-NMPC), it is not always the case, as extending the prediction horizon adds the uncertainty associated with the extended measured disturbance (i.e. power demand) to the system. Fig. 5.7 and Fig. 5.8 show the performance of the system using a P-NMPC and FT-NMPC strategy respectively. It is noticed that although increasing the prediction horizon improves the performance using a P-NMPC strategy, the inverse correlation is observed when applying a FT-NMPC system. The reason for this lies in the fact that the assumption of a constant power demand over the prediction horizon becomes more and more inaccurate when a longer prediction horizon is used. Fig. 5.9 shows the performance of the system with respect to the length of the prediction horizon with a TP-NMPC employed. It is observed that while using a TP-NMPC strategy, increasing the prediction horizon improves the performance upto $N_p = 10$, after which the performance drops due to increased uncertainty. It might be worth reminding that through out this work, the time step has been set constant and equal to $T_c = 1s$. Fig. 5.10 shows the turn-around time associated with the length of the prediction horizon. The correlation is shown to be linear.

The control horizon length defines the size of the optimization problem and therefore has a great impact on both computational time and performance. By increasing the length of the control horizon, the number of optimization variables and constraints, consequently increase. While the correlation between the prediction horizon length and turnaround time is shown to be linear in Fig. 5.10, extending the control horizon length increases the turnaround time in the form of a second order polynomial as shown in Fig. 5.11. Also, as increasing N_c increases the size of the optimization problem, the resolution of the solution improves and more accurate results are obtained. While the improvement in the performance is shown to be negligible for the FT-NMPC strategy, it is shown to have a major impact on the performance of the TP-NMPC system as shown in Fig. 5.12.

As the upper bound for the turn-around time has been set to one millisecond, the

NMPC controller has been tuned for maximum performance with respect to N_p, N_c and K_{max}, to operate within the allowable time frame. The TP-NMPC shows to have optimal performance within the turn-around time of 1 millisecond while $N_p = 10$, $N_c = 10$ and $K_{max} = 3$. However, the maximum performance of the FT-NMPC is obtained when setting $N_p = N_c = 2$. The HIL results confirm that the proposed SQP solver enables real-time implementation of the NMPC method for the EMC of a SC/battery HESS.

5.4 Stochastic NMPC Results

S-NMPC could be used with no on-board technology for future trip prediction in hand. Comparative results of FT-NMPC, P-NMPC and S-NMPC on the model demonstrated in Fig. 3.1 are presented in this section. The NLPs obtained from the optimal control problem have been solved using MATLAB's *fmincon* function with an interior-point solver. In order to allow real-time implementation, the solver has been limited to perform a small number of iterations per sampling time. The length of the prediction horizon (N_p), control horizon (N_c), number of scenarios (N_S) and maximum number of iterations (K_{max}) should be chosen properly to maintain a fair trade-off between performance and computational time.

The transition probabilities of the TPM matrix have been estimated offline from a number of urban driving cycles, consisting of standard driving cycles (FTP75 and US06) and several real world trips. For the real world data, the publicly available dataset provided by Chargecar have been used [82]. 100 kilometers of real world data have been used to obtain the transition probabilities of the TPM matrix. The real world data belongs to a verity of routes from all differnt locations in North America. In order to examine the potential of the S-NMPC strategy, MIL simulations have been demonstrated for a different driving cycle from what the TPM matrix was generated with. The driver has been tuned to follow the desired reference speed.

The expected value of perfect information (EVPI) is a parameter in stochastic programming that measures the difference between the stochastic solution (SS) and the mean value of all deterministic solutions for each one of the expected scenarios; also known as the wait and see solution (WSS), i.e. EVPI=SS-WSS [83]. The EVPI demonstrates the cost of uncertainties in our problem. In Fig. 5.13 the battery current-squared sum of the S-NMPC has been compared with that of a D-NMPC when each one of the expected scenarios is applied separately. Five scenarios have been employed and the prediction horizon and control horizon lengths have been set to be five seconds while the solver is allowed to perform a maximum of four iterations per sampling time. The figure shows significant decrease in the battery current-squared sum of the S-NMPC in comparison with solving a D-NMPC with each of the five scenarios.

Fig. 5.14 shows the state of charge of the SC during the simulation. Although, SOC_{SC} has been bounded to stay between 0.5 and 1, by using the proposed cost function, the SC is mostly empty. However, by using stochastic programming, as a variety of scenarios are introduced to the optimal control problem, the SC is managed to stay more distant from the lower bound in order to be prepared for different possible scenarios.

In Fig. 5.15, comparative results of the S-NMPC and two deterministic-NMPCs discussed in section 4.2.3 have been presented. The performance of the S-NMPC lies between the performance of the FT-NMPC and P-NMPC as expected. Although full-knowledge of the future trip is not available a priori, by using S-NMPC we are able to improve the performance significantly and bring the cost closer to that of the P-NMPC. Also, the performance of the S-NMPC when using 10 scenarios shows an improvement in comparison to the case where 5 scenarios have been adopted. The reason for this lies in the fact that by increasing the number of scenarios involved in the stochastic programming (N_S), more probable scenarios are taken into account and a better decision could be made by the controller. However, increasing the number of scenarios increases the size of the NLP problem and slows down the computational speed. Therefore, a proper investigation should be performed in order to come up with the best trade-off between performance and computational time. Also, due to limitation in computational time in S-NMPC, scenario reduction and aggregation strategies could be promising solutions in order to decrease the number of scenarios while preserving accuracy [84].

5.5 Chapter Summary

Test results for various controllers implemented as the EMC of SC/battery HESSs have been presented in this chapter. MIL simulations have been established for performance investigation and comparison. While prior knowledge of the future trip shows to improve the performance of the NMPC, the aforementioned controller outperforms the implemented RBC even with no prior trip information available. The performance of the NMPC also shows improvement over the LMPC as it does not include rounding errors due to linearization. HIL implementation has been conducted using a dSPACE setup for investigating the computational time of the NMPC. The NMPC controller has been tuned with respect to N_p, N_c and K_{max} for optimal performance within the allowable turn-around time in vehicle applications. While NMPCs are usually known to suit application with slow dynamics, the results demonstrate the NMPC capability for real-time implementation of a SC/battery

HESS. The S-NMPC, although only adopting knowledge of previous trips, demonstrates significantly improved performance compared to deterministic predictive controllers using no on-board technology. However, the increased size of the NLP can be challenging when considering real-time implementation. An increase in the prediction and control horizon as well as the number of scenarios adopted in the stochastic programming and the maximum number of iterations the solver is allowed to perform, all can result in an improved performance with the cost of increasing the computational time. HIL testing for discovering the optimal values of these parameters, inhibiting real-time implementation, is still in hand.

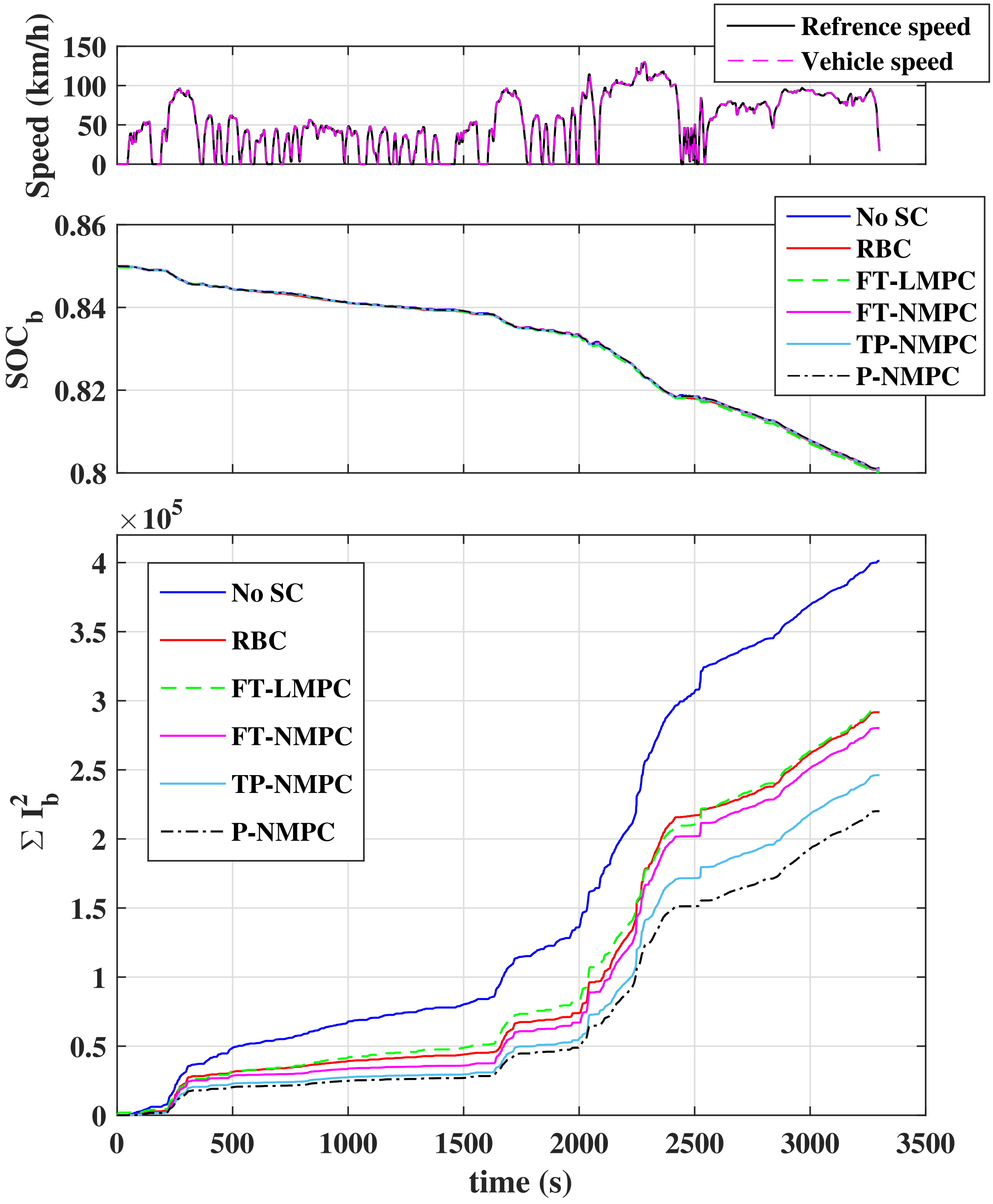

Figure 5.5: Comparison of the battery current squared sum of different EMCs over FTP75, US06 and HWFET driving cycles

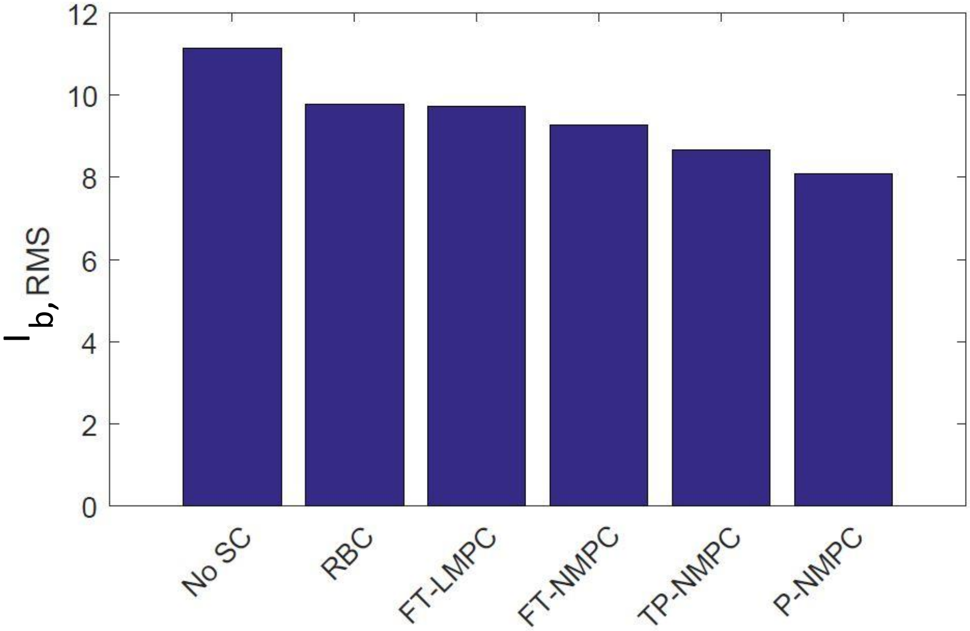

Figure 5.6: Comparison of the battery current RMS over FTP75, US06 and HWFET driving cycle

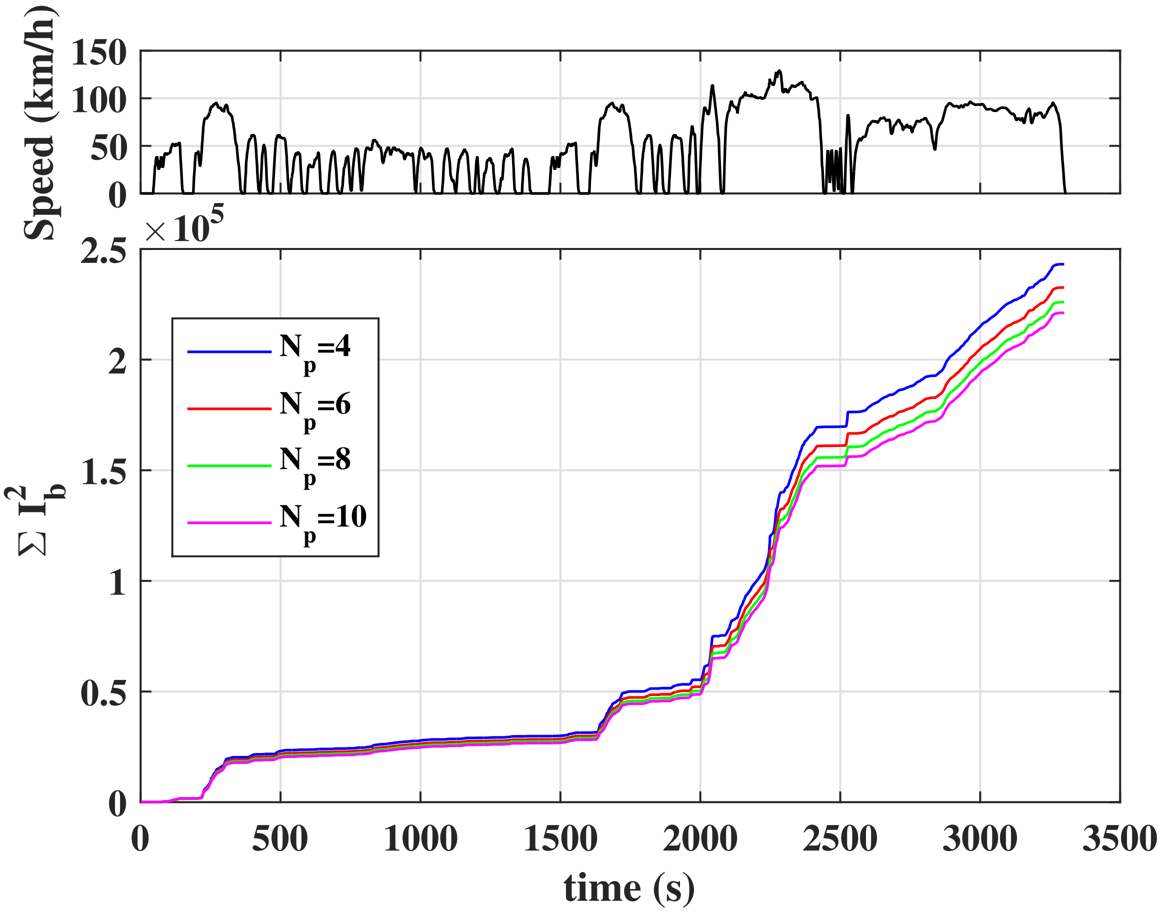

Figure 5.7: Comparison of the battery current-squared sum for different lengths of prediction horizons in P-NMPC, $N_c = 4$, $K_{max} = 3$

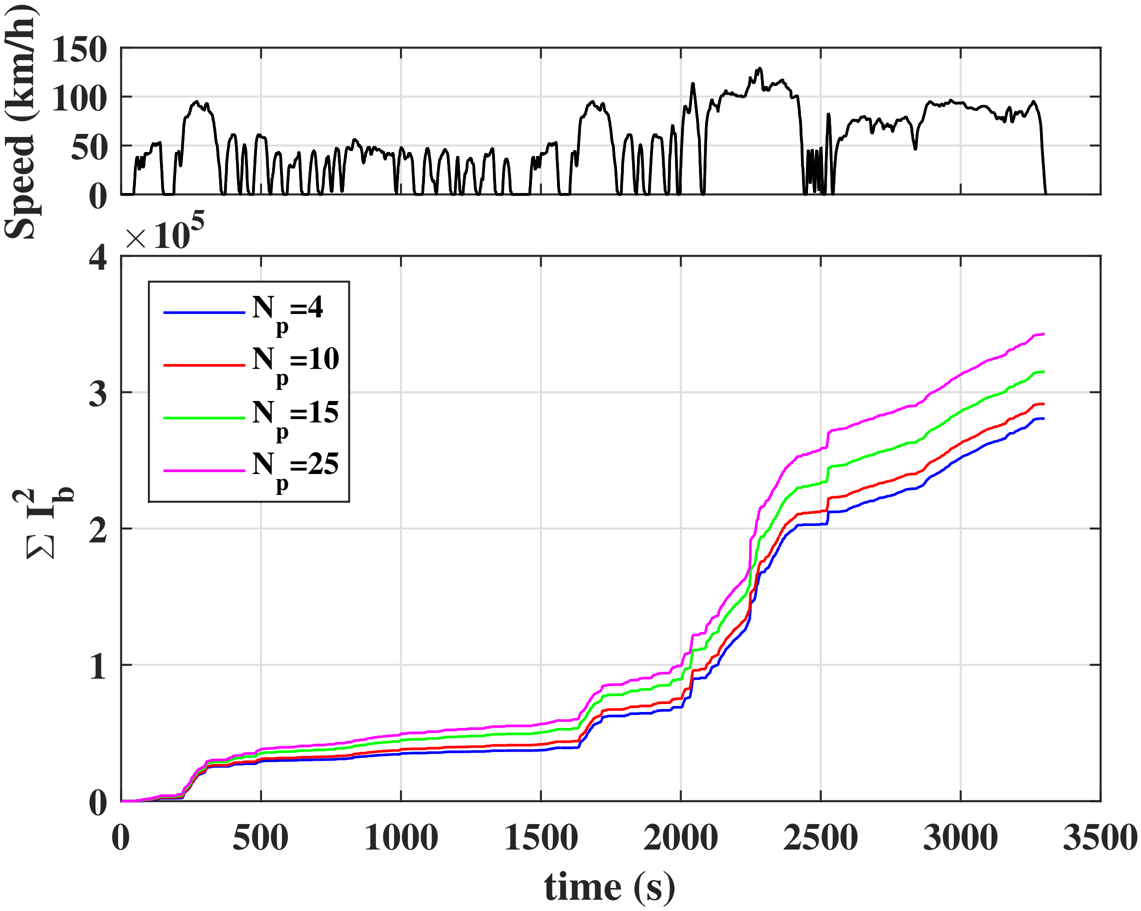

Figure 5.8: Comparison of the battery current-squared sum for different lengths of prediction horizons in FT-NMPC, $N_c = 4$, $K_{max} = 3$

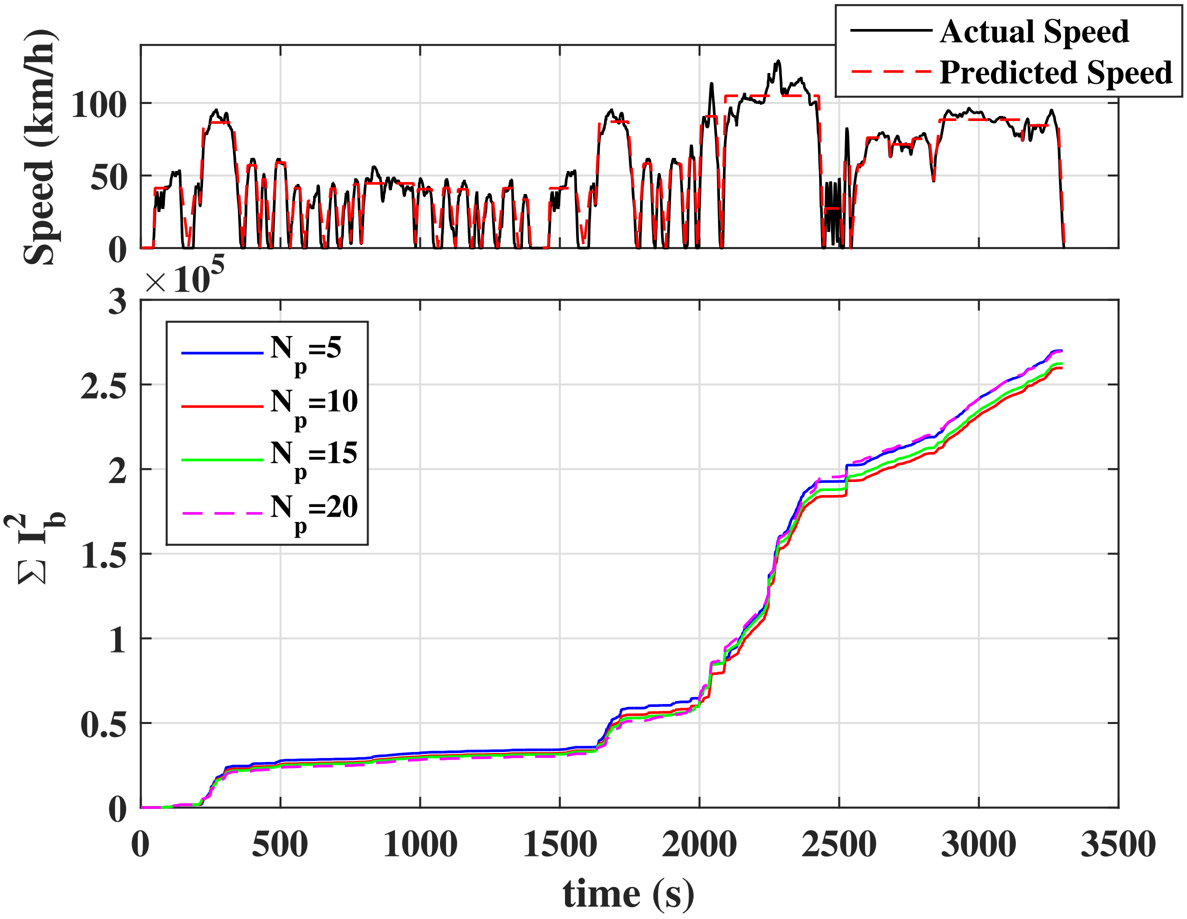

Figure 5.9: Comparison of the battery current sum for different lengths of prediction horizons in TP-NMPC, $N_c = 4$, $K_{max} = 3$

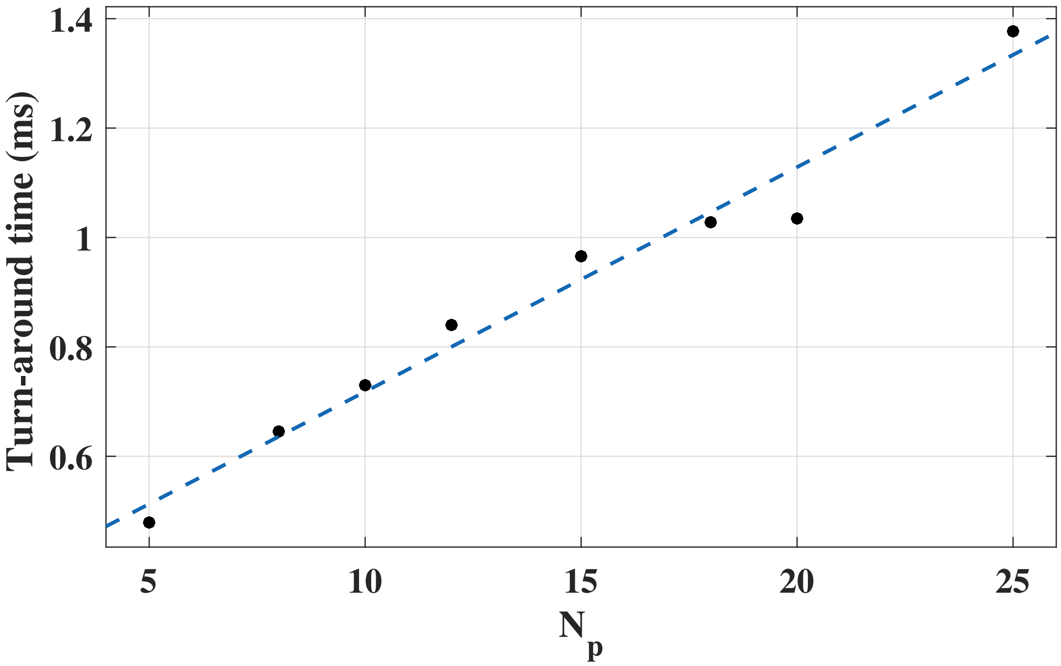

Figure 5.10: ECU maximum turn-around time for different lengths of prediction horizons, $N_c = 4$, $K_{max} = 3$

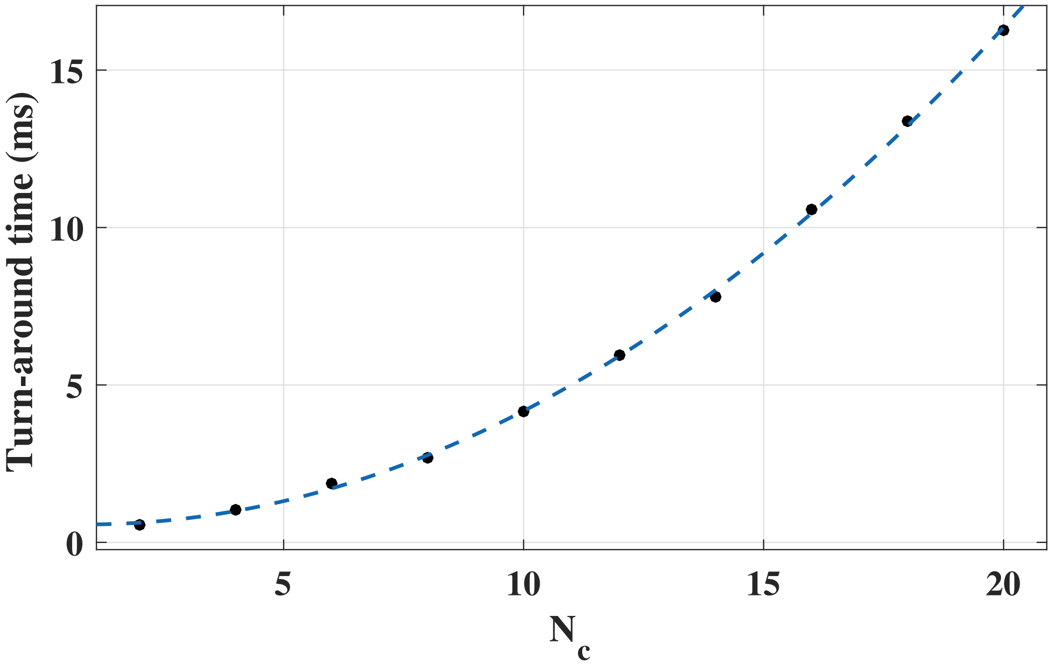

Figure 5.11: ECU maximum turn-around time for different lengths of control horizon, $N_p = 20$, $K_{max} = 3$

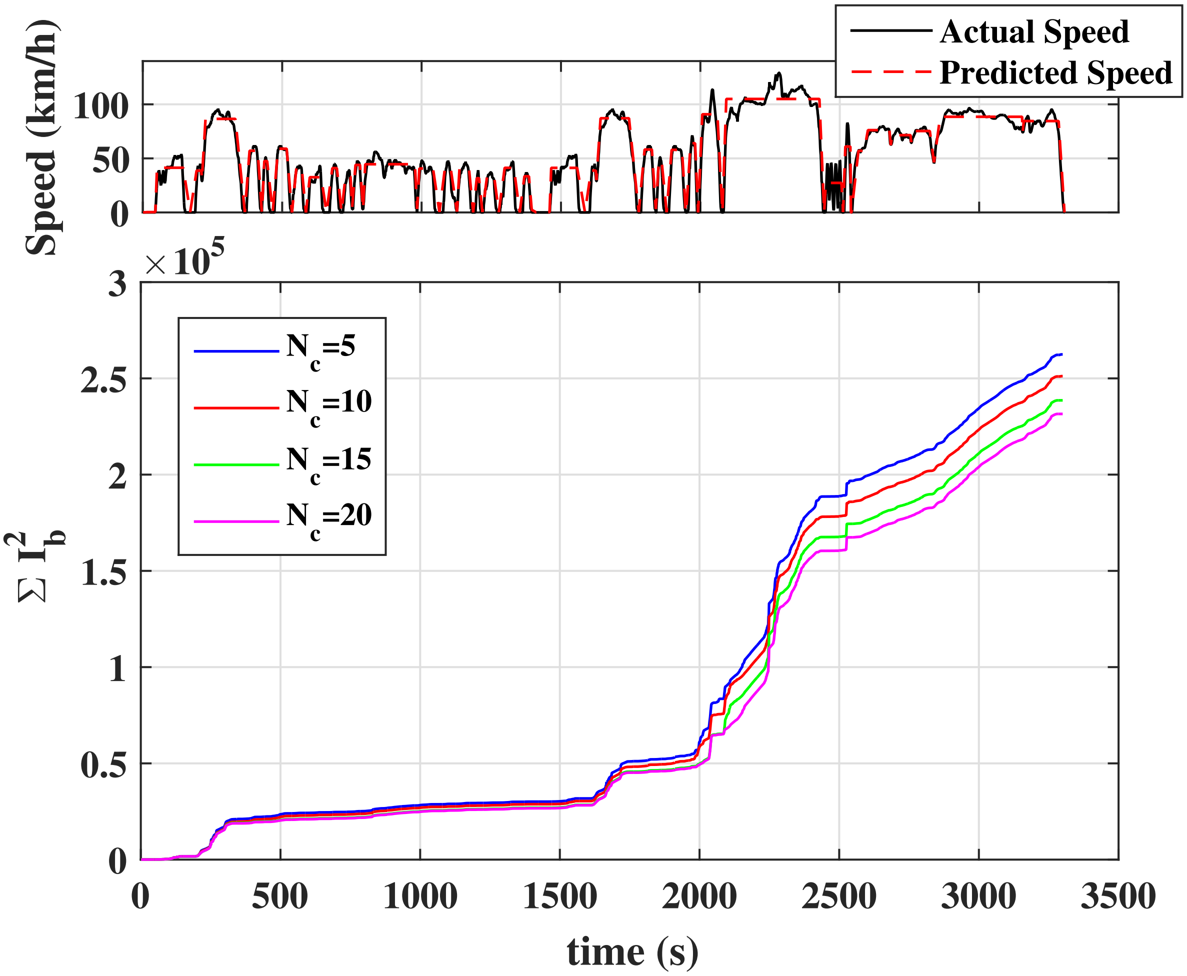

Figure 5.12: Comparison of the battery current-squared sum for different lengths of control horizons in TP-NMPC, $N_p = 20$, $K_{max} = 3$

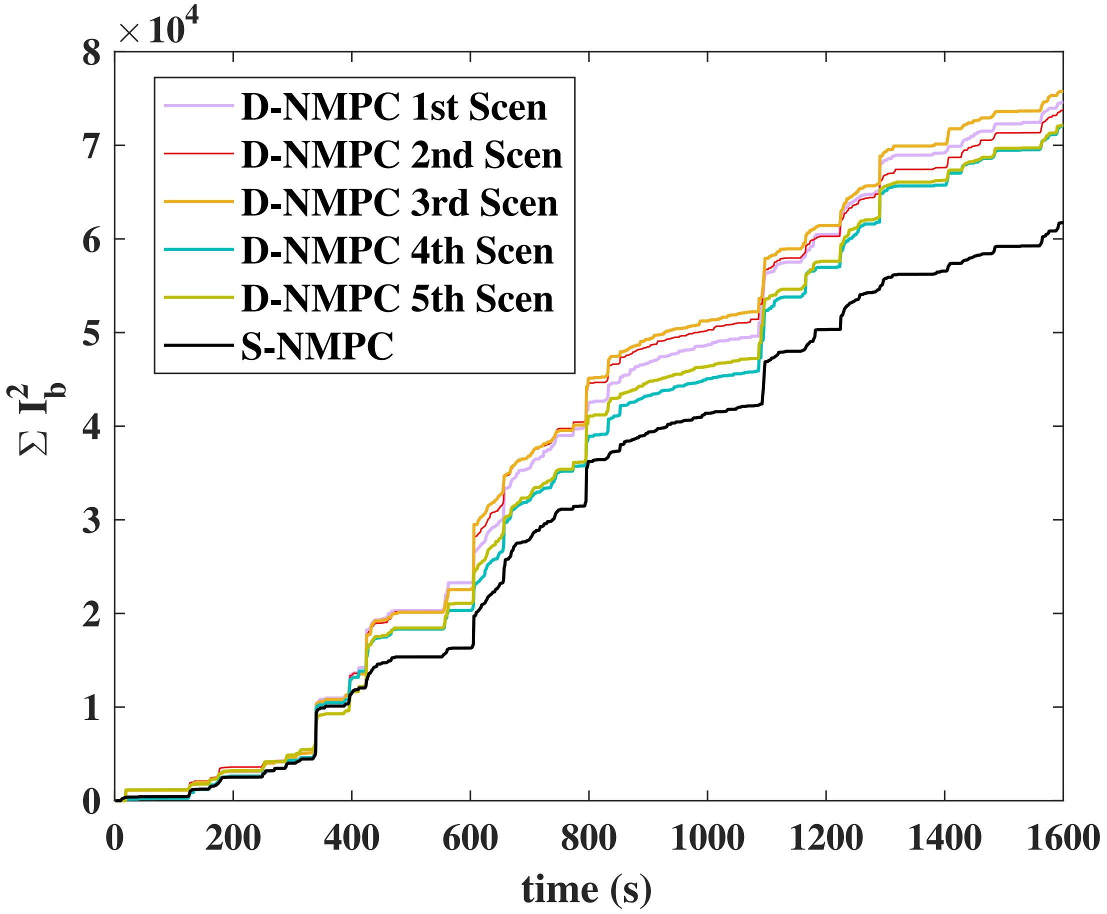

Figure 5.13: Cost comparison of the S-NMPC using 5 scenarios with the D-NMPC solved for the same 5 scenarios, $N_p = N_c = 5$.

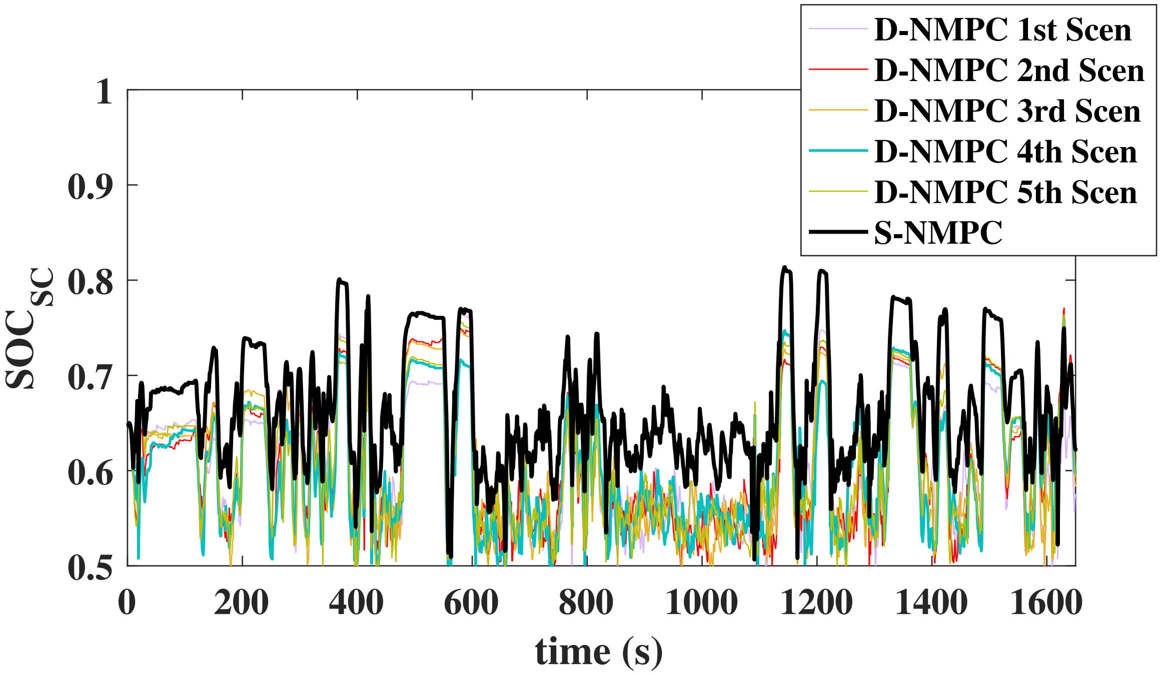

Figure 5.14: SOC_{SC} comparison of the S-NMPC using 5 scenarios with the D-NMPC solved for the same 5 scenarios, $N_p = N_c = 5$.

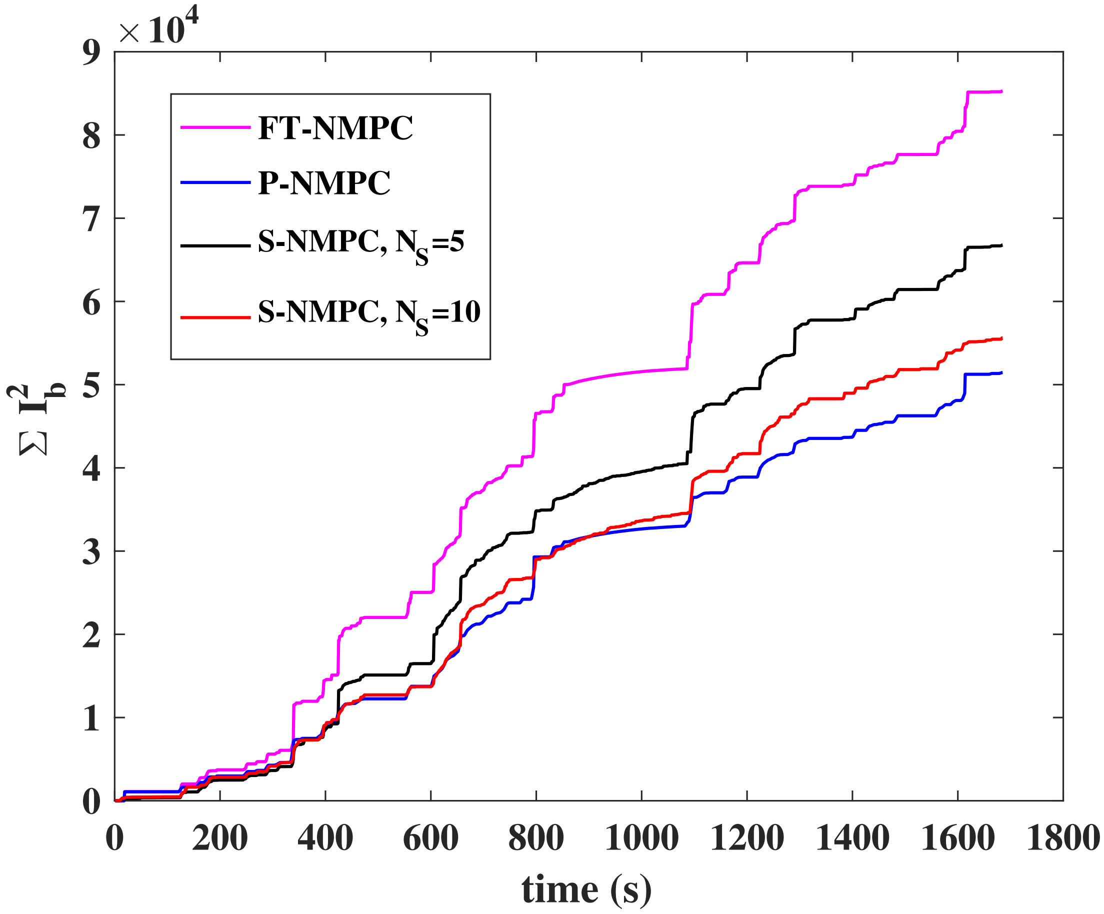

Figure 5.15: Comparison of the S-NMPC with the FT-NMPC and P-NMPC while $N_p = N_c = 5$.

Chapter 6

Conclusion and Future Work

The energy management of a SC/battery HESS has been presented using a number of deterministic and stochastic NMPC strategies for a Toyota Rav4EV model. MIL simulations, using a high-fidelity model of the system, have been established for performance investigation and comparison to state-of the-art EMCs in the literature. While prior knowledge of the future trip shows to improve the performance of the NMPC, the proposed NMPC demonstrates to outperform the implemented RBC and LMPC even with no prior trip information available. The NMPC performance, however, shows improvement by adopting methods that provide some level of prior trip prediction e.g. trip planning and Markov chain modeling.

HIL testing has been established using a dSPACE setup for investigating the computational time and evaluating real-time capability of the deterministic NMPC. During HIL testing, the controller runs on an independent ECU, similar to that on the actual vehicle, while the high-fidelity model runs on a powerful real-time simulator. In order to permit code generation for control embedding on the ECU, a SQP solver in MATLAB has been coded and used. The NMPC has been tuned with respect to the length of the control and prediction horizon as well as the allowed maximum number of iterations, for optimal performance within the allowable turn-around time in vehicle applications. While NMPCs are usually known to suit application with slow dynamics, the results demonstrate the determined NMPC capability for real-time implementation of a SC/battery HESS for the first time. HIL testing and real-time evaluation for the stochastic NMPC is still under study and has been left to future work.

6.1 Summary of Contributions

The major contributions of this research can be summarized as follows:

- This thesis has demonstrated an NMPC method for the EMC of a battery-SC HESS in EVs for the first time to be real-time implementable. Even though a battery-SC HESS is considered to have fast dynamics and NMPCs are known to be computationally slower than LMPCs, this thesis has presented an NMPC method for the first time to be real-time implementable for these fast systems. This has been done using HIL testing while the controller is implemented on a MicroAutoBox ECU. Upon carefully choosing the prediction horizon and control horizon length, as well as the maximum number of iterations, the turnaround time for each control update is shown to fall far bellow the necessary sampling time of 10 milliseconds, for real-time implementation purposes in vehicle control. While the NMPC maintains high computational speed, it shows to outperform its equivalent LMPC and the used RBC.

- In the case of using predictive controllers for battery-SC HESSs, the impact of the prediction horizon and control horizon length as well as the maximum number of allowable iterations on the turn-around time and performance have been studied in this research.

- Comparative results of LMPC and NMPC for the EMC of battery-SC HESSs is another contribution of this work. Although LMPCs also maintain the benefits accompanied by predictive control, it is noticed that the error caused by linearization in LMPCS has a significant effect on the optimization results; causing the NMPC to have a better performance.

- A novel scenario-based two-stage stochastic nonlinear model predictive control approach has been introduced as an EMC for SC/battery HESSs. The method has been inspired from multistage stochastic programming algorithms in the literature and demonstrated improvement in battery health when no knowledge of the future trip and/or no on-board technologies such as GPS and ITS are available. A new method for scenario generation has also been proposed.

- This research also demonstrates preliminary results from using a novel trip planning approach for future power demand predictions. The trip planning method uses on-board technologies such as GPS and ITS, coupled with information transfer to and from vehicles to form a rough, but accurate enough, prediction of the vehicle's future speed and power demand. The performance of the system using trip planning has been compared to the case where no knowledge of the future trip is available.

6.2 Future Work

The recommended future work for expanding this research is proposed below:

Stochastic NMPC HIL testing: Although the S-NMPC demonstrates significantly improved performance, compared to deterministic predictive controllers, the increased size of the NLP can be challenging for real-time implementation. An increase in the prediction and control horizon as well as the number of scenarios adopted in the stochastic programming and the maximum number of iterations the solver is allowed to perform, all can result in an improved performance with the cost of increasing the computational time. Investigation needs to be performed to discover the optimal values of these parameters inhibiting real-time implementation.

High-fidelity Model Improvement: The battery and SC high-fidelity model can be validated by running actual tests on the SC and battery pack and obtaining modeling parameters. This can be done by using battery test cyclers where the battery and SC are charged and discharged with various currents and in various temperatures.

Stability Analysis: MIL simulations demonstrate the NMPC capability of obtaining feasible solutions in almost all time-steps of the tested drive cycles. However, if the output of the controller is detected to be unfeasible, a control rule located after the predictive controller, forces the system to idle the SC and solely use the battery like any conventional EV. Further analysis are needed to theoretically analyze the stability of the proposed NMPC controllers.

Component-in-the-Loop testing: After MIL simulations and HIL testing are performed, the next step for controller evaluation is known as Component-in-the-Loop testing where the high-fidelity model of the system, embedded in the real-time simulator, is exchanged in whole or partially with the actual system of interest i.e. battery and/or SC.

References

[1] João P. Trovão, Paulo G. Pereirinha, Humberto M. Jorge, and Carlos Henggeler Antunes. A multi-level energy management system for multi-source electric vehicles – an integrated rule-based meta-heuristic approach. *Applied Energy*, 105:304 – 318, 2013.

[2] Nasa, noaa analyses reveal record-shattering global warm temperatures in 2015, January 2016.

[3] Sources of greenhouse gas emissions, 2014.

[4] Paul Dille, Matthew Duescher, Illah Nourbakhsh, Gregg Podnar, and Joshua Schapiro. Evaluating the urban electric vehicle. 2010.

[5] A. Styler, G. Podnar, P. Dille, M. Duescher, C. Bartley, and I. Nourbakhsh. Active management of a heterogeneous energy store for electric vehicles. In *Integrated and Sustainable Transportation System (FISTS), 2011 IEEE Forum on*, pages 20–25, June 2011.

[6] Stefano Ermon, Yexiang Xue, Carla Gomes, and Bart Selman. Learning policies for battery usage optimization in electric vehicles. *Machine Learning*, 92(1):177–194, 2013.

[7] John M Miller. Energy storage technology markets and application's: ultracapacitors in combination with lithium-ion. pages 16–22, 2007.

[8] M. Vajedi and N. L. Azad. Ecological adaptive cruise controller for plug-in hybrid electric vehicles using nonlinear model predictive control. *IEEE Transactions on Intelligent Transportation Systems*, 17(1):113–122, Jan 2016.

[9] A. Mozaffari, N. L. Azad, and J. K. Hedrick. A nonlinear model predictive controller with multiagent online optimizer for automotive cold-start hydrocarbon emission reduction. *IEEE Transactions on Vehicular Technology*, 65(6):4548–4563, June 2016.

[10] Mahyar Vajedi, Maryyeh Chehrehsaz, and Nasser L. Azad. Intelligent power management of plug–in hybrid electric vehicles, part i: real–time optimum soc trajectory builder. *International Journal of Electric and Hybrid Vehicles*, 6(1):46–67, 2014.

[11] Yuan Zou, Teng Liu, Dexing Liu, and Fengchun Sun. Reinforcement learning-based real-time energy management for a hybrid tracked vehicle. *Applied Energy*, 171:372 – 382, 2016.

[12] O. Laldin, M. Moshirvaziri, and O. Trescases. Predictive algorithm for optimizing power flow in hybrid ultracapacitor/battery storage systems for light electric vehicles. *IEEE Transactions on Power Electronics*, 28(8):3882–3895, Aug 2013.

[13] J. Cao and A. Emadi. A new battery/ultracapacitor hybrid energy storage system for electric, hybrid, and plug-in hybrid electric vehicles. *IEEE Transactions on Power Electronics*, 27(1):122–132, Jan 2012.

[14] S. M. Lukic, S. G. Wirasingha, F. Rodriguez, J. Cao, and A. Emadi. Power management of an ultracapacitor/battery hybrid energy storage system in an hev. pages 1–6, Sept 2006.

[15] M. Ortuzar, J. Moreno, and J. Dixon. Ultracapacitor-based auxiliary energy system for an electric vehicle: Implementation and evaluation. *IEEE Transactions on Industrial Electronics*, 54(4):2147–2156, Aug 2007.

[16] A. Di Napoli, F. Crescimbini, F. Giuli Capponi, and L. Solero. Control strategy for multiple input dc-dc power converters devoted to hybrid vehicle propulsion systems. 3:1036–1041 vol.3, 2002.

[17] A. Di Napoli, F. Crescimbini, S. Rodo, and L. Solero. Multiple input dc-dc power converter for fuel-cell powered hybrid vehicles. 4:1685–1690, 2002.

[18] R. Carter, A. Cruden, and P. J. Hall. Optimizing for efficiency or battery life in a battery/supercapacitor electric vehicle. *IEEE Transactions on Vehicular Technology*, 61(4):1526–1533, May 2012.

[19] A. A. Ferreira, J. A. Pomilio, G. Spiazzi, and L. de Araujo Silva. Energy management fuzzy logic supervisory for electric vehicle power supplies system. *IEEE Transactions on Power Electronics*, 23(1):107–115, Jan 2008.

[20] D. Tarsitano, L. Mazzola, S. Arrigoni, F. L. Mapelli, and F. Cheli. Energy management algorithms comparison for an electric bus with an hybrid energy storage system by

means of dynamic programming. In *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, pages 1–5, Sept 2014.

[21] C. Romaus, D. Wimmelbücker, K. S. Stille, and J. Böcker. Self-optimization energy management considering stochastic influences for a hybrid energy storage of an electric road vehicle. In *Electric Machines Drives Conference (IEMDC), 2013 IEEE International*, pages 67–74, May 2013.

[22] Ziyou Song, Heath Hofmann, Jianqiu Li, Xuebing Han, and Minggao Ouyang. Optimization for a hybrid energy storage system in electric vehicles using dynamic programing approach. *Applied Energy*, 139:151 – 162, 2015.

[23] Parisa Golchoubian and Nasser L Azad. An optimal energy management system for electric vehicles hybridized with supercapacitor. In *ASME 2015 Dynamic Systems and Control Conference*, pages V001T10A004–V001T10A004. American Society of Mechanical Engineers, 2015.

[24] M. E. Choi, J. S. Lee, and S. W. Seo. Real-time optimization for power management systems of a battery/supercapacitor hybrid energy storage system in electric vehicles. *IEEE Transactions on Vehicular Technology*, 63(8):3600–3611, Oct 2014.

[25] Ali Castaings, Walter Lhomme, Rochdi Trigui, and Alain Bouscayrol. Comparison of energy management strategies of a battery/supercapacitors system for electric vehicle under real-time constraints. *Applied Energy*, 163:190 – 200, 2016.

[26] Yi-Hsuan Hung and Chien-Hsun Wu. An integrated optimization approach for a hybrid energy system in electric vehicles. *Applied Energy*, 98:479 – 490, 2012.

[27] Alexander David Styler and Illah Reza Nourbakhsh. Real-time predictive optimization for energy management in a hybrid electric vehicle. pages 737–744, 2015.

[28] B. Hredzak, V. G. Agelidis, and M. Jang. A model predictive control system for a hybrid battery-ultracapacitor power source. *IEEE Transactions on Power Electronics*, 29(3):1469–1479, March 2014.

[29] Ziyou Song, Heath Hofmann, Jianqiu Li, Jun Hou, Xuebing Han, and Minggao Ouyang. Energy management strategies comparison for electric vehicles with hybrid energy storage system. *Applied Energy*, 134:321 – 331, 2014.

[30] X. Lin, M. Hu, S. Song, and Y. Yang. Battery-supercapacitor electric vehicles energy management using dp based predictive control algorithm. In *Computational Intelligence in Vehicles and Transportation Systems (CIVTS), 2014 IEEE Symposium on*, pages 30–35, Dec 2014.

[31] Richard T. Meyer, Raymond A. DeCarlo, and Steve Pekarek. Hybrid model predictive power management of a battery-supercapacitor electric vehicle. *Asian Journal of Control*, 18(1):150–165, 2016. asjc.1259.

[32] Hans Joachim Ferreau, Peter Ortner, Peter Langthaler, Luigi del Re, and Moritz Diehl. Predictive control of a real-world diesel engine using an extended online active set strategy. *Annual Reviews in Control*, 31(2):293 – 301, 2007.

[33] P. Golchoubian and N. L. Azad. Real-time nonlinear model predictive control of a battery-supercapacitor hybrid energy storage systems in electric vehicles. *IEEE Transactions on Vehicular Technology*, submitted for publication.

[34] K.I. Kouramas, C. Panos, N.P. Faísca, and E.N. Pistikopoulos. An algorithm for robust explicit/multi-parametric model predictive control. *Automatica*, 49(2):381 – 389, 2013.

[35] Alessandro Alessio and Alberto Bemporad. A survey on explicit model predictive control. In *Nonlinear model predictive control*, pages 345–369. Springer, 2009.

[36] Kaijiang Yu, Masakazu Mukai, and Taketoshi Kawabe. A battery management system using nonlinear model predictive control for a hybrid electric vehicle. *IFAC Proceedings Volumes*, 46(21):301 – 306, 2013.

[37] Payman Shakouri, Andrzej Ordys, and Mohamad R. Askari. Adaptive cruise control with stop&amp;go function using the state-dependent nonlinear model predictive control approach. {*ISA*} *Transactions*, 51(5):622 – 631, 2012.

[38] T. Ohtsuka. Continuation/gmres method for fast algorithm of nonlinear receding horizon control. In *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, volume 1, pages 766–771 vol.1, 2000.

[39] Moritz Diehl, H.Georg Bock, Johannes P. Schlöder, Rolf Findeisen, Zoltan Nagy, and Frank Allgöwer. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Process Control*, 12(4):577 – 585, 2002.

[40] A. Domahidi, A. U. Zgraggen, M. N. Zeilinger, M. Morari, and C. N. Jones. Efficient interior point methods for multistage problems arising in receding horizon control. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 668–674, Dec 2012.

[41] Alberto Bemporad and Manfred Morari. *Robust model predictive control: A survey*, pages 207–226. Springer London, London, 1999.

[42] Z. Q. Zheng and M. Morari. Robust stability of constrained model predictive control. pages 379–383, June 1993.

[43] Saša V. Raković, Basil Kouvaritakis, and Mark Cannon. Equi-normalization and exact scaling dynamics in homothetic tube model predictive control. *Systems & Control Letters*, 62(2):209 – 217, 2013.

[44] Giulio Ripaccioli, Daniele Bernardini, Stefano Di Cairano, Alberto Bemporad, and IV Kolmanovsky. A stochastic model predictive control approach for series hybrid electric vehicle power management. In *American Control Conference (ACC)*, volume 2010, pages 5844–5849, 2010.

[45] Ashwin Carvalho, Yiqi Gao, Stéphanie Lefevre, and Francesco Borrelli. Stochastic predictive control of autonomous vehicles in uncertain environments. 2014.

[46] A. Mesbah, S. Streif, R. Findeisen, and R. D. Braatz. Stochastic nonlinear model predictive control with probabilistic constraints. pages 2413–2419, June 2014.

[47] Ali Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine, Accepted*, 2016.

[48] A. Prekopa. *Stochastic Programming.* Boston, 1995.

[49] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczynski. Lectures on stochastic programming, volume 9 of mps/siam series on optimization. *Philadelphia, PA: SIAM. Modeling and theory*, 2009.

[50] D. Bernardini and A. Bemporad. Scenario-based model predictive control of stochastic constrained linear systems. pages 6333–6338, Dec 2009.

[51] Ivo Batina. *Model predictive control for stochastic systems by randomized algorithms.* Technische Universiteit Eindhoven, 2004.

[52] Marco C. Campi, Simone Garatti, and Maria Prandini. The scenario approach for systems and control design. *Annual Reviews in Control*, 33(2):149 – 157, 2009.

[53] Giuseppe C Calafiore, Marco C Campi, et al. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.

[54] Holger Heitsch and Werner Römisch. Scenario reduction algorithms in stochastic programming. *Computational Optimization and Applications*, 24(2):187–206, 2003.

[55] M. C. Campi and S. Garatti. A sampling-and-discarding approach to chance-constrained optimization: Feasibility and optimality. *Journal of Optimization Theory and Applications*, 148(2):257–280, 2011.

[56] X. Zhang, S. Grammatico, G. Schildbach, P. Goulart, and J. Lygeros. On the sample size of randomized mpc for chance-constrained systems with application to building climate control. pages 478–483, June 2014.

[57] Xiaojing Zhang, Sergio Grammatico, Georg Schildbach, Paul Goulart, and John Lygeros. On the sample size of random convex programs with structured dependence on the uncertainty. *Automatica*, 60:182 – 188, 2015.

[58] M. Batra, J. McPhee, and N. L. Azad. Parameter identification for a longitudinal dynamics model based on road tests of an electric vehicle. In *ASME International Design Engineering Technical Conference and Computers and Information in Engineering Conference*, August 2016.

[59] Richard T. Meyer. *Modeling and control of a fuel cell-battery hybrid vehicle*. PhD thesis, 2012. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2016-03-11.

[60] Maxwell: K2 2.7v series.

[61] Supercapacitor: Implement generic supercapacitor model.

[62] T. van Keulen, B. de Jager, D. Foster, and M. Steinbuch. Velocity trajectory optimization in hybrid electric trucks. pages 5074–5079, June 2010.

[63] S. D. Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky. Stochastic mpc with learning for driver-predictive vehicle control and its application to hev energy management. *IEEE Transactions on Control Systems Technology*, 22(3):1018–1031, May 2014.

[64] M. Bichi, G. Ripaccioli, S. Di Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky. Stochastic model predictive control with driver behavior learning for improved powertrain control. In *49th IEEE Conference on Decision and Control (CDC)*, pages 6077–6082, Dec 2010.

[65] Moritz Diehl, Hans Georg Bock, Holger Diedam, and P-B Wieber. Fast direct multiple shooting algorithms for optimal robot control. In *Fast motions in biomechanics and robotics*, pages 65–93. Springer, 2006.

[66] John R. Birge and François Louveaux. *Introduction to Stochastic Programming*. New York: Springer-Verlag, 2011.

[67] M Mahootchi, K Ponnambalam, and HR Tizhoosh. Comparison of risk-based optimization models for reservoir management. *Canadian Journal of Civil Engineering*, 37(1):112–124, 2010.

[68] Constrained nonlinear optimization algorithms:sequential quadratic programming (sqp).

[69] Michael JD Powell. A fast algorithm for nonlinearly constrained optimization calculations. In *Numerical analysis*, pages 144–157. Springer, 1978.

[70] Stanford business software inc. distribution of sol/ucsd optimization software.

[71] Iwr-simulation and optimization, muscod-ii.

[72] Abebe Geletu. Quadratic programming problems-a review on algorithms and applications (active-set and interior point methods).

[73] Model predictive control toolbox: Design and simulate model predictive controllers.

[74] Donald E Kirk. *Optimal control theory: an introduction*. Courier Corporation, 2012.

[75] Mahyar Vajedi. *Real-Time Optimal Control of a Plug-in Hybrid Electric Vehicle Using Trip Information*. PhD thesis, 2016.

[76] Matlab function: Include matlab code in models that generate embeddable c code.

[77] M. Vukov, W. Van Loock, B. Houska, H. J. Ferreau, J. Swevers, and M. Diehl. Experimental validation of nonlinear mpc on an overhead crane using automatic code generation. In *2012 American Control Conference (ACC)*, pages 6264–6269, June 2012.

[78] Y. Wang and S. Boyd. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2):267–278, March 2010.

[79] Scott B. Peterson, Jay Apt, and J.F. Whitacre. Lithium-ion battery cell degradation resulting from realistic vehicle and vehicle-to-grid utilization. *Journal of Power Sources*, 195(8):2385 – 2392, 2010.

[80] E. Vinot and R. Trigui. Optimal energy management of {HEVs} with hybrid storage system. *Energy Conversion and Management*, 76:437 – 452, 2013.

[81] Noshin Omar, Mohamed Abdel Monem, Yousef Firouz, Justin Salminen, Jelle Smekens, Omar Hegazy, Hamid Gaulous, Grietus Mulder, Peter Van den Bossche, Thierry Coosemans, and Joeri Van Mierlo. Lithium iron phosphate based battery – assessment of the aging parameters and development of cycle life model. *Applied Energy*, 113:1575 – 1585, 2014.

[82] Charge car: Driving data.

[83] Parisa Golchoubian, Nasser L. Azad, and Kumaraswamy Ponnambalam. Stochastic nonlinear model predictive control of battery-supercapacitor hybrid energy storage systems in electric vehicles. In *American Control Conferance (ACC)*, submitted for publication.

[84] Alireza Abbasi, Ghahraman Solookinejad, Mehdi Tadayon Fard, and Ali-Reza Zare. Consideration effect of uncertainty in the reliability indices of power systems using a scenario-based approach. *Journal of Intelligent & Fuzzy Systems*, 28(1):291 – 299, 2015.

[85] H. Borhan, A. Vahidi, A. M. Phillips, M. L. Kuang, I. V. Kolmanovsky, and S. Di Cairano. Mpc-based energy management of a power-split hybrid electric vehicle. *IEEE Transactions on Control Systems Technology*, 20(3):593–603, May 2012.