Advances in Industrial Control

Amir Taghavipour
Mahyar Vajedi
Nasser L. Azad

Intelligent Control of Connected Plug-in Hybrid Electric Vehicles

AIC
Springer

Advances in Industrial Control

Advances in Industrial Control is a series of monographs and contributed titles focusing on the applications of advanced and novel control methods within applied settings. This series has worldwide distribution to engineers, researchers and libraries.

The series promotes the exchange of information between academia and industry, to which end the books all demonstrate some theoretical aspect of an advanced or new control method and show how it can be applied either in a pilot plant or in some real industrial situation. The books are distinguished by the combination of the type of theory used and the type of application exemplified. Note that "industrial" here has a very broad interpretation; it applies not merely to the processes employed in industrial plants but to systems such as avionics and automotive brakes and drivetrain. This series complements the theoretical and more mathematical approach of Communications and Control Engineering.

Indexed by SCOPUS and Engineering Index.

Amir Taghavipour · Mahyar Vajedi
Nasser L. Azad

Intelligent Control of Connected Plug-in Hybrid Electric Vehicles

Springer

Amir Taghavipour
Mechanical Engineering Department
K. N. Toosi University of Technology
Tehran, Iran

Nasser L. Azad
Department of Systems Design Engineering
University of Waterloo
Waterloo, ON, Canada

Mahyar Vajedi
Department of Systems Design Engineering
University of Waterloo
Waterloo, ON, Canada

ISSN 1430-9491 ISSN 2193-1577 (electronic)
Advances in Industrial Control
ISBN 978-3-030-00313-5 ISBN 978-3-030-00314-2 (eBook)
https://doi.org/10.1007/978-3-030-00314-2

Library of Congress Control Number: 2018953710

Mathematics Subject Classification (2010): 49J20, 49J24, 49K20, 49L20, 49M37, 49N90, 49Q12

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Series Editor's Foreword

The series *Advances in Industrial Control* reports and encourages technology transfer in control engineering. The rapid development of new control theory and technology has an impact on all areas of control engineering and engineering applications. There are new control theories, control design methods, actuators, sensors, computing methods, philosophies, and of course many new applications. This would provide sufficient justification for a specialized monograph series but there is a more important reason for its development. The development of control theory needs to be stimulated and driven by the needs and challenges of applications. A focus on applications is also essential if the related subject of control design is to receive sufficient attention. The series provides an opportunity for researchers to present an extended exposition of new work on industrial control, raising awareness of the substantial benefits that can accrue and describing the problems that may arise.

This text covers the exciting growth area of hybrid electric vehicles. The importance of the subject is clear from the high profile the topic receives in the daily news. The primary objective of the control systems described is to minimize the energy cost both in the form of electricity from the grid and the fuel used. A hierarchical solution to the problem is described where the upper level involves trip planning and the lower level involves vehicle speed control.

The development of suitable control philosophies depends upon a thorough understanding of the physical problem and the control-oriented design models that are required. The development and validation of such models, that have to be sufficiently accurate to characterize the system, is described. The authors describe the design methods utilized, including cluster-based optimisation and nonlinear model predictive control. They also cover topics that are often neglected, namely hardware-in-the-loop testing and real-time implementation. After introducing the subject, a high-fidelity simulation model for a Toyota hybrid powertrain is described.

Nonlinear model predictive control is one of the most popular techniques utilized in research studies for powertrain controls in the automotive industry. In fact, the first application in a production engine has been announced recently.

Consequently, the background material provided in this text and the application described are both important to engineers in the field. The use of predictive control methods for the energy management system is in line with the industry trends. Another technique covered is sliding mode control, which is another topic generating great interest in applications. All of the application problems considered are of great importance, like that of emissions control, which is of course a topic that has involved major controversies in recent years.

An explicit model predictive control approach is taken with the aim of reducing the computational time. The complexity of the solution is determined by the number of regions that form the explicit solution. Standard model predictive control is limited because of its computational requirement when used in applications like automotive. In the explicit approach to model predictive control (MPC), most of the calculations are performed offline using a multiparametric programming approach. Important practical topics are discussed such as the reduction in the number of regions, which affect the computational load. The text demonstrates the benefits for implementation. Optimization is clearly important for several aspects of this problem and techniques such as dynamic programming are also described.

A discussion of hardware-in-the-loop testing procedures is provided in the appendix, which illustrates how close to the real application these authors came. The control solutions described for the different aspects of the problem are important in automotive systems but they should also be of wider interest to the control community for other application sectors. This is therefore a welcome addition to the *Advances in Industrial Control* monograph series.

Glasgow, UK
May 2018

Michael J. Grimble

Preface

This book presents the development of real-time intelligent control system which involves control-oriented modeling, controller design, and performance evaluation. The primary objective of the control systems is to minimize the total energy cost, including both the electricity derived from the grid and fuel. The upper level system is trip planning using an algorithm designed to take advantage of the previewed trip information to optimize the battery depletion profiles. The Eco-Cruise controller adjusts the speed considering the upcoming trip. Finally, the route-based EMS optimally distributes propulsion power between the batteries and engine. The real-time implementation capability of the intelligent controller is achieved through efficient control-oriented models and fast optimization techniques. Control-oriented Models are sufficiently simple and fast for real-time application and are accurate enough to characterize the system. These models have been validated using a high-fidelity model, which is more complex and considers more system details. Cluster-based optimization, nonlinear model predictive controller, and equivalent consumption minimization strategy are developed to solve the optimal control problem. Model-in-the-loop and hardware-in-the-loop testing are critical steps in control validation and in ensuring real-time implementation capability. The results demonstrate that the computational time for energy-optimal strategies is less than the target sampling time, and they can be implemented in real time.

Waterloo, ON, Canada
April 2018

Amir Taghavipour
Mahyar Vajedi
Nasser L. Azad

Acknowledgements

We dedicate this book to all our loved ones and all of those who have helped us in preparing this piece of work. We specially thank Prof. John McPhee at University of Waterloo and Toyota Technical Center, Ann Arbor, MI for his great contributions to this book.

Contents

Chapter 1
Introduction

1.1 Background

Energy and environmental issues are among the most serious concerns facing today's society, with the automotive industry figuring prominently in the global dialogue. At its current trajectory, the transportation sector is expected to become the largest greenhouse gas producer over the next 50 years [1], and will continue to rank among the world's leading energy consumers. This situation has spurred significant investment in the research and development of fuel efficient, sustainable transportation systems that will minimize, and eventually replace, non-renewable, fossil-based energy sources with clean, renewable ones.

Electric vehicles (EV) consume electric energy from the grid, which can be provided from renewable resources. In addition, EVs demonstrate outstanding energy efficiency, at roughly 70%, in comparison with conventional Internal Combustion Engine (ICE) vehicles, which have peaked at just 30% efficiency [2]. Therefore, EVs are increasingly considered viable long-term solutions to the sustainable transportation challenge. However, broad take-up of current EV technologies has been hampered by their limited operating range (about 150 km) and high initial costs.

PHEVs are promising options for near-term improvements to transportation sustainability. Utilizing two sources of energy, batteries (to power an electric motor) and gasoline (to power an ICE), allows them to achieve much higher energy efficiencies than conventional ICE vehicles, without sacrificing performance. Furthermore, with an ability to store electric energy directly from the grid, PHEVs offer the combined advantages of both electric and Hybrid Electric Vehicles (HEVs). That is, they can operate in full electric mode in urban areas (zero emissions, zero fuel consumption) and provide the extended range and efficiency typical of HEVs during highway driving (low emissions, low fuel consumption).

The potential to maximize energy efficiency and minimize environmental impacts through vehicle electrification has captured the attention of most automotive manufacturers. In response to energy and environmental concerns, governments across the world have established strict vehicle standards, such as the American Corporate

A. Taghavipour et al., *Intelligent Control of Connected Plug-in Hybrid Electric Vehicles*,
Advances in Industrial Control, https://doi.org/10.1007/978-3-030-00314-2_1

Average Fuel Economy (CAFE), which require automotive companies to dramatically reduce fleet emissions and improve efficiencies on a very tight timeline. Most, if not all, major automotive manufacturers have made major investments in HEV, PHEV, and/or EV development, not only to meet future market demands, but also to maintain their annual average fuel economy within the range allowable by government standards.

1.2 Motivation and Challenges

Due to rising fuel cost and emission concerns, PHEVs are increasingly considered a viable alternative to conventional vehicle technologies. Although these vehicles demonstrate significantly improved vehicle performance, high initial and maintenance costs have proved an impractical trade-off for most consumers. According to the report by National Academy of Sciences [1], EVs and PHEVs, which were once expected to dominate the automotive industry by 2020, will likely capture only 5% of the market in the next decade.

To increase demand for PHEVs, long-term operation cost issues must be resolved. In other words, vehicle systems should be enhanced to improve fuel efficiency, which in turn, will provide long-term cost advantages over conventional vehicles. One approach is to improve total energy cost through advanced powertrain control systems, in particular the optimal trip planning module, Energy Management System (EMS), and ecological cruise controller. Trip planning modules optimize electrical energy profiles based on long-range trip information for any driving scenario. EMS strategies compute the power distribution between two sources of energy (fuel and battery). Ecological cruise controllers adjust vehicle speeds to minimize total energy cost.

Although PHEVs offer improved efficiencies similar to conventional HEVs, they also present new challenges in energy-optimal powertrain control problem. It is difficult to find control solutions that can optimally distribute power demand between the two sources of energy. This difficulty largely stems from the fact that PHEVs differ from HEVs in two key ways: they can fully charge the battery by connecting to the grid and they can operate in charge depletion mode, reducing the battery SOC to the minimum permissible value. Furthermore, the ability to restore the electrical energy during regenerative braking makes ecological cruise control for PHEVs more complicated than that for a conventional vehicle. Therefore, approaches to the development of energy-optimal controllers for PHEVs must consider these differences in the dynamics of the system, initial conditions and constraints.

A global optimum solution for the PHEV powertrain control problem can only be addressed with advance knowledge of driving conditions. Access to these data enables optimal distribution of electrical energy throughout any trip. We can take advantage of recent advancements in GPS, GIS, Intelligent Vehicle Technologies, such as Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) Communica-

tion Systems, and radar sensor to predict future driving conditions and create more effective energy-optimal controllers.

In particular, control systems that optimize EMS and enable ecological cruise control are very complex, making real-time implementation a significant challenge. Several global optimization methods that aim to provide an optimum powertrain control strategy already exist; however, computationally they are extremely costly, particularly for long trips, and cannot be implemented in real time. This book will address these challenges by developing a novel energy-optimal controller for power-split PHEVs, with real-time implementation capability.

1.3 Objectives and Methods

The goal of this book was to design a real-time energy-optimal controller to minimize total energy cost, including both fuel and electrical energy taken from the grid. To enhance the performance of the controller, prediction of future driving conditions were incorporated. The energy-optimal controller has been designed for real-time implementation. To address the potential for practical implementation, the controller has been fine-tuned to a commercial PHEV architecture, the Toyota Prius Plug-in Hybrid, for testing and validation purposes. The devised energy-optimal controller consists of three main systems:

- Trip Planning module
- Route-based EMS
- Eco-Cruise controller

The real-time implementation capability of the energy-optimal controller is achieved through efficient control-oriented models and fast optimization techniques. Control-oriented models are sufficiently simple and fast for real-time implementation, and are accurate enough to characterize the system. These models have been validated using a high-fidelity model, which is more complex and considers more system details. The high-fidelity model of baseline PHEV has been developed in Prof. Nasser L. Azad and Prof. John McPhee's research group by Taghavipour [3] and Chehresaz [4] (Please see Sect. 3.5).

In each aforementioned energy-optimal system, a proper and fast optimization algorithm is developed based on the problem characteristics such as desired computational speed, number of variables and optimization parameters, and dynamics of the system. The Trip Planning module deals with huge number of variables and optimization parameters, especially during long trips. Therefore, a Real-time Cluster-based Optimization (RCO) algorithm is developed for this module that reduces the number of optimization parameters using clustering technique. Then, Dynamic Programming (DP) is applied to solve the simplified optimization problem. Route-based EMS is based on ECMS strategy, which is fast technique for solving EMS problems. This book improves EMS performance by incorporating trip information. Finally, the NMPC technique is utilized for solving the Eco-Cruise control problem. NMPC

can solve multi-objective control problems and can handle constraints on states and inputs. Therefore, it's a promising technique for the Eco-Cruise controller which optimizes both the total energy cost and driving safety while considering constraints on vehicle speed and powertrain variables.

This book evaluates the new energy-optimal controller using following strategies:

- Energy-optimal controller performance compared against:
  - Autonomie software rule-based controllers; and
  - Global optimum control methods: Dynamic Programming and Pontryagin's Minimum Principle (PMP).
- MIL simulations to evaluate the new controllers; and
- HIL tests to evaluate the real-time implementation capability of the energy-optimal controller.

1.4 Book Organization

This book is organized as follows: Chap. 2 reviews state-of-the art HEV and PHEV control strategies, including both EMS and speed controllers. This discussion also addresses current methods used to incorporate trip information within control strategies that aim to improve vehicle performance. Chapters 3, 4, 5 and 6 discuss the modeling approach, nonlinear and multi-parametric model predictive control and control-relevant parameter estimation to make the controller prediction more accurate. Chapters 7, 8, and 9 cover the steps in the design of the proposed Trip Planning, Route-based EMS, and Eco-Cruise controller. Finally, Chap. 10 discusses the conclusions and future work.

References

1. Unger, N., Bond, T., Wang, J., Koch, D., Menon, S., Shindell, D., Bauer, S.: Attribution of climate forcing to economic sectors. National Academy of Sciences of the United States of America (2010)
2. Reinhardt, W., Hadrovic, A.: Roadmap on ICT for energy efficiency. Technical Report (2010)
3. Taghavipour, A.: Real-time optimal energy management system for plug-in hybrid electric vehicles. Ph.D. thesis, University of Waterloo (2014)
4. Chehresaz, M.: Modeling and design optimization of plug-in hybrid electric vehicle powertrains. Master's thesis, University of Waterloo (2013)

Chapter 2
Related Work

This chapter reviews recent advances in HEV and PHEV research, spanning Trip Planning algorithms, EMS strategies, and cruise control techniques. It also investigates approaches to improve control system efficiency. The chapter concludes with a summary of the relevant literature and an overview of the significant contributions of the designed energy-optimal control system.

2.1 Trip Planning

The model-based and data-driven methods are the two most commonly utilized approaches used to forecast traffic conditions [1]. Imminent traffic speed is determined based on traffic flow theory when utilizing the model-based method. Alternatively, the data-driven method uses current and past traffic data without integrating a traffic model when used to anticipate future traffic speeds. Among the most popular data-driven methods are neural networks. To illustrate, Park [2, 3] modelled traffic conditions using an artificial neural network that was based on GPS and ITS trip information, to model traffic conditions, which then led to improvements in EMS performance and fuel efficiency.

Model-based driving condition forecasts are used by some researchers in the speed controller system to optimize fuel efficiency. Studies by Kuelen et al. [4–6] are some of the more recent examples. In one study, the researchers reached optimum future speed trajectory for an HEV based on trip information. To ensure that the vehicle maintained the ideal speed trajectory, the data was fed to a cruise controller. In a subsequent project [7], Kuelen et al. presented an optimized EMS controller for hybrid electric trucks based on trip information, where they computed optimal future speed trajectories using said trip information. By implementing an online EMS controller, optimum deceleration rate during regenerative braking was obtained to

A. Taghavipour et al., *Intelligent Control of Connected Plug-in Hybrid Electric Vehicles*,
Advances in Industrial Control, https://doi.org/10.1007/978-3-030-00314-2_2

achieve maximum energy recovery. The result was a demonstration of considerably reduced fuel consumption through the utilization of the proposed control approach.

The two-scale DP method to generate optimized energy management schemes for PHEVs was proposed by Gong et al. [8, 9]. The method involved calculating the power demand controlled by the trip information, followed by the optimization of the SOC profile based on the linear model of the battery. In order to merge trip modelling into EMS, utilization of the Gas-kinetic based models and neural networks were examined, leading to the conclusion that numerous limitations complicate the Gas-kinetic model, whereas improved trip modelling accuracy and more realistic real-time implementation characterized the neural networks model. Traffic data was applied in order to train the neural network and impending traffic conditions were used in the EMS of parallel PHEV. Gong et al. [9, 10] applied a statistical method in another PHEV EMS project in order attain traffic information, while explicitly utilizing a Markov chain model for velocity trajectory generation. Controller tuning and EMS strategy expansion was achieved by using the predicted velocity trajectory, resulting in improved vehicle performance.

In another study, Bin et al. [11] worked with an adjustable segment scheme to normalize trip segment lengths based on trip information, and to decrease the computational effort required for effective EMS. The (simulation) results were compared to earlier-referenced DP-results that provided SOC and power demand data calculated numerically, finding that the two-scale DP scheme is more computationally efficient.

Katsargyri et al. [12, 13] optimized the battery SOC profile of power-split PHEVs based on trip information. The PSAT software was used to track the optimum SOC and its rule-based controlled proposed a receding horizon approach in the segmentation procedure. A virtual route replaced a largely segmented original route. The virtual route consisted of a number of original route segments, plus an additional last segment assumed equivalent to the remaining sections. Optimum SOC of each segment was achieved using DP, while using the first segments reference point in the EMS strategy, resulting in a demonstration of the methods reduced computational cost compared to the prime solution for the original route.

2.2 HEV/PHEV Energy Management Strategies

To attain optimal fuel economy, an EMS strategy that optimizes energy flow between two power sources is an essential requisite. EMS schemes may be divided into two principal categories: (1) reactive; and (2) route-based strategies. Reactive EMS strategies can be characterized as utilizing present driving information in the controller scheme and can only find near optimal resolutions for potential problems. Rule-based controllers, Charge Depleting Charge Sustaining strategy (CDCS), Stochastic Dynamic Programming (SDP), and the ECMS strategy belong to this group.

Rule-based approaches typically utilize maps constructed from engineering expertise, or more formal methods, such as optimization [14]. These are characterized as high performing inflexible schemes for known drive cycle patterns that are not

optimized. In the CDCS strategy, the vehicle is thrusted principally by utilizing battery-generated energy in charge depleting (CD) mode, until a predefined level of SOC is attained. This is followed by the EMS controller switching to charge sustaining (CS) mode, which keeps the SOC at the predefined level.

The SDP method is an appealing EMS approach because it provides an ability to optimize system performance with respect to a probabilistic distribution of different drive cycles [15]. Bashash et al. [16] and Moura et al. [17, 18] utilized the SDP method to establish an appropriate EMS strategy for power-split PHEVs. Bashash predicted the PHEV's power demand based on stochastic data, while Moura used SDP to derive an optimal EMS strategy for a PHEV platform, strategically endeavoring to minimize both fuel and battery maintenance costs. His simulation results revealed that initially more electric power is expanded to rapidly drain the battery SOC during high battery resistance growth. As such, a combined approach is a more effectual strategy while the SOC is low.

A truly optimized EMS scheme must integrate look-ahead trip and driving conditions that comprises the complete drive cycle. Considerable research has been undertaken to reduce fuel consumption and vehicle emissions with the help of preview trip information [19–23]. Gonder et al. [24, 25] investigated the efficacy of using route-based control algorithms to improve the fuel economy of HEVs and demonstrated that fuel consumption is improved between 24% through the utilization of look-ahead control. While the development may appear immaterial, it does present an opportunity for substantial fuel savings. Based on estimated annual HEV fuel consumption in 2006, a 2–4% improvement in efficiency could potentially reduce annual fuel consumption by 6.5 million gallons in the United States alone [24, 25].

Taking advantage of trip information increases the complexity of the energy management problem and makes the real-time implementation of these control strategies very challenging in practice. Numerous prime control approaches for HEV and PHEV energy management schemes were examined in the literature [26], including: DP, PMP, Model Predictive Control (MPC), and Adaptive-ECMS (A-ECMS). In order to refine preceding EMS research directly related to the research presented in this book, the ensuing discussion focuses on the aforementioned optimal control techniques.

2.2.1 Dynamic Programming

DP is a global optimization scheme that resolves a vehicle's power distribution based on trip information. The method is unfavorable due to its computational expansiveness and frequent impracticality for real-time application, especially for complex problems. Lin et al. [27, 28] has found near-optimal rules in the EMS controller of parallel HEVs based on DP global minimum results. By employing this method, the researcher attained vast improvements in fuel efficiency, as well as ideal power distribution between the engine and the electric motor.

Gonder et al. [29] studied several PHEV EMS strategies during CD mode, including all-electric-range and blended strategy. These strategies were compared in terms

of travel distance and prior knowledge of the trip, resulting in demonstrating that the all-electric-range operation mode has all of the advantages of a full electric vehicle, but require electric components much higher in cost, such as a larger battery and electric motor. Alternatively, the blended strategy demonstrated the best fuel efficiency for long-distance travel. O'Keefe and Markel [30] showed that in optimal battery charge distribution strategies, SOC reaches the minimum level exactly when the trip terminates.

2.2.2 Pontryagin's Minimum Principle

The PMP technique finds the global optimal solution for control problems based on reducing the Hamiltonian. This technique can minimize the performance index in the presence of states and inputs constraints. Razavian et al. [31, 32] developed a real-time EMS strategy for a series HEV using the PMP technique. To tune its parameters, the controller only claims the cruise time and the amount of energy used during regenerative braking, as it works independently of speed trajectory. With this approach, the optimal control problem was reduced from the integral cost minimization to the instantaneous Hamiltonian minimization, enabling real-time implementation of the EMS strategy. In order to evaluate its performance, the controller was applied to a high-fidelity HEV model developed in MapleSim, resulting in a substantial enhancement in fuel economy.

Ebbesen et al. [33] considered battery life in the optimum EMS strategy of a parallel HEV using the PMP technique, claiming development of the first state of health model Li-ion battery. Due to their extreme complexity, electrochemical models cannot be implemented in real-time EMS schemes; as such, Ebbesen utilized a throughput-based capacity fade model in his research, assuming that under constant conditions, the batterys energy capacity is the same as the remaining number of cycles prior to its end-of-life.

2.2.3 Model Predictive Control

Due to its ability to handle constrained multi-input, multi-output optimal control problems, model predictive control is a promising method for exploiting the potentials of modern concepts and fulfilling automotive requirements, [34]. The application of MPC to hybrid vehicles has been examined previously. Wang [35] proposed a real-time control system for different hybrid architectures using the MPC concept. Kim et al. [36] utilized MPC to calculate an optimal torque split in the parallel HEV. Borhan et al. [37, 38] applied MPC to a power-split HEV, ignoring the dynamics of the powertrain against other faster dynamics for the model inside the controller. To assess performance, he implemented a number of different controllers, such as a nonlinear MPC, linearized MPC, and the rule-based controller strategy offered by

the PSAT software. Based on his results, the nonlinear MPC provided the greatest improvement to fuel economy in comparison to the other controllers examined.

Taghavipour et al. [39] applied MPC to a power-split PHEV and used DP as a benchmark for evaluating the EMS performance. In another study [40], the authors developed an EMS strategy based on MPC approach and applied it to a high-fidelity model of a power split PHEV in the MapleSim software in order to minimize fuel consumption.

2.2.4 Explicit Model Predictive Control

Classical model predictive control (MPC) requires on-line optimal solution of linear or quadratic programming problem at each sample time, using the current states. Such on-line computation is the main disadvantage that limits applications of MPC in general only for controlling slow processes [41]. For instance, the first generation of MPC algorithms was aimed at solving multi-variable constrained control problems typical to the oil and chemical industries [42]. However, computation speed is not the only limitation: the code implementing the solver might generate concerns due to software certification issues, a problem which is particularly acute in safety critical applications [43].

Explicit MPC (eMPC) techniques [44] can be used to synthesize the controller as a piecewise affine function. With this approach appropriately applied, the MPC can be implemented in a micro-controller without the need for an optimization solver and satisfying the stringent memory and chronometric constraints of automotive electronic control units (ECUs) [45].

eMPC allows one to solve the optimization problem off-line for a given range of operating conditions of interest. By exploiting multi-parametric programming techniques, eMPC computes the optimal control action off-line as an "explicit" function of the state and reference vectors, so that on-line operations reduce to a simple function evaluation. Such a function is piecewise affine in most cases, so that the MPC controller maps into some polyhedral regions that can be stored as a look-up table of linear gains [43].

Automotive actuators can often be adequately characterized by low dimensional models, and in this case an explicit implementation of the MPC controller becomes possible, whereby the solution is pre-computed off-line and its representation is stored for on-line application.

Beside eMPC advantages, this approach suffers from some drawbacks. In practice, eMPC is limited to relatively small problems (typically 1–2 inputs, up to 5–10 states, up to 3–4 free control moves) but allows one to reach very high sampling frequencies (up to 1 MHz) and requires a very simple control code to be embedded in the system [43].

For a fixed number of states and reference signals, the complexity of the solution is given by the number of regions that form the explicit solution. This number mainly depends (exponentially, in the worst case) on the number of constraints (and also of

binary variables/system modes in the hybrid case) included in the MPC problem formulation, and only mildly on the number of state variables. It also depends on the number of optimization variables. Bemporad [46] has introduced a table which shows such dependencies on random linear MPC problems. In the multi-parametric QP case, an upper-bound to the number of regions is 2 to the power of constraints number, which is the number of all possible combinations of active constraints at optimality [43].

One problem with the approach is that, as the horizon size, the number of states, and the number of constraints grow, the number of polyhedral regions grows quickly, making the look-up table approach difficult to implement in practice. Therefore, various eMPC design and search algorithms with some minor sacrifice of optimality were proposed [42].

Industrial problems addressed through eMPC techniques have been reported in technical papers, starting from what is probably the first work in this domain (traction control) [47]. The most suitable applications for eMPC are fast-sampling problems (in the order of 1–50 ms) and relatively small size. Most of the applications of eMPC have been reported in the automotive domain and electrical power converters.

Borrelli et al. [48] describe a hybrid model and a MPC strategy for solving a traction control problem. The problem is tackled in a systematic way from modeling to control synthesis and implementation. The resultant optimal controller is converted to its equivalent piecewise affine form by employing multi-parametric programming techniques, and finally experimentally tested on a car prototype. Nausa et al. [49] describe a systematic approach for the design of a parameterized adaptive cruise control (ACC), based on eMPC. A unique feature of the synthesized ACC is its parameterization in terms of key characteristics, which, after the parameterization, makes it easy and intuitive to tune, even for the driver.

Stewart and Borrelli [50] present the development and implementation of a practical eMPC approach that allows sub-controllers to receive and accommodate time-varying setpoints and constraints from higher levels in standard industrial automotive controller hierarchies. The proposed approach was demonstrated on production ECU controlling a real 2.2 litre diesel engine in which the variable-geometry turbocharger (VGT) and exhaust gas recirculation (EGR) actuators were used to track setpoints on mass air flow (MAF) and manifold absolute pressure (MAP) sensors while respecting a time-varying constraint on engine-out NOx emissions. Widd et al. [51] present results on model predictive control of the combustion phasing in an homogeneous charge compression ignition (HCCI) engine based on a hybrid model formulation composed of several linearizations of a physics-based non-linear model. The explicit representation of the MPC was implemented experimentally and the performance during setpoint changes was compared to that of a switched state feedback controller. The hybrid MPC produced smoother transients without overshoot when the setpoint change traversed several linearizations.

Di Cairano et al. [52] have introduced an energy management strategy that focuses on optimizing the engine efficiency for a series HEV. The experimental results executed on a fully functional vehicle on a chassis-roll dynamometer using the UDDS cycle, show fuel economy improvements with respect to two base strategies.

2.2.5 Control-Relevant Parameter Estimated eMPC

Effective system identification of highly interactive processes for multi-variable control purposes has been viewed as a challenging problem by many investigators [53–55]. Recently, control-relevant parameter estimation has become an important subject to control researchers, because of its potential to make significant improvements in the field of model-based control. It is suggested that model identification and controls design should not be performed independently, and this has led to the iterative design of a model-based controller. While conventional identification approaches emphasize obtaining an accurate model, the objective of the control-relevant identification is to find an approximate model that is appropriate for the design of a high-performance controller.

The goal of control-relevant parameter estimation is to make appropriate choices of design variables in the estimation procedure so that the important properties of the plant with respect to the intended control application are retained in the estimated model [56]. Control-relevant parameter estimation problems naturally arise in reduced-order controller design [57, 58] and system identification [59]. Various control-relevant parameter estimation schemes [60–63] have been proposed and also achieved good results in several industrial applications [64, 65].

Among them, most of the control-relevant identification schemes adopt a time domain identification technique and use frequency weighting functions in both identification and controller design stages [65–67]. The frequency domain weighting function in the controller design stage is obtained from differences between estimated and predicted input/output signals during iteration. The cost function is updated at every iteration step and the results of identification considerably affect the controller gain tuning, which means that the cost function is changed at every step to minimize the cost function. Jun et al. [68] describe a MATLAB-based computer-aided design tool, IRA-HPC, which accomplishes integrated system identification and robustness analysis for receding horizon control (RHC), a model predictive control algorithm implemented on the Application Module of the Honeywell TDC 3000 distributed control system and shows its benefits in terms of simplifying the choices of design variables in integrated identification and control design. Lee and Rivera [56] has presented a novel integrated framework for multi-variable system identification and control system design leading to desirable models for the control of highly interactive multi-variable process systems. The resulting models serve as a useful nominal model for a high performance advanced control system, such as model predictive control. Verboven et al. [69] presents a computational approach for the frequency-domain identification of multi-variable, discrete-time transfer function models based on a cost function minimization. The algorithm is optimized for the parametric characterization of complex high-order multi-variable systems requiring a large number of model parameters, including sparse matrix methods for the reduction of computation time and memory requirements. The algorithm supports a multi-variable frequency-dependent weighting, which generally improves the quality of the transfer function model estimate. The overall approach is successfully demonstrated for a typical case

encountered in experimental structural dynamics modeling (using modal analysis) and compared with related algorithms in order to assess the gain in computational efficiency.

2.2.6 Equivalent Consumption Minimization Strategy

ECMS has been widely employed for building EMS schemes. This approach is helpful in developing an optimal control system that minimizes total energy consumption, whether electric energy (consumed by the battery) or fuel (consumed by the engine) [70–79].

Fuel consumed by the engine and electric energy expended by the battery cannot be compared directly; as such, the ECMS scheme is utilized to obtain an optimal control system that curtails total fuel consumption. Tulpule et al. [80] applied the ECMS technique to series and parallel PHEVs, taking into consideration two different strategies: CD and Blended, resulting in greater efficiencies for both designs dependent on trip conditions. Musardo et al. [81] proposed an A-ECMS method based on driving conditions, calculating the equivalency factor in ECMS technique for parallel HEVs.

With its ability to decouple the engine crankshaft from the road and ability to allow electric machines to move the engine operation point to its maximum fuel efficiency, the power-split configuration has the greatest advantages among the different PHEV architectures available. He [82] adopted the A-ECMS for power-split PHEVs. Wollaeger and Rizzoni [83] showed that a near-optimal EMS strategy can be achieved by depleting the battery SOC with respect to the driving distance. This optimal strategy assumes that prior knowledge of the travelling distance exists, while speed trajectory remains unknown. Stockar et al. [84, 85] proposed a novel supervisory EMS strategy for series-parallel PHEVs. The researchers applied the ECMS to compute the overall vehicle CO_2 emissions, taking into account emissions indirectly produced during electric power generation, while also applying PMP to find the optimal power distribution between the engine and motor-generators.

2.3 Cruise Controller

Reports released by the World Health Organization estimate worldwide costs resulting from road traffic injuries at approximately $518 billion annually [86], with human perception error being the most significant factor, contributing to 90.

2.3.1 Adaptive Cruise Controller

ACC systems represent a significant step towards traffic and transportation system efficiency improvements, as they aim to increase traffic flow, while reducing the likelihood of scenarios causing road accidents. Sensor technologies enable ACC to measure headway distance from vehicles and other objects, while a controller interprets the data along with current driving and environmental conditions to adjust both the velocity and distance to them. Utilizing more effective ACCs can considerably enhance the driving safety and performance of vehicles [87, 88]. ACC systems come in two groups: rule-based and model-based approaches. Model-based approaches include sliding mode [89], optimal control [90], and model predictive control [91].

To lower the frequency collisions involving passengers and other vulnerable road users Ferrara and Vecchio [92] proposed a collision avoidance ACC system. They employed a supervisory control technique based on trip data to switch the control mode between normal state and collision avoidance mode. The normal state utilized the sliding mode control system to follow the preferred distance from the preceding vehicle. The control mode was changed to collision avoidance when the supervisory control detected the possibility of collision, by performing a collision avoidance maneuver or by activating the emergency breaking.

Moon et al. [93] integrated ACC with collision avoidance by proposing a control scheme that operates the vehicle in three modes based on specific driving conditions, including comfort-mode, large- deceleration mode, and severe-braking mode. Driving conditions are identified through two indicators: the time-to-collision index and the non-dimensional warning index, which is calculated based on vehicle spacing. The no-crash condition employed manual driving data to tune the controller parameters. Full vehicle testing confirmed that the ACC system was capable of operating the vehicle resembling human operation and collision avoidance.

2.3.2 Ecological Cruise Controller

Some researchers have begun to investigate ecological cruise controller for conventional vehicles. Huang [94] employed constrained nonlinear programming to predict optimal throttle, gear shifting and velocity trajectory of heavy trucks. Hellstom et al. [95] incorporated MPC in real-time, set the optimum speed trajectory as a set point of the cruise control system, and implemented this solution on a heavy diesel truck. The results demonstrated enhanced fuel efficiency in look-ahead cruise control systems compared to conventional cruise control systems. This was particularly evident during demonstrations on hilly roads, as the look-ahead control system demonstrably improved fuel economy by efficiently expanding the vehicles speed prior to an uphill climb and decreasing it prior to a downhill slope. Zhaun and Xia [96] formulated a solution to the nonlinear problem, with specific application to trains, by propos-

ing and output regulation approach incorporating measurement feedback, with the locomotive following a prearranged speed profile.

MPC theory has received a great deal of attention for automotive control applications, as its formulation is best suited for online optimizations, as well as dealing with constraints in large multivariate systems [49, 97–99]. The scope of MPC technique application has also been extended to ITS, where principal focus is on using impending information from the intended route to adjust the control variable of vehicles. An extensive literature review demonstrates the applicability of MPC controllers for ITS applications. Li et al. [100] employed MPC in their ACC system design and proposed a multi-objective controller, which simultaneously considered fuel economy and speed tracking. Kamal et al. [101] took advantage of the MPC theory to design a novel ecological driving system for running a vehicle on roads with up-down slopes. Simulation results confirmed that the use of MPC controllers extensively reduce fuel consumption.

Wang et al. [102] analysis of driver assistance systems established that using a control strategy with the ability to predict the future dynamics of proceeding vehicles' speed can result in efficient ACC, demonstrating that MPC theory has a high potential to be used for ACC systems. Groot et al. [103] applied the MPC theory to design an integrated predictive traffic and emission control, observing that the MPC technique can be fused with a mixed-logic dynamical model description to efficiently control system performance. Groot et al. [104] employed the MPC technique to reduce traffic delays and emissions on urban roads, demonstrating that the predictive nature of MPC significantly contributes to the efficiency of results. Kamal et al. [105, 106] used MPC in designing a predictive control scheme capable of using impending information on traffic flow to adjust vehicle distance under a bounded driving torque condition. An exhaustive comparative study for different traffic conditions demonstrated the efficacy of the MPC-based ACC.

2.4 Summary

This chapter introduced optimal powertrain control strategies including trip planning, energy management strategy, and cruise controller. Also, it described what have been done for designing optimal powertrain control strategies in the literature. Its found that using advance knowledge of the entire driving cycles can improve powertrain controllers. Processing trip date increases the complexity of control systems, and presents significant challenges for real-time implementation.

Most research to date has been conducted with the goal of developing efficient control systems for traditional gasoline powered or conventional hybrid vehicles. Some recent studies have reported progress towards energy management of power-split PHEVs, which have more complicated dynamics and provide more flexibility to reduce total energy cost. There exist only rare reports in the literature addressing the applicability of ACCs for HEVs and PHEVs. A complicated architecture of a PHEV propulsion system can restore energy during regenerative braking. Therefore,

the ecological cruise control problem in PHEV is more complicated than that for a conventional vehicle. This book develops a novel energy-optimal controller for PHEV that addresses these challenges.

References

1. Shen L., Huang, M.: Assessing dynamic neural networks for travel time prediction. In: Applied Informatics and Communication International Conference, pp. 469–477 (2011)
2. Park, B., Malakorn, K., Lee, J.: Quantifying benefits of cooperative adaptive cruise control toward sustainable transportation system. Technical Report. May, Center for Transportation Studies, University of Virginia, Charlottesville, VA (2011)
3. Park, J., Chen, Z., Kiliaris, L., Kuang, M., Masrur, M., Phillips, A., Murphey, Y.: Intelligent vehicle power control based on machine learning of optimal control parameters and prediction of road type and traffic congestion. IEEE Trans. Veh. Technol. **58**, 4741–4756 (2009)
4. Keulen, T.V., Jager, B.D., Foster, D., Steinbuch, M.: Velocity trajectory optimization in hybrid electric trucks. In: American Control Conference, pp. 5074–5079 (2010)
5. Keulen, T.V., Jager, B.D., Steinbuch, M.: Optimal trajectories for vehicles with energy recovery options. In: 18th IFAC World Congress Milano, pp. 3831–3836 (2011)
6. Keulen, T.V., Naus, G., Jager, B.D., Molengraft, V., Steinbuch, M., Aneke, E.: Predictive cruise control in hybrid electric vehicles. World Electr. Veh. J. **3**, 1–11 (2009)
7. Keulen, T.V., Jager, B.D., Serrarens, A., Steinbuch, M.: Optimal energy management in hybrid electric trucks using route information. Oil Gas Sci. Technol. **65**, 103–113 (2010)
8. Gong, Q., Li, Y., Peng, Z.: Trip-based optimal power management of plug-in hybrid electric vehicles. IEEE Trans. Veh. Technol. **57**, 3393–3401 (2008)
9. Gong, Q., Li, Y., Peng, Z.: Optimal power management of plug-in HEV with intelligent transportation system. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 1–6 (2007)
10. Gong, Q., Tulpule, P., Marano, V., Rizzoni, G.: The role of ITS in PHEV performance improvement. In: American Control Conference, pp. 2119–2124 (2011)
11. Bin, Y., Li, Y., Gong, Q.: Multi-information integrated trip specific optimal power management for plug-in hybrid electric vehicles. In: American Control Conference, pp. 4607–4612 (2009)
12. Katsargyri, G., Kolmanovsky, I., Michelini, J., Kuang, M., Phillips, A., Rinehart, M., Dahleh, M.: Path dependent receding horizon control policies for hybrid electric vehicles. In: American Control Conference, pp. 4613–4617 (2009)
13. Katsargyri, G.: Optimally controlling hybrid electric vehicles using path forecasting. Master of Science, Massachusetts Institute Of Technology (2008)
14. Rousseau, A., Pagerit, S., Gao, D.: Plug-in hybrid electric vehicle control strategy parameter optimization. J. Asian Electr. Veh. **6**, 1125–1133 (2008)
15. Lin, C., Jeon, S., Peng, H., Moo Lee, J., Moo, J.: Driving pattern recognition for control of hybrid electric trucks. Veh. Syst. Dyn. Int. J. Veh. Mech. Mobility **42**, 41–58 (2004)
16. Bashash, S., Arbor, A., Moura, S.J.: Battery health-conscious plug-in hybrid electric vehicle grid demand prediction. In: ASME 2010 Dynamic Systems and Control Conference, pp. 1–9 (2010)
17. Moura, S.J., Stein, J.L., Fathy, H.K.: Battery-health conscious power management in plug-in hybrid electric vehicles via electrochemical modeling and stochastic control. IEEE Trans. Control Syst. Technol. **1**(3), 1–16 (2013)
18. Moura, S.J., Fathy, H., Callaway, D., Stein, J.: A stochastic optimal control approach for power management in plug-in hybrid electric vehicles. IEEE Trans. Control Syst. Technol. **19**, 545–555 (2011)
19. Murphey, Y.: Intelligent vehicle power management: an overview. Comput. Intell. Automot. Appl. SE **10**(190), 169–190 (2008)

20. Manzie, C., Watson, H., Halgamuge, S.: Fuel economy improvements for urban driving: Hybrid versus intelligent vehicles. Transp. Res. Part C Emerg. Technol. **15**, 1–16 (2007)
21. Kim, T.C, Manzie, C., Sharma, R.: Two-stage optimal control of a parallel hybrid vehicle with traffic preview. In: The 18th IFAC World Congress, vol. 2, pp. 2115–2120 (2011)
22. Bartholomaeus, R., Klingner, M., Lehnert, M.: Prediction of power demand for hybrid vehicles operating in fixed-route service. In: Proceedings of the 17th IFAC World Congress, pp. 5640–5645 (2008)
23. Ichikawa, S., Yokoi, Y., Doki, S., Okuma, S., Naitou, T., Shiimado, T., Miki, N.: Novel energy management system for hybrid electric vehicles utilizing car navigation over a commuting route. In: Proceedings of the 2004 IEEE Intelligent Vehicles Symposium, pp. 161–166 (2004)
24. Gonder, J., Markel, T., Simpson, A., Thornton, M.: Using GPS travel data to assess the real world driving energy use of plug-in hybrid electric vehicles (PHEVs). Transp. Res. Rec. **2017**, 26–32 (2007)
25. Gonder, J.: Route-Based Control of Hybrid Electric Vehicles. In: SAE SP, vol. 2199 (2008)
26. Serrao, L., Onori, S., Rizzoni, G.: A comparative analysis of energy management strategies for hybrid electric vehicles. J. Dyn. Syst. Meas. Control **133** (2011)
27. Lin, C., Peng, H., Grizzle, J., Liu, J., Busdiecker, M.: Control system development for an advanced-technology medium-duty hybrid electric truck. In: SAE paper, vol. 20 (2003)
28. Lin, C., Peng, H., Grizzle, J.: Power management strategy for a parallel hybrid electric truck. IEEE Trans. Control Syst. Technol. **11**, 839–849 (2003)
29. Gonder, J., Markel, T.: Energy management strategies for plug-in hybrid electric vehicles. Energy **1** (2007)
30. O'Keefe, M., Markel, T.: Dynamic programming applied to investigate energy management strategies for a plug-in HEV. Technical Report, National Renewable Energy Laboratory (2006)
31. Razavian, R.S.: Design and hardware-in-the-loop testing of optimal controllers for hybrid electric powertrains. Ph.D. thesis, University of Waterloo (2012)
32. Razavian, R.S., Taghavipour, A., Azad, N.L., McPhee, J.: Design and evaluation of a real-time fuel-optimal control system for series hybrid electric vehicles. Int. J. Electr. Hybrid Veh. **4**, 260–288 (2012)
33. Ebbesen, S., Elbert, P., Guzzella, L.: Battery state-of-health perceptive energy management for hybrid electric vehicles. IEEE Trans. Veh. Technol. **16**, 2893–2900 (2012)
34. Re, L., Ortner, P., Alberer, D.: Chances and challenges in automotive predictive control, 1–22 (2010)
35. Wang, L.: Model Predictive Control System Design and Implementation Using MATLAB. Springer (2009)
36. Kim, T.S., Manzie, C., Sharma, R.: Model predictive control of velocity and torque split in a parallel hybrid vehicle. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 2014–2019 (2009)
37. Borhan, H., Vahidi, A., Phillips, A.M., Kolmanovsky, I.: Predictive energy management of a power-split hybrid electric vehicle. In: American Control Conference, pp. 3970–3976 (2009)
38. Borhan, H., Vahidi, A., Phillips, A.M., Kuang, M.L., Kolmanovsky, I.V., Di Cairano, S.: MPC-based energy management of a power-split hybrid electric vehicle. IEEE Trans. Control Syst. Technol. **20**, 593–603 (2012)
39. Taghavipour, A., Azad, N.L., McPhee, J.: An optimal power management strategy for power split plug-in hybrid electric vehicles. Int. J. Veh. Des. **60**, 286–304 (2012)
40. Taghavipour, A., Masoudi, R., Azad, N.L., McPhee, J.: High-fidelity modeling of a power-split plug-in hybrid electric powertrain for control performance evaluation. In: ASME: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 1–9 Aug 2013 (2013)
41. Kouramasa, K.I., Panosa, C.: Fascab, N.P., Pistikopoulos, E.N.: An algorithm for robust explicit/multi-parametric model predictive control. Automatica **49**, 381–389 (2013)
42. Lee, J.: Model predictive control: review of the three decades of development. Int. J. Control Autom. Syst. **9**, 415–424 (2011)

43. Alessio, A., Bemporad, A.: A survey on explicit model predictive control. Lecture Notes in Control and Information Sciences, vol. 384, pp. 345–369 (2009)
44. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N.: The explicit linear quadratic regulator for constrained systems. Automatica **38**, 3–20 (2002)
45. Di Cairano, S., Bemporad, A., Kolmanovsky, I., Hrovat, D.: Model predictive control of magnetically actuated mass spring dampers for automotive applications. Int. J. Control **80**, 1701–1716 (2000)
46. Bemporad, A.: Model-based predictive control design: New trends and tools. In: 45th IEEE Conference on Decision and Control (CDC), pp. 6678–6683 (2006)
47. Borrelli, F., Bemporad, A., Fodor, M., Hrovat, D.: A hybrid approach to traction control. In: Di Benedetto, M.D., Sangiovanni-Vincentelli, A.L. (eds.) In: HSCC 2001. Lecture Notes in Computer Science, vol. 2034, pp. 162–174 (2001)
48. Borrelli, F., Bemporad, A., Fodor, M., Hrovat, D.: An mpc/hybrid system approach to traction control. IEEE Trans. Control Syst. Technol. **14**, 541–552 (2006)
49. Nausa, G., Ploegb, J., Van de Molengrafta, M., Heemelsa, W., Steinbuch, M.: Design and implementation of parameterized adaptive cruise control: an explicit model predictive control approach. Control Eng. Pract. **18**, 882–892 (2000)
50. Stewart, G., Borrelli, F.: A model predictive control framework for industrial turbodiesel engine control. In: 48th IEEE Conference on Decision and Control (CDC), pp. 5704–5711 (2008)
51. Widd, A., Hsien-Hsin, L., Gerdesand, J., Tunestal, P., Johansson, R.: Highspeed on-line MPC based on a fast gradient method applied to power converter control. In: American Control Conference (ACC), pp. 420–425 (2011)
52. Di Cairano, S., Liang, W., Kolmanovsky, I.V., Kuang, M.L., Phillips, A.M.: Engine power smoothing energy management strategy for a series hybrid electric vehicle. In: American Control Conference (ACC), pp. 2101–2106 (2011)
53. Andersen, H., Kummel, M.: Evaluating estimation of gain directionality parts 1: methodology and 2: a case study of binary distillation. J. Process Control **2**, 59–86 (1992)
54. Jacobsen, E., Skogestad, S.: Inconsistencies in dynamic models for ill-conditioned plants application to low-order models of distillation columns. Ind. Eng. Chem. Res. **33**, 631–640 (1994)
55. Koung, C., MacGregor, J.: Design of identification experiments for robust control: a geometric approach for bivariate processes. Ind. Eng. Chem. Res. **32**, 1658–1666 (1993)
56. Lee, H., Rivera, D.E.: An integrated input signal design and control-relevant parameter estimation approach for highly interactive multivariable systems. In: American Control Conference (ACC) (2006)
57. Rivera, D.E., Morari, M.: Control-relevant model reduction problems for siso h_2, h_∞, and μ-controller synthesis. Int. J. Control **46**, 505–527 (1987)
58. Rivera, D.E., Morari, M.: Plant and controller reduction problems for closed-loop performance. In: 27th IEEE Conference on Decision and Control (CDC), pp. 1143–1148 (1988)
59. Rivera, D.E., Webb, C., Morari, M.: A control-relevant identification methodology. In: Annual AIChE Meeting (1987)
60. Schrama, R.: Approximate identification and control design with application to a mechanical system. Ph.D. dissertation, Delft University of Technology (1992)
61. Zang, Z., Bitmead, R., Gevers, M.: h_2 iterative model refinement and control robustness enhancement. In: 30th IEEE Conference on Decision and Control (CDC), pp. 279– 284 (1991)
62. Hakvoort, R.G., Schrama, R., Van den Hof, P.: Approximate identification with closed-loop performance criterion and application to LQG feedback design. Automatica **30**, 679–690 (1994)
63. Hjalmarsson, H., Gevers, M., Gunnarsson, S., Lequin, O.: Iterative feedback tuningtheory and applications. IEEE Control Syst. Mag. **18**, 26–41 (1998)
64. de Callafon, R.A., Van den Hof, P., Steinbuch, M.: Control relevant identification of a compact disc pick-up mechanism. In: 32nd IEEE Conference on Decision and Control (CDC), pp. 2050–2055 (1993)

65. Partanen, A.G., Bitmead, R.R.: The application of an iterative identification and controller design to a sugar cane crushing mill. Automatica **31**, 1547–1563 (1995)
66. Zang, Z., Bitmead, R., Gevers, M.: Iterative weighted least-squares identification and weighted LQG control design. Automatica **31**, 15771594 (1995)
67. Michelberger, P., Bokor, J., Palkovics, L., Nandori, E., Gaspar, P.: Iterative identification and control design for uncertain parameter suspension system. In: IFAC Transportation Systems. Preprints of the 8th IFAC/IFIP/IFORS Symposium, vol. 2, pp. 464–469 (1997)
68. Jun, K.S., Rivera, D.E., Elisante, E., Sater, V.E.: A computer-aided design tool for robustness analysis and control-relevant identification of horizon predictive control with application to a binary distillation column. J. Process Control **6**, 177–186 (1996)
69. Verboven, P., Guillaume, P., Cauberghe, B.: Multivariable frequencyresponse curve fitting with application to modal parameter estimation. Automatica **41**, 1773–1782 (2005)
70. Liu, J., Peng, H.: Control optimization for a power-split hybrid vehicle. In: American Control Conference (2006)
71. Liu, J., Peng, H.: Modeling and control of a power-split hybrid vehicle. IEEE Trans. Control Syst. Technol. **16**, 1242–1251 (2008)
72. Serrao, L., Rizzoni, G.: Optimal control of power split for a hybrid electric refuse vehicle. In: American Control Conference, pp. 4498–4503 (2008)
73. Pisu, P., Rizzoni, G.: A comparative study of supervisory control strategies for hybrid electric vehicles. IEEE Trans. Control Syst. Technol. **15**, 506–518 (2007)
74. Paganelli, G., Guerra, T.M., Delprat, S., Santin, J., Delhom, M., Combes, E.: Simulation and assessment of power control strategies for a parallel hybrid car. Proc. Inst. Mech. Eng. Part D J. Autom. Eng. **214**, 705–717 (2000)
75. Paganelli, G., Delprat, S., Guerra, T.M., Rimaux, J., Santin, J.: Equivalent consumption minimization strategy for parallel hybrid powertrains. In: IEEE 55th Vehicular Technology Conference, vol. 4, pp. 2076–2081 (2002)
76. Sezer, V., Gokasan, M., Bogosyan, S.: A novel ECMS and combined cost map approach for high-efficiency series hybrid electric vehicles. IEEE Trans. Veh. Technol. **60**, 3557–3570 (2011)
77. Shan, M.: Modeling and control strategy for series hydraulic hybrid vehicles. Doctor of Philosophy, The University of Toledo (2009)
78. Zhang, C., Vahidi, A., Pisu, P., Tennant, K.: Role of terrain preview in energy management of hybrid electric vehicles. IEEE Trans. Veh. Technol. **59**, 1139–1147 (2010)
79. Zhang, C., Vahidi, A.: Route preview in energy management of plug-in hybrid vehicles. IEEE Trans. Control Syst. Technol. **20**, 546–553 (2012)
80. Tulpule, P., Marano, V., Rizzoni, G.: Effects of different PHEV control strategies on vehicle performance. In: American Control Conference, pp. 3950–3955 (2009)
81. Musardo, C., Rizzoni, G., Staccia, B.: A-ECMS: an adaptive algorithm for hybrid electric vehicle energy management. Eur. J. Control, 509–524 (2005)
82. He, Y., Chowdhury, M., Pisu, P., Ma, Y.: An energy optimization strategy for power-split drivetrain plug-in hybrid electric vehicles. Transp. Res. Part C Emerg. Technol. **22**, 29–41 (2012)
83. Wollaeger, J., Rizzoni, G.: ITS in energy management systems of PHEV's. Master of Science, Ohio State University (2012)
84. Stockar,S., Marano, V., Rizzoni, G., Guzzella, L.: Optimal control for plug-in hybrid electric vehicle applications. In: American Control Conference, pp. 5024–5030 (2010)
85. Stockar, S., Marano, V., Canova, M., Rizzoni, G., Guzzella, L.: Energy-optimal control of plug-in hybrid electric vehicles for real-world driving cycles. IEEE Trans. Veh. Technol. **60**, 2949–2962 (2011)
86. Global status report on road safety: time for action. World Health Organization (2009)
87. Xiao, L., Gao, F.: A comprehensive review of the development of adaptive cruise control systems. Veh. Syst. Dyn. **48**, 1167–1192 (2010)
88. Bengtsson, J.: Adaptive cruise control and driver modeling. No. November, Department of Automatic Control, Lund Institute of Technology (2001)

89. Lu, X., Hedrick, K., Drew, M.: ACC/CACC—control design, stability and robust performance. In: American Control Conference, pp. 4327–4332 (2002)
90. Liang, C., Peng, H.: Optimal adaptive cruise control with guaranteed string stability. Veh. Syst. Dyn. **32**, 37–41 (1999)
91. Corona, D., Schutter, B.D.: Comparison of a linear and a hybrid adaptive cruise controller for a SMART. In: 46th IEEE Conference on Decision and Control, pp. 4779–4784 (2007)
92. Ferrara, A., Vecchio, C.: Second order sliding mode control of vehicles with distributed collision avoidance capabilities, **19**, 471–477 (2009)
93. Moon, S., Moon, I., Yi, K.: Design, tuning, and evaluation of a full-range adaptive cruise control system with collision avoidance. Control Eng. Pract. **17**, 442–455 (2009)
94. Huang, W.: Design and evaluation of a 3D road geometry based heavy truck fuel optimization system. Doctor of Philosophy, Auburn University (2010)
95. Hellstrom, E., Ivarsson, M., Aslund, J., Nielsen, L.: Look-ahead control for heavy trucks to minimize trip time and fuel consumption. Control Eng. Pract. **17**, 245–254 (2009)
96. Zhuan, X., Xia, X.: Speed regulation with measured output feedback in the control of heavy haul trains. Automatica **44**, 242–247 (2008)
97. Luo, L., Liu, H., Li, P., Wang, H.: Model predictive control for adaptive cruise control with multi-objectives: comfort, fuel-economy, safety and car-following. J. Z. Univ. Sci. A **11**, 191–201 (2010)
98. Asadi, B., Vahidi, A.: Predictive cruise control: utilizing upcoming traffic signal information for improving fuel economy and reducing trip time. IEEE Trans. Control Syst. Technol. **19**, 707–714 (2011)
99. Shakouri, P., Ordys, A.: Nonlinear model predictive control approach in design of adaptive cruise control with automated switching to cruise control. Control Eng. Pract. **26**, 160–177 (2014)
100. Li, S., Li, K., Rajamani, R., Wang, J.: Model predictive multi-objective vehicular adaptive cruise control. IEEE Trans. Control Syst. Technol. **19**, 556–566 (2011)
101. Kamal, M., Mukai, M., Murata, J., Kawabe, T.: Ecological vehicle control on roads with up-down slopes. IEEE Trans. Intell. Transp. Syst. **12**, 783–794 (2011)
102. Wang, M., Daamen, W., Hoogendoorn, S., van Arem, B.: Driver assistance systems modeling by model predictive control. In: 15th International IEEE Conference on Intelligent Transportation Systems, pp. 1543–1548 (2012)
103. Groot, N., Schutter, B.D., Hellendoorn, H.: Integrated model predictive traffic and emission control using a piecewise-affine approach. IEEE Trans. Intell. Transp. Syst. **14**, 587–598 (2013)
104. Groot, N., Schutter, B.D., Xi, Y., Hellendoorn, H.: Integrated urban traffic control for the reduction of travel delays and emissions. IEEE Trans. Intell. Transp. Syst. **14**, 1609–1619 (2013)
105. Kamal, M., Imura, J., Hayakawa, T., Ohata, A., Aihara, K.: Smart driving of a vehicle using model predictive control for improving traffic flow. IEEE Trans. Intell. Transp. Syst. **15**, 878–888 (2014)
106. Kamal, M., Mukai, M., Murata, J., Kawabe, T.: Model predictive control of vehicles on urban roads for improved fuel economy. IEEE Trans. Control Syst. Technol. **21**, 831–841 (2013)

Chapter 3
High-Fidelity Modeling of a Plug-in Hybrid Electric Powertrain

This chapter introduces the validated high-fidelity simulation model of the Toyota Prius plug-in hybrid powertrain that will be used in MIL and HIL tests for the designed controls performance evaluation.

This chapter is organized as follows: after a brief introduction, the Toyota Prius Plug-in powertrain specifications are given. Then, a high-fidelity PHEV simulation model in the MapleSim software is introduced. Finally, the validation procedure (especially the Lithium-ion battery parameters) will be discussed.

3.1 Introduction

This chapter introduces a high-fidelity simulation model of the Toyota Prius plug-in hybrid powertrain developed in the MapleSim software. MapleSim is an environment for sensitivity analysis, model reduction and optimization for multi-domain system simulation with direct access to system equations. Symbolic calculation and optimized code generation in MapleSim reduces simulation time and makes the model more suitable for MIL and HIL tests. The performance of a PHEV is contingent on its battery and as such, this physics-based model of a power-split plug-in powertrain contains a chemistry-based Lithium-ion battery pack, which is vastly different from other models employed and analyzed in the literature.

Development of an advanced battery model that is both simple and accurate is an exigent undertaking. In order to enhance the fidelity of the model, a full chemistry-based Lithium-ion battery model developed by Newman and Tiedemann [1] and Doyle et al. [2], then simplified by Dao et al. [3], is applied in this research. The model is an isothermal battery model based on concentrated solution theory, porous electrode theory, and the variations in electronic/ionic conductivities and diffusivities. The equations governing the dynamic behavior of the battery model are non-linear

A. Taghavipour et al., *Intelligent Control of Connected Plug-in Hybrid Electric Vehicles*,
Advances in Industrial Control, https://doi.org/10.1007/978-3-030-00314-2_3

partial differential algebraic equations (PDAEs), which are not suitable for real time applications.

As a result, a simplified version of the full battery model presented by Dao et al. [3] is utilized in this book, that maintains the precision and substantially relegates the computational cost.

Dao et al. utilized the nature of the battery equations in their simplification technique, in addition to combining other systems, such as volume-averaging, Galerkin's method, and curve-fitting.

The resulting equations consist of 14 non-linear differential algebraic equations (DAEs) that carry both accuracy and simplicity for the battery simulation model. The development of a physical systems simulation model necessitates parameters that must be categorized by an acceptable degree of precision.

As such, parameter identification of the Lithium-ion battery plays a key role in evaluating this electrochemical subsystem of the full vehicle model.

The homotopy optimization procedure presented by Vyasarayani et al. [4] appears to be notably dependable among parameter identification techniques, due to its ability in approaching to global extremum.

The Autonomie software of Argonne National Lab was used to cross-validate the MapleSim model. Autonomie provides reliable component models that have been validated from a number of testing results [5]. It includes forward-looking models that allows advanced powertrain designers to develop realistic control strategies and assess component behaviors in a system environment by using models that are close to reality [6].

3.2 Toyota Prius Plug-in Hybrid Powertrain

The powertrain of this vehicle closely resembles the 3rd generation Toyota Prius, with the exception of the battery pack. The larger Lithium-ion battery pack delivers extended full electric driving range, and also significantly diminishes the environmental footprint. Characteristically, there are two electric motors (MG1 and MG2) that are connected to the engine and final drive with two planetary gear sets. The schematic of the power-split device [7] is shown in Fig. 3.1. Note that GR is the ratio of ring gear teeth number to sun gear teeth number in each planetary gear set.

One of the two planetary gear sets splits the power flow from the engine like the 2nd version, but an additional gear is operated as a reduction gear for the motor [5]. MG1 and the engine are connected to the sun gear and carrier of the first planetary gear set, respectively [7].The vehicles engine runs on the Atkinson cycle. The engine is beltless to reduce mechanical loss, and similar to the air conditioning compressor, its water pump runs on the electric side of the powertrain. The engine main specifications are shown in Table 3.1 [8]. A PHEV might have subsequent engine stop-start events during its trip, so an exhaust heat recirculation system is constructed to rapidly heat up the engine coolant to its normal operating temperature to prevent cold start events and improve fuel economy.

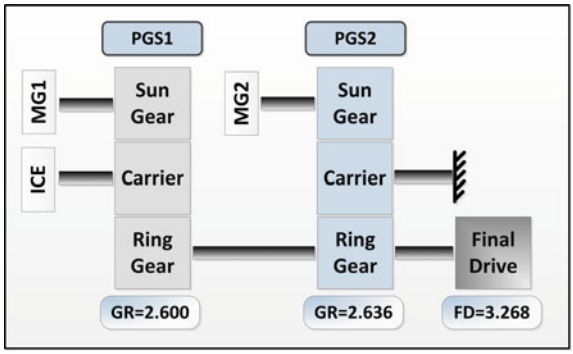


Fig. 3.1 The transmission of Toyota Prius plug-in hybrid

Table 3.1 Engine specifications

Engine model	2ZR-FXE (Atkinson cycle)
Engine type	In-Line 4 cylinder DOHC 16-valve
Displacement	1797 cc
Max. Torque	142 Nm @ 4000 rpm
Max. Power	73 kW @ 5200 rpm
Bore	80.5 mm
Stroke	88.3 mm
Compression ratio	13:1

The generator (MG1) and the traction motor (MG2) are air-cooled permanent magnet AC synchronous machines. MG1 can start the engine and charge the battery with the maximum power of 42 kW. The MG2 specification is demonstrated in Table 3.2 [8]. The addition of speed reduction planetary gear set requires the MG2 to operate at high speed levels.

The Prius Plug-in hybrid features a newly-developed large capacity Lithium-ion battery pack (Table 3.3) [8]. Contingent of the vehicle driving mode, this battery has two levels of output power. According to the Environmental Protection Agency (EPA) rating the electric mode (EV) range is up to 18 km. Yhe energy management strategy switches to the hybrid mode (HV) after partial battery depletion.

Table 3.2 MG2 specifications

Type	Permanent magnet AC synchronous
Max. Torque	300 Nm
Max. Power	60 kW
Max. Speed	13600 rpm
Voltage	650 V

Table 3.3 Battery-pack specifications

Type	Lithium-ion
Number of cells	56
Number of cells in each Module	14
Nominal voltage	207.2 V
Nominal capacity	4.5 kWh
Power output	38 kW
Weight	80 kg

3.3 High-Fidelity Model in MapleSim

MapleSim is an environment for multi-domain system simulation with direct access to system equations, making it more convenient to perform model reduction and optimization. Symbolic calculation in MapleSim reduces simulation time and makes the model more suitable for MIL and HIL tests. The high-fidelity simulation model of the Toyota Prius plug-in hybrid powertrain is developed in MapleSim (Fig. 3.2). This model consists of five main parts for the power-split architecture: (1) internal combustion engine, (2) electric drive, (3) battery, (4) driveline, and (5) vehicle dynamics.

3.3.1 Mean-Value Internal Combustion Engine

Different model complexities are created for different applications in the area of engine modeling [9, 10]. A mean-value engine modeling approach has been the most widely employed engine modeling technique by most researchers for controls development [11, 12]. The PHEV high-fidelity simulation model utilizes a mean-value engine model that is precise and sufficiently swift for powertrain simulation.

The mean value engine model consists of four parts: (1) engine control unit (ECU), (2) throttle body, intake/exhaust manifold, and combustion chamber.

The throttle body obtains the throttle angle command from the ECU and alters it to the air flow entering the intake manifold. Air that passes the throttle cannot enter the combustion chamber. For naturally aspirated engines, the amount of air depends on intake manifold pressure and engine speed, like what is shown here—referred to as engine volumetric efficiency. The power generated in the combustion chamber is determined in correspondence with the air fuel ratio, and mechanical and thermal losses.

The first dynamic equation (3.1) is related to the manifold pressure [13]:

$$\dot{P}_m = -\frac{\eta_v N_{cyl} V_D \omega_e}{60 N_{eng} V_m} P_m + \frac{R_{air} T_{man}}{V_m} (C_D \times MA \times PRI) A_{th} \tag{3.1}$$

where A_{th} is the throttle area, one of the control inputs. Throttle area can be found according to throttle angle θ and geometry (d_{th} and D_{th} are diameter values for input and output vents and θ_0 is the angle when throttle is totally closed):

$$A_{th} = \frac{d_{th}.D_{th}}{2}\left(-\sqrt{1-\left(\frac{d_{th}}{D_{th}}\right)^2}+\sqrt{1-\left(\frac{d_{th}cos\theta_0}{D_{th}cos\theta}\right)^2}\right)$$
$$+\frac{D_{th}^2}{2}\left(sin^{-1}\left(\sqrt{1-\left(\frac{d_{th}}{D_{th}}\right)^2}\right)-\frac{cos\theta_0}{cos\theta}sin^{-1}\left(\sqrt{1-\left(\frac{d_{th}cos\theta_0}{D_{th}cos\theta}\right)^2}\right)\right) \tag{3.2}$$

Moreover η_v is the volumetric efficiency and a function of manifold pressure and engine speed (ω_e), N_{cyl} is the number of cylinders that is 4 here, V_d and V_m are the engine displacement and air manifold volume respectively, N_{eng} is 2 for four stroke engine, R_{air} is air constant, T_{man} is the manifold temperature (considered constant for simplicity), C_D is the throttle discharge coefficient, $MA = \frac{P_0}{\sqrt{R_{air}T_0}}$ where P_0 and T_0 are atmosphere pressure and temperature, and PRI is a non-dimensional value to consider sub and supersonic air flow which depends on air heat capacity, manifold and atmosphere pressure:

$$PRI = \begin{cases} \left(\frac{P_m}{P_0}\right)^{\frac{1}{\gamma}}\sqrt{\left(\frac{2\gamma}{\gamma-1}\right)\left(1-\left(\frac{P_m}{P_0}\right)^{\frac{\gamma-1}{\gamma}}\right)} & : \frac{P_m}{P_0} > \left(\frac{2\gamma}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \\ \sqrt{\gamma\left(\frac{2\gamma}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma+1)}}} & : \frac{P_m}{P_0} \leq \left(\frac{2\gamma}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \end{cases} \tag{3.3}$$

where γ is air heat capacity ratio.

The air mass rate entering the cylinders can be found as:

$$\dot{m}_{air} = \frac{\eta_v N_{cyl} V_D \, \omega_e}{60 N_{eng} V_m} P_m \tag{3.4}$$

The engine generated torque can be estimated via:

$$T_{ind} = \frac{\dot{m}_{air}}{AFR} \frac{H_f \eta_\Delta \eta_{AFR} \eta_i}{\omega_e} \tag{3.5}$$

where η_i and H_f are engine thermal efficiency (approximately a function of engine speed and manifold pressure) and gasoline heat of combustion. Meanwhile η_Δ and η_{AFR} are the efficiencies associated with ignition timing and air/fuel ratio.

The mean value engine model may be utilized to forecast the specified torque and fuel consumption based on throttle angle, air/fuel ratio and ignition timing. The main concern is to modify the parameters in order to match the fuel consumption map of the mean-value engine model to the real Toyota Prius plug-in hybrid engine, which will be addressed later in this chapter.

3.3.2 Electric Machines

Various components of the MapleSim software allow to model the high-fidelity power electronics along with permanent magnet AC synchronous machines. Nonetheless, the full model of these components increase simulation time and the powertrain model cannot run in real time. Consequently, we replace PM synchronous machines with DC motors with the same power rating. It is assumed that the electric drive is an ideal DC-DC converter, but the electric parts of the powertrain are modified to consider the associated efficiencies.

3.3.3 Lithium-Ion Battery Pack

The Lithium-ion battery used in the simulation model of the PHEV is a custom element developed in MapleSim, by which the inputs and outputs are related using an acausal representation.

The differential-algebraic equations are introduced into the custom component block along with the inputs and outputs required in the design of the control strategy for the PHEV simulation model. The differential equations are in the form of

$$\sum : \begin{cases} \boldsymbol{E}\dot{\boldsymbol{x}} = \mathbf{f}(\boldsymbol{x}(t), t) + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y}(t) = \boldsymbol{C}^T \boldsymbol{x}(t) \end{cases} \tag{3.6}$$

where $\boldsymbol{E}$ is the state matrix, which is singular due to algebraic constraints, $\boldsymbol{x}$ is the state vector, $\boldsymbol{y}$ is the output vector, $\mathbf{f}$ is the column vector of non-linear functions, $\boldsymbol{B}$ is the input matrix, and $\boldsymbol{u}$ is the input to the simulation model, which is the current passing through the battery, i_{batt}. Each step of the simulation requires five differential equations and nine algebraic equations to be solved concurrently. The battery state variables and parameters have been defined in [3]. The output vector $\boldsymbol{y}$ can be battery voltage, V_{batt}, and/or state of charge, SOC, depending on the control strategy.

3.3.4 Power-Split Device

As seen in Fig. 3.2, the power-split device is modeled as two sets of ideal planetary gears with appropriate gear ratios given in Fig. 3.1.

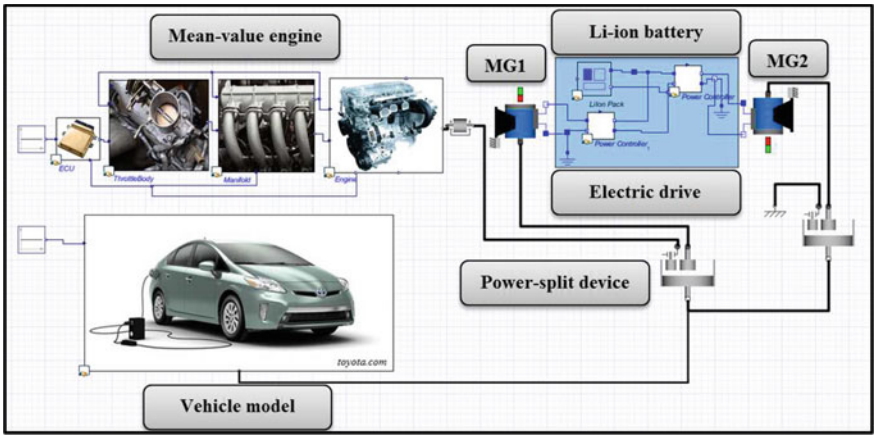


Fig. 3.2 Toyota Prius plug-in hybrid high-fidelity simulation model in MapleSim

3.3.5 Vehicle Model

The vehicle model has 14 degrees of freedom (DOF), including 6 DOF for the chassis. The 4 suspension displacements and 4 wheel spins add 8 DOF to the model. Moreover, this vehicle is capable of being steered, with tires on this vehicle are modeled according to the magic formula by Pacejka, including rolling resistance. In order to measure fuel consumption, the longitudinal dynamics of the vehicle is the most significant DOF. Nonetheless, the hybrid powertrain final drive is connected to the wheels of the vehicle model for the simulation to represent the full 3D vehicle motion, as well as cover plausible distinctive maneuver in customary drive cycles. The aerodynamic drag force is simulated through an external load acting on the vehicle's center of mass.

3.4 Model Validation

The parameters of the model must be categorized using the existing experimental data for each component in order to authenticate the Toyota Prius plug-in hybrid high-fidelity simulation model. The Autonomie software experimental database commonly recognized for energy management design in industry, —was utilized for the majority of the validation procedure. Autonomie is the revised version of the PSAT software, which was developed at Argonne National Laboratory. It is developed using MATLAB/Simulink and is a forward-looking model that simulates vehicle fuel economy, emissions, and performance in a realistic manner, and employs a virtual driver, who compares the trace speed and the actual vehicle speed and controls the vehi-

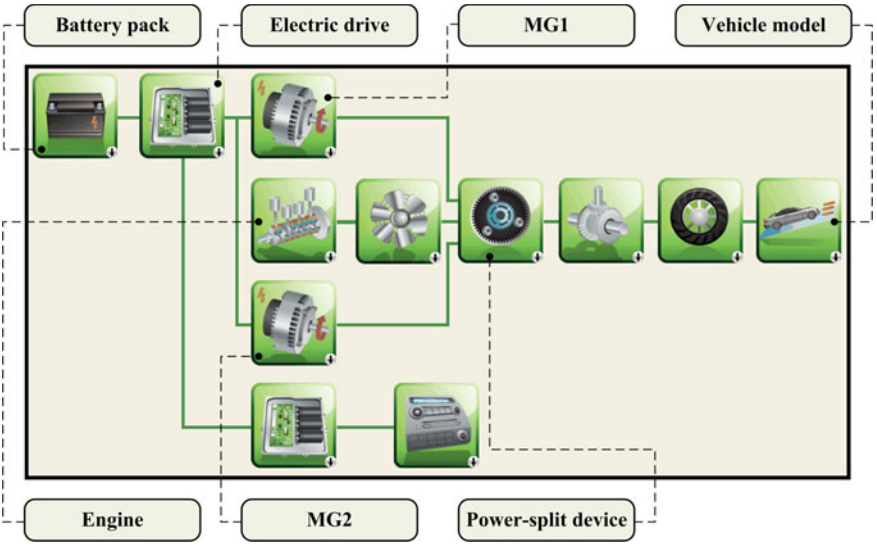


Fig. 3.3 PHEV powertrain model in Autonomie

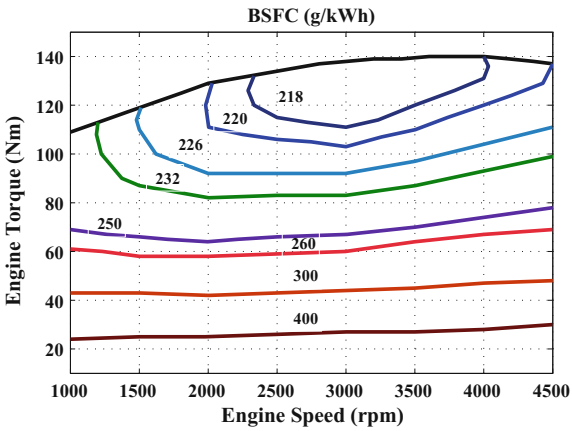


Fig. 3.4 The engine brake specific fuel consumption (BSFC) map

cle with a torque input [14]. The schematic of a PHEV powertrain in Autonomie is shown in Fig. 3.3.

3.4.1 Mean-Value Internal Combustion Engine

The Toyota Prius plug-in hybrid engine brake specific fuel consumption (BSFC) map (Fig. 3.4) is the validation reference.

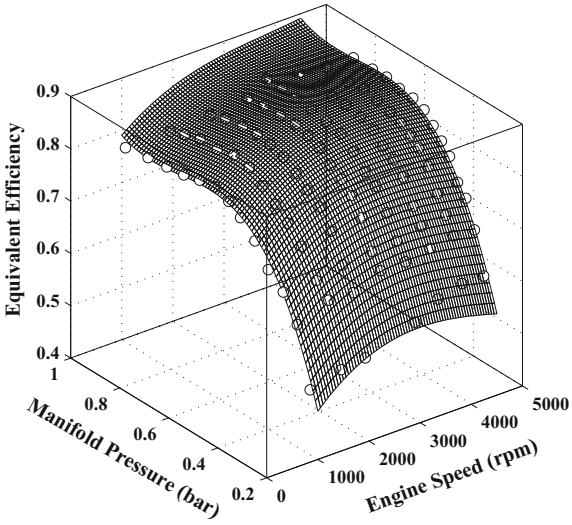


Fig. 3.5 Equivalent thermal efficiency for different values of engine speed and manifold pressure

The parameters related to the geometry of the engine are used in Eqs. (3.1)–(3.4). We assume that η_v, η_Δ and η_{AFR} are constant. To match the fuel consumption map of the mean-value engine model to the reference BSFC map, an equivalent thermal efficiency (η_i) is defined. It can be indicated that if the thermal efficiency (η_i) picks the values shown in Fig. 3.5, the reference fuel consumption map can be reproduced. So if a surface of 3rd-order polynomial is fit as the equivalent efficiency, the reference fuel consumption map can be obtained with an error of 2%. This error mostly belongs to the engine low torque region.

3.4.2 Electric Machines

As previously outlined, DC machines are used for modeling MG1 and MG2. Inductance and resistance of the armature are the two parameters available for DC machines; in addition to the torque and speed constants, this affects the transient response of the machine. There is no other model parameter to change the efficiency of the machine. To consider the efficiency map [7] of the traction motor (MG2), which is shown in Fig. 3.6, an energy dissipation element (a mechanical brake) is added in connection of the machines to the power-split device. If the brake activation signal follows $\sigma T_m(1-\eta_m)$ the efficiency map given in Fig. 3.6 can be reproduced in the MapleSim high-fidelity simulation model. Note that T_m is the motor torque, α is a constant related to the geometry of the brake, and η_m is the motor efficiency.

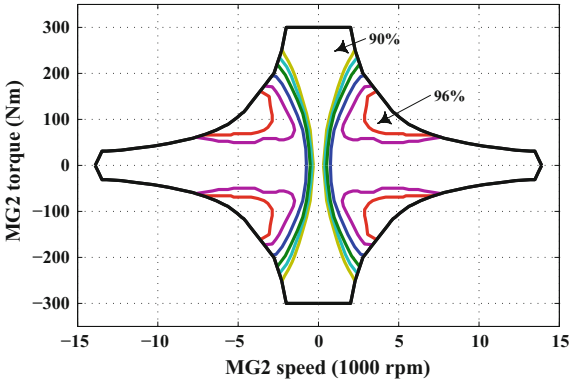


Fig. 3.6 MG2 efficiency map

3.4.3 Lithium-Ion Battery Pack

In order to achieve a more realistic battery model, its parameters must be projected based on experimental data, which in this chapter were extracted using the parameters of the full-order rigorous model used by Dao et al. [3].

Generally, one of the most frequent issues while utilizing determinist methods in optimization process of parameter identification is converging to local minima.

The homotopy method is an effective solution to obtain the global minimum of the optimization problem [4]. In homotopy optimization process, the original differential equations of the problem

$$\dot{\xi} = \mathbf{G}(\xi, \boldsymbol{\Gamma}, t) \tag{3.7}$$

in which ξ is the vector of state variables, $\mathbf{G}$ is the column vector of non-linear functions, and $\boldsymbol{\Gamma}$ is the column vector of the parameters to be identified, are modified by coupling the vector of experimental data, ξ_{exp}, to the original differential equation as [4]

$$\dot{\xi} = \mathbf{G}(\xi, \boldsymbol{\Gamma}, t) + \upsilon K_i(\xi_{exp} - \xi) \tag{3.8}$$

The homotopy parameter υ, which is initially one and decreased by a specified decrement, has been designed to construct the homotopy transformation as a high-gain observer. The gain K_i is incorporated to synchronize the experimental data and simulation results. Accordingly, minimizing the objective function

$$V(\boldsymbol{\Gamma}) = \frac{1}{2}\sum_{j=1}^{n}\left\{\int_0^T \left(\xi_{exp}^j - \xi^j(\boldsymbol{\Gamma}, t)\right)^2 \mathrm{d}t\right\} \tag{3.9}$$

with ξ^j and ξ_{exp}^j as the jth component of ξ and ξ_{exp}, respectively, is performed based on sensitivity equations used in evaluating the gradient and Hessian of the objective function during the identification process. The minimization procedure can

be based on an iterative method such as Gauss-Newton algorithm, with a quadratic rate of convergency. Accordingly, the parameter vector will be updated based on a recurrence relation

$$\boldsymbol{\Gamma}^{(r+1)} = \boldsymbol{\Gamma}^{(r)} - \kappa \left(\boldsymbol{H}^{-1}(\boldsymbol{\Gamma}^{(r)})\, \boldsymbol{g}^{T}(\boldsymbol{\Gamma}^{(r)})\right) \tag{3.10}$$

in which κ denotes the step size, $\boldsymbol{g}$ is the gradient vector, and $\boldsymbol{H}$ is the Hessian of the objective function. The second term in the right-hand side is used as the search direction in the algorithm whose components are estimated using the following definitions [4]. For gradient:

$$\boldsymbol{g}(\boldsymbol{\Gamma}) = \frac{\partial V}{\partial \boldsymbol{\Gamma}} = -\sum_{j=1}^{n} \left\{ \int_0^T \left(\xi_{exp}^{j} - \xi^{j}(\boldsymbol{\Gamma}, t)\right) \frac{\partial \xi^{j}}{\partial \boldsymbol{\Gamma}}\, \mathrm{d}t \right\} \tag{3.11}$$

and Hessian can be approximated as

$$\boldsymbol{H}(\boldsymbol{\Gamma}) = \frac{\partial^2 V}{\partial \boldsymbol{\Gamma}^2} \approx -\sum_{j=1}^{n} \left\{ \int_0^T \frac{\partial \xi^{j}}{\partial \boldsymbol{\Gamma}}^{T} \frac{\partial \xi^{j}}{\partial \boldsymbol{\Gamma}}\, \mathrm{d}t \right\} \tag{3.12}$$

Starting from unity for υ and a large value of K_i, the experimental data and simulated response match for any set of parameters to be identified [4]. At the start of each iteration, the value of υ will be decreased by a specified decrement, then using the optimized parameters of the previous step as initial guess for $\boldsymbol{\Gamma}$, the parameters for the current are calculated. This process will be continued until $\upsilon = 0$, for which the optimized parameters are the same as the original equations. It makes the optimized parameters for different values of υ lie within the neighborhood of the ones corresponding to the global minima of the modified mathematical models, when approaching to the original problem for which $\upsilon = 0$.

The parameters selected from the full-chemistry battery model are the ones that are difficult to be estimated or measured, which is essential in parameter identification study. Volume fraction of separator region (ϵ_s), Li^+ transference number in the electrolyte (t_+), electronic conductivity of solid phase of electrode n (σ_n), and initial electrolyte concentration in region s, n, and p (c_{e0}) are identified based on the reference results for the battery voltage. The physical significance of the parameters can be found in the battery model developed by Dao et al. [3]. To apply the homotopy optimization procedure in identifying the battery parameters, the modified Eq. (3.8) need to be solved during the optimization process.

Consistent preliminary provisions that placate the algebraic equations in the governing DEAS play a key role in the solutions accuracy. As a result, the solver parameters need to be intricately adjusted for the preliminary settings to be updated in every step of the optimization phase for various values of the model factors.

The optimization technique will only converge under these settings, with the adjusted parameters lying within a satisfactory range.

The aim is to ascertain certain battery model parameters while substantiating the variation between investigational and simulation results for the battery voltage to be below anticipated tolerance ε.

Simulation results contrasted with tests of continuous discharge current applied to the battery are shown in Fig. 3.7. The convergence trend of the objective function throughout the simulation process is shown in Fig. 3.8.

A high quality match denotes the efficiency of the homotopy optimization method in detecting the model parameters, which are effective in substantiating the PHEV simulation model.

The value for the identified parameters are listed in Table 3.4. The acquired values are in an adequate range for the battery model based on the projected parameters for the exhibited abridged model.

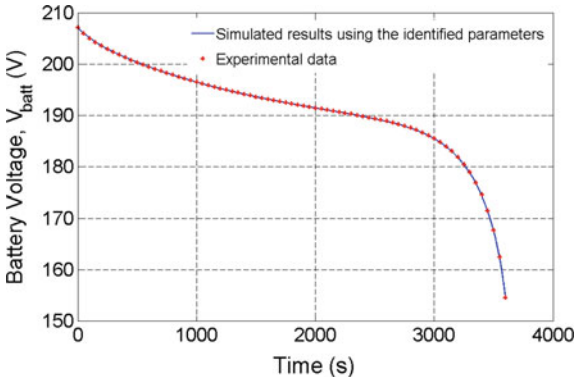


Fig. 3.7 Time history of the simulated battery voltage for experimental data and simulated response using the identified parameters

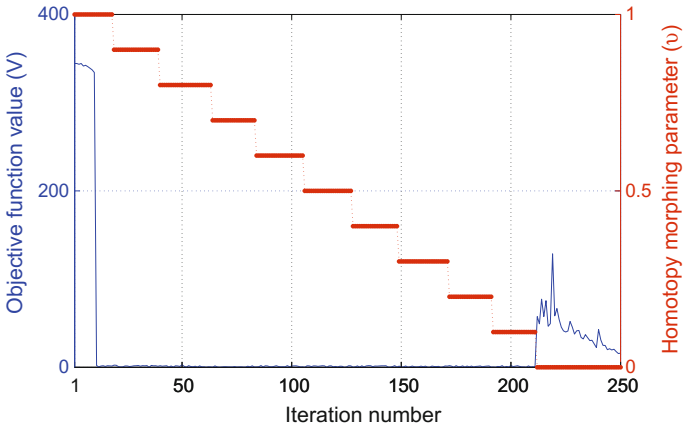


Fig. 3.8 Convergence trend of the objective function during the optimization process

Table 3.4 Identified values for the battery parameters

Parameter	Description	Value
ϵ_s	Volume fraction of separator region	0.55
t_+	Li^+ transference number in the electrolyte	0.21
σ_n	Electronic conductivity of solid phase of electrode n	101.2 (Sm^{-1})
c_{e0}	Initial electrolyte concentration in region s, n, and p	785.8 (mol m^{-3})

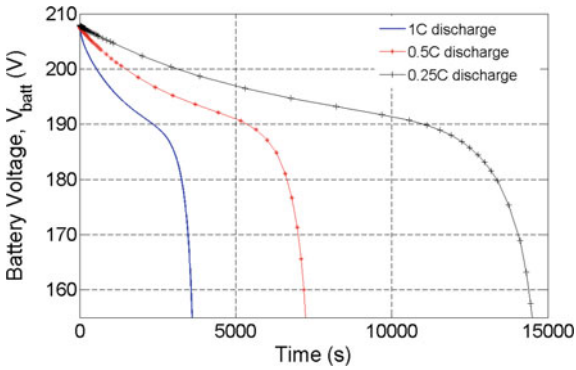


Fig. 3.9 Time history of the battery voltage for various discharge rates

The battery discharge voltage for a range of battery currents, together with the identified parameters are shown in Fig. 3.9, which implies an applicable dynamic performance of the simulated battery model.

3.4.4 Power-Split Device

The features associated with the geometry of the gear sets, including the amount of gear teeth and inertia, are attuned in accordance with the actual Toyota Prius plug-in powertrain.

3.4.5 Vehicle Model

Lastly, the entire authenticated powertrain model can be simulated in MapleSim and contrasted against the Autonomie model. The model simulation outcome based on charge depletion/charge sustenance (CDCS) approach along two consecutive UDDS drive cycles are shown in Fig. 3.10. Initially, the vehicle moves in pure electric mode with the battery discharged from a high level, followed by the control strategy attempting to maintain it as close to that level as possible when the battery state of charge (SOC) drops to a reference value (30).

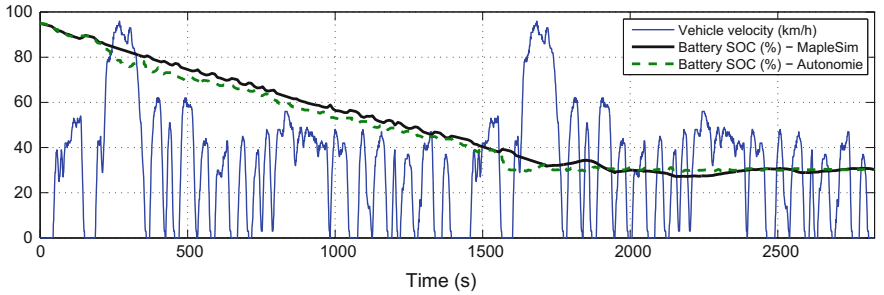


Fig. 3.10 Powertrain model in MapleSim and Autonomie: Simulation result along 2 UDDS drive cycles

3.5 High-Fidelity Model in Autonomie

The baseline PHEV powertrain includes an engine and two electric machines, MG-1 and MG-2, both of which can operate as a generator and as a motor. These components are coupled to the wheels through a power-split transmission system consisting of two planetary gear sets, as shown in Fig. 3.11.

The ring gears of both planetary gears are coupled to the wheels. The sun gear and planet carrier of the first planetary gear are coupled with MG-1 and the engine, whereas the sun gear and planet carrier of the second planetary gear are coupled to MG-2 and the chassis. In this configuration, the power-split transmission decouples

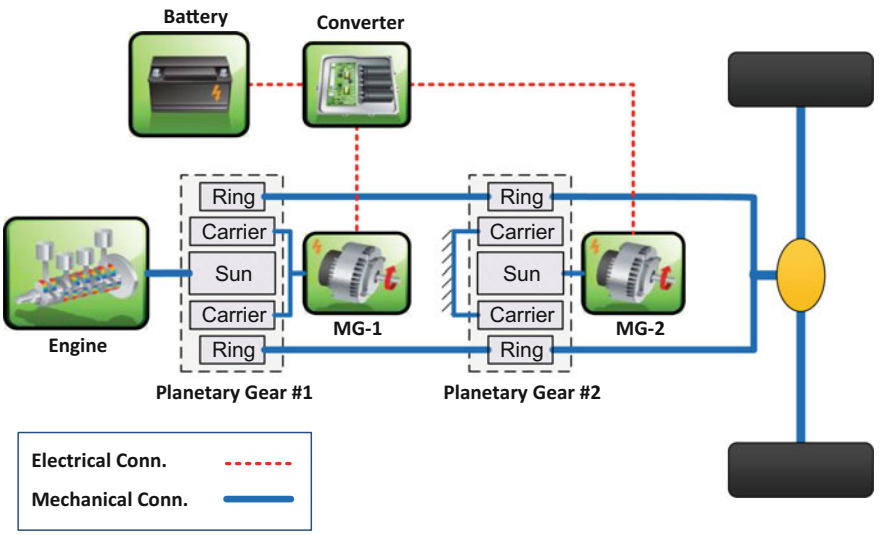


Fig. 3.11 Schematic of the power-split PHEV

the engine crankshaft from the road and allows the electric machines to adjust the engine's operation so that it functions at its maximum fuel efficiency point.

In a power-split system, the engine use can be minimized in low-efficiency operating conditions. For example, the engine is turned off when driving at low speeds and when the vehicle accelerates from a standstill. In these cases, electricity generated from the battery is the only power source used to operate the vehicle. While driving at normal speeds, the power-split system distributes the engine power in two ways. One power stream is used to drive the wheels, while the other is applied to MG-1, which operates as a generator. The electric power produced by the generator is converted to mechanical power in MG-2 and then transmitted to the final drive.

Control system design, tuning, testing, and validation activities typically incorporate several different types of model, which can be broadly divided into: high-fidelity, control-oriented, and online-optimization models. High-fidelity models are complex and describe the plant in detail, but are computationally expensive to run. Control-oriented or online-optimization models are sufficiently simple and fast for real-time implementation, and are accurate enough to characterize the system.

In order to assess the performance of the designed controller, the high-fidelity model of the baseline PHEV powertrain is created in the Autonomie software, a next generation of PSAT software developed by Aragon National Lab. The high-fidelity model is used for the assessment of a controllers performance through MIL and HIL analysis, but it can also be utilized for the parameter identification and model validation of control-oriented models.

A top level Simulink model of the PHEV, including the driver, vehicle powertrainarchitecture (VPA), and vehicle powertrain controller is shown in Fig. 3.12. The VPA, VPC, driver and environment blocks are interconnected via buses that contain information about the vehicle. The main info bus leaves the VPA and collects all the signals from the vehicle's powertrain systems (VPA). Then this main VPA info bus enters the VPC, driver and environment blocks. The environment blocks also send a bus, with all of the signals, into the VPC. The main VPA info bus and environment bus come together, along with input from the driver, before they enter the VPC, where they are used by the control strategy to adjust system behaviour. Afterwards, signals from all of the VPC subsystems enter the VPA [15].

3.5.1 Powertrain Model

Fig. 3.13 shows the powertrain configuration built in Autonomie. The main components of the powertrain are an internal combustion engine (Eng), two electric motors (more specifically, a traction motor (MG-2) and a generator (MG-1)), and a battery pack (B). Table 3.5 shows the characteristics of Toyota Prius Plug-in Hybrid.

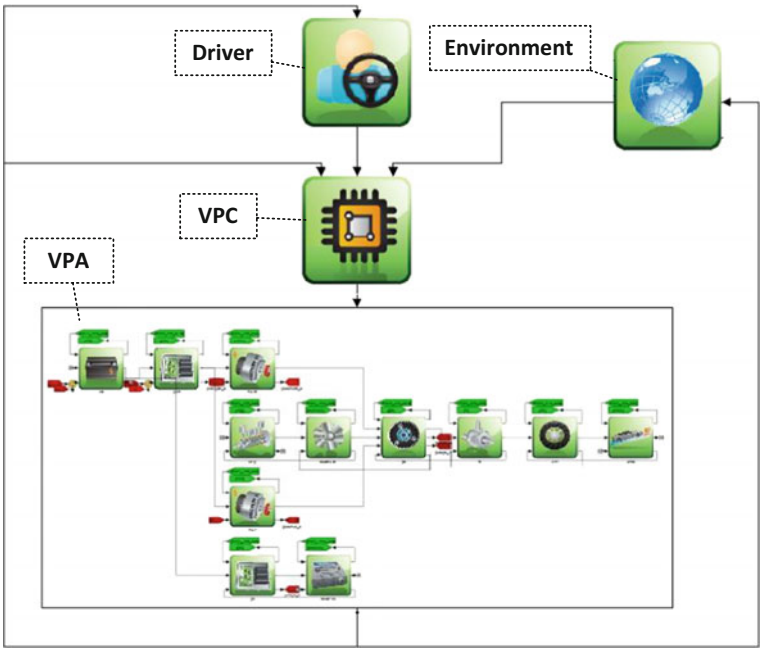


Fig. 3.12 Interconnection of blocks in the Autonomie model of the PHEV

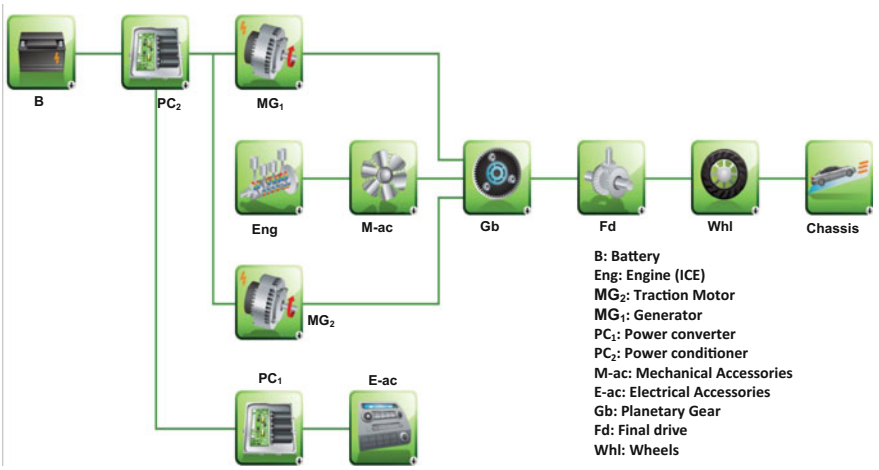


Fig. 3.13 Power-split PHEV powertrain configuration

Table 3.5 The characteristics of Toyota Prius plug-in hybrid

Parameter	Symbol	Unit	Value
Drag coefficient	C_D	–	0.26
Frontal area	A	m^2	2.25
Vehicle mass	m	kg	1525
Rolling resistance	f	–	0.008
Engine power	P_e	kW	73
Motor power	P_m	kW	50
Generator power	P_g	kW	30
Number of battery cells	N_b	–	56
Battery cell nominal voltage	V	V	3.7
Battery nominal capacity	Q	Ah	21
Wheel radius	r	m	0.3

3.5.1.1 Engine

The internal combustion engine was modeled using look-up tables based on the engine torque and speed. The ICE of Prius 2012 is a 1.8 L spark ignition engine with a maximum power of 73 kW. Figure 3.14 depicts the ICE efficiency maps for the Prius.

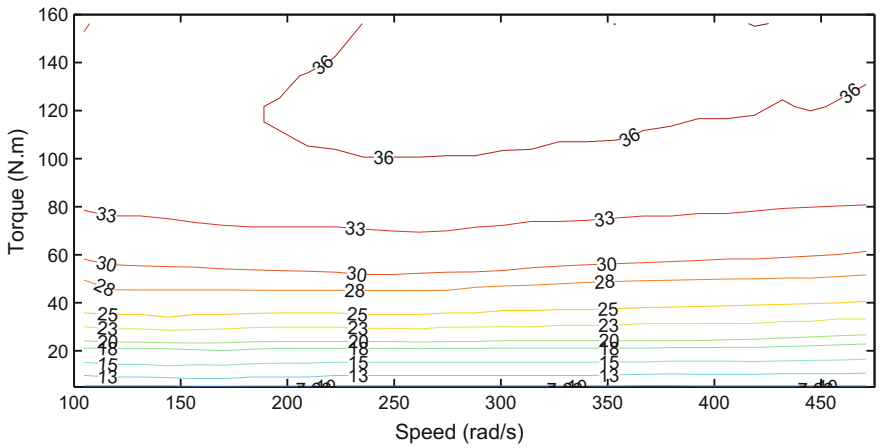


Fig. 3.14 Engine efficiency map

3.5.1.2 Electric Motor-Generators

The Prius is equipped with two permanent magnet electric motors. The efficiency of these electric motor-generators are represented by two look-up tables, as shown in Fig. 3.15.

3.5.1.3 Battery

The Lithium-ion battery pack is characterized using dual polarization circuit model. Battery variables such as open circuit voltage (V_{oc}), Internal resistance (R_b), and polarization variables (R_1, R_2, C_1, C_2) are obtained using look-up tables. The initial SOC and the minimum SOC are set to 0.9 and 0.2, respectively (Fig. 3.16).

3.5.2 Driver Model

The driver model block controls the speed of the vehicle. It contains of both feedforward and feedback controller. The feedforward controller is fast but not accurate. On the other hand, the feedback controller is more accurate but it needs feedback from the system that makes it relatively slow. The feedforward controller calculates desired torque based on the longitudinal dynamics (Eq. 3.13), and the feedback controller adjusts the desired torque using a PID controller.

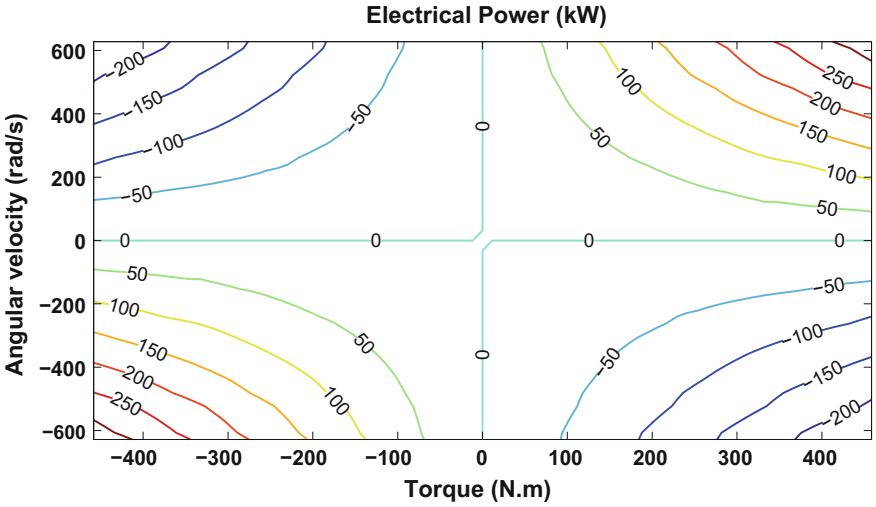


Fig. 3.15 Motor-generator electrical power map

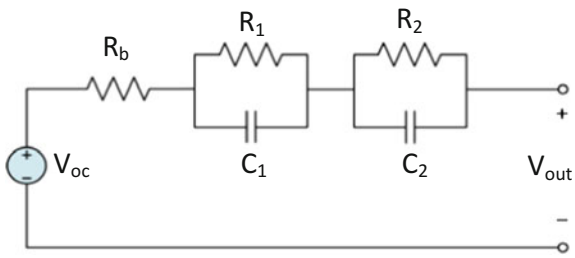


Fig. 3.16 Schematic of dual polarization battery model

$$T_{loss} = R \times (mg \sin\theta + \frac{1}{2}\rho A C_D v^2 + C_r mg \cos\theta + ma) \tag{3.13}$$

where R is the wheel radius, θ is the road grade, a is the vehicle acceleration, C_r and C_D are the rolling resistance and drag coefficients, respectively.

3.5.3 Powertrain Controller

The Autonomie rule-based EMS strategy is similar to the CDCS strategy. The only difference is that the rule-based EMS operates the engine in CS mode when the power demand is high. Based on this strategy, the engine would be turned on when the requested power is above a threshold or the battery SOC is lower than a threshold, or the electric motor cannot provide the requested wheel torque. First, the engine power command P_{eng} is calculated to determine the engine ON/OFF status.

$$P_{eng} = P_d + P_{e,b} \tag{3.14}$$

where P_d is the requested power, $P_{e,b}$ is the additional power to maintain the SOC of the battery during the CS operation. P_d is calculated from the vehicle longitudinal dynamics. $P_{e,b}$ is obtained from the Eq. 3.15.

$$P_{e,b} = k_p(SOC_{ref} - SOC) + \frac{k_i}{s}(SOC_{ref} - SOC) + P_0 \tag{3.15}$$

where, k_p and k_i are the proportional and integral gains, respectively. P_0 is obtained from the look-up table.

The controller determine the engine On/Off status using the engine power and SOC; i.e., if P_{eng} is more than threshold for a certain SOC, the engine is turned on and vice versa.

In the Low-level controllers (shown in Fig. 3.17), the torques of each powertrain component are computed based on torque demand and the optimum engine power. The angular velocity and torque of the engine are obtained from look-up tables based on optimum working point of the engine. Then, Generator torque (T_g) is controlled

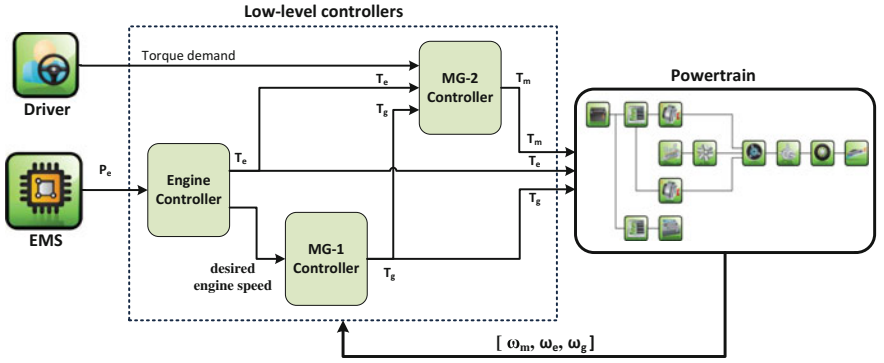


Fig. 3.17 Schematic of the low-level controllers

to track the optimum angular velocity of the engine. A PID controller adjust the electric motor torque in a way that wheel torque follow the desired torque which is calculated from driver model.

3.6 Summary

In this chapter, a high-fidelity simulation model of the Toyota Prius plug-in hybrid powertrain was developed in MapleSim. To do cross-validation of the MapleSim model, the Autonomie software has been used. Furthermore, the homotopy optimization approach was applied to identify the parameters of the chemistry-based Lithium-ion battery pack model. Since the symbolic computation power of MapleSim can reduce the simulation time significantly, the developed model can be used to conduct model and hardware-in-the-loop simulations. Therefore, by developing the MapleSim high-fidelity PHEV model, one can perform controls evaluation and validation, or come up with a simple and sufficiently accurate control-oriented model to be used for model-based controls design with more confidence. On the other hand, we introduced the PHEV model in Autonomie which is derived based on real-world testing so that we can use it for validating some of our designed controllers performance.

References

1. Newman, J., Tiedemann, W.: Porous-electrode theory with battery applications. AIChE J. **21**, 25–41 (2004)
2. Doyle, M., Fuller, T.F., Newman, J.: Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. J. Electrochem. Soc. **140**, 1526–1533 (1993)

3. Dao, T., Vyasarayani, C.P., McPhee, J.: Simplification and order reduction of lithium-ion battery model based on porous-electrode theory. J. Power Sour. **198**, 329–337 (2012)
4. Vyasarayani, C.P., Uchida, T., Carvalho, A., McPhee, J.: Parameter identification in dynamic systems using the homotopy optimization approach. Multibody Syst. Dyn. **26**, 411–424 (2011)
5. Kim, N., Rousseau, A., Rask, E.: Autonomie Model Validation with Test Data for 2010 Toyota Prius. SAE 2012-01-1040 (2012)
6. Rousseau, A., Sharer, P., Pasquier, M.: Validation Process of a HEV System Analysis Model: PSAT. SAE 2001-01-0953 (2001)
7. Mizuno, Y., Ibaraki, R., Kondo, K., Odaka, K., Watanabe, H., Mizutani, T., Kaneshige, K., Kitada, D.: Development of New Hybrid Transmission for Compact-Class Vehicles. SAE 2009-01-0726 (2009)
8. Burress, T.A., Campbell, S.L., Coomer, C.L., Ayers, C.W., Wereszczak, A.A., Cunningham, J.P., Marlino, L.D., Seiber, L.E., Lin, H.-T.: Evaluation of the 2010 Toyota Prius Hybrid Synergy Drive System. ORNL/TM-2010/253 (2011)
9. Grondin, O., Stobart, R., Chafouk, H., Maquet, J.: Modeling the Compression Ignition Engine for Control: Review and Future Trends. SAE 2004-01-0423 (2004)
10. Schulten, P.J.M., Stapersma, D.: Mean Value Modelling of the Gas Exchange of a 4-stroke Diesel Engine for Use in Powertrain Applications. SAE 2003-01-0219 (2003)
11. Kao, M., Moskwa, J.J.: Turbocharged diesel engine modeling for nonlinear engine control and state estimation. ASME J. Dyn. Syst. Measur. Control **117**, 20–30 (1995)
12. Kolmanovsky, I., Moraal, P., van Nieuwstadt, M., Stefanopoulou, A.: Issues in modelling control of intake flow in variable geometry turbocharged engines. In: Proceedings of 18th IFIP Conference on System Modeling and Optimization (1997)
13. Saeedi, M.: A Mean Value Internal Combustion Engine Model in MapleSim. M.Sc dissertation, University of Waterloo, Waterloo (ON) (2010)
14. Bogosyan, S., Gokasan, M., Goering, D.J.: A novel model validation and estimation approach for hybrid serial electric vehicles. IEEE Trans. Veh. Technol. **56**, 1485–1497 (2007)
15. Slezak, L.: Autonomie Training Part 1 Overview (2015)

Part I
Energy Management Approach

Chapter 4
Nonlinear Model Predictive Control

This chapter presents a model-based strategy for a PHEV with the use of MPC concept. MPC appears to be an appropriate scheme to utilize contemporary concept potentials and to satisfy the automotive requisites, as most can be defined in the form of a constrained multi-input, multi-output optimal control problem, and MPC provides approximate resolution to this class of problems [7].

The first part of the chapter will focus on addressing differing approaches to devise an energy management strategy for a PHEV based on different levels of trip information, and MPC is utilized to compose a strategy in the following cases: (1) No knowledge of trip information; (2) Known traveling distance; (3) Known entire trip information; and (4) Possibility of manually switching between electric mode and charge sustaining mode. Lastly, the outcome of weighting parameters inside the MPC cost function is examined together with evaluating fuel economy in differing cases based on UDDS cycle.

The second part of the chapter focuses on presenting a control-oriented model to design a novel supervisory controller for the PHE high-fidelity simulation model in addition to engine and electric motor low-level controls.

A sliding mode control method is considered for the engine low-level controller in order for the engine to follow the MPC-prescribed torque trajectory, while reducing hydrocarbon (HC) emissions resulting from multiple engine starts.

This controller is robust and can guarantee a good performance in the real-world experiment. To design this energy management scheme, both CDCS and blended mode strategies are investigated. Finally, the performance of the proposed energy management scheme is evaluated by applying it to the high-fidelity simulation model. A more reliable trade-off between fuel economy and emissions using a near-optimal energy management scheme for both CDCS and blended mode strategies can be obtained.

A. Taghavipour et al., *Intelligent Control of Connected Plug-in Hybrid Electric Vehicles*,
Advances in Industrial Control, https://doi.org/10.1007/978-3-030-00314-2_4

4.1 NMPC Energy Management Design

4.1.1 Theory of Model Predictive Control (MPC)

4.1.1.1 Problem Formulation

The design intent of MPC is to calculate the trajectory of an impending input to enhance the potential conduct of the plant output. The optimization is executed within a restricted timeframe based on the information of the plant at the start of the timeframe. The quadratic programming problem is defined if a quadrative objective function is utilized for the optimization. Certain requirements must be considered in order to guarantee the most optimal MPC performance. It is vital to have a suitably precise, but uncomplicated model that attains all characteristics of the original system behavior with a low computational effort. The subsequent step is to evaluate the quantifiable states and approximate the other states of the system. The final requirement is the implementation of prearranged actions such as the control unit.

The moving horizon window is a frequently used notation expression in MPC, which refers to the time interval in which the optimization is applied. The length of this window is referred to as the prediction horizon (N_p) and it governs the degree to which the future can be forecast. The prediction horizon length affects the optimal controller performance [37]. The purpose of MPC problem solving is to locate a vector containing the variation of inputs to reach the desired trajectory outputs. The length of this vector is referred to as the control horizon (N_c). Although the optimal trajectory of the impending control signal is fully depicted within the moving horizon frame, if the actual control input to the plant applies only the initial sample of the control signal and disregards the remainder of the trajectory, the principle is referred to as a receding horizon control scheme.

Information regarding the state variables at each point in time is required in the preparation phase in order to forecast the future. This information is represented by $x(t_i)$, which is a vector encompassing numerous applicable factors, and is either or is measured directly. A sound dynamic model provides an accurate prediction of the future [42].

In the meantime, an integrator is intuitively entrenched in the design, causing the predictive control system to track continuous references, while rejecting continual instabilities without steady-state errors. An additional major benefit of this method is that it necessitates neither the steady-state information about the control signals ($u(k) = u(k-1) + \Delta u(k)$), nor the steady-state information about the state variables in practice. For a linear MPC problem, the model inside the controller is amplified to contain an integrator for each output.

Consider the following augmented discrete system:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{4.1}$$

where x, u , and y are the state, input, and output variables of the linear system. The relation between the predicted output of the system inside the prediction window (Y), the time step k_i, the measured states at time t_i , and the designed variation of the inputs will be [42]:

$$Y = Fx(k_i) + \Phi \Delta U \tag{4.2}$$

where

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ . \\ . \\ . \\ CA^{N_p} \end{bmatrix} \quad \Phi = \begin{bmatrix} CB & 0 & ... & 0 \\ CAB & CB & ... & 0 \\ CA^2B & CAB & ... & 0 \\ . & . & ... & . \\ . & . & ... & . \\ . & . & ... & . \\ CA^{N_p-1}B & CA^{N_p-2}B & ... & CA^{N_p-N_c}B \end{bmatrix}$$

$$Y = \begin{bmatrix} y(k_i+1|k_i) \ ... \ y(k_i+N_p|k_i) \end{bmatrix}^T$$

$$\Delta U = \begin{bmatrix} \Delta u(k_i+1|k_i) \ ... \ \Delta u(k_i+N_c|k_i) \end{bmatrix}^T$$

where $y(k_i + 2 \mid k_i)$ means the predicted output on step $k_i + 2$ based on the measurement from step k_i and $\Delta u(k_i)$ is the variation of control input in that time step [42].

The functioning of a control system may weaken considerably when the control signals for the original design meet with operational limitations. Nonetheless, through a minor alteration, the degree of performance deterioration may be diminished if the constrictions are integrated in the execution, which leads to the notion of constrained control. All constraints must be written in the form of variation input signal in order to adjust the controller. The ensuing relation can be written for the constraints on the amplitude of the inputs, the variation of the inputs, and the outputs:

$$\begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} \Delta U \leq \begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix} \tag{4.3}$$

where

$$M_1 = \begin{bmatrix} -C_2 \\ C_2 \end{bmatrix}; N_1 = \begin{bmatrix} -U^{min} + C_1 u(k_i - 1) \\ U^{max} - C_1 u(k_i - 1) \end{bmatrix}$$

$$M_2 = \begin{bmatrix} -I \\ I \end{bmatrix}; N_2 = \begin{bmatrix} -\Delta U^{min} \\ \Delta U^{max} \end{bmatrix}$$

$$M_3 = \begin{bmatrix} -\Phi \\ \Phi \end{bmatrix}; N_3 = \begin{bmatrix} -Y^{min} + Fx(k_i) \\ Y^{max} - Fx(k_i) \end{bmatrix}$$

In Eq. (4.3), I and 0 are identity and zero matrices of size $N_c \times N_c$, Y^{min}, Y^{max} are vectors containing the minimum and maximum of all predicted outputs, U^{min}, U^{max} are the minimum and maximum of all inputs, and ΔU^{min}, ΔU^{max} are the minimum and maximum allowable variations for all inputs.

As previously mentioned, the existing optimization problem can be converted into a quadratic form. Assume that the cost function is written as follows:

$$J = \frac{1}{2}\Delta U^T H \Delta U + \Delta U^T E$$
$$M\Delta U \leq N \tag{4.4}$$

where M and N are specified by the constraints of (4.3), and H, E are matrices derived based on the cost function. Note that the inputs for the MPC problem are the inputs variations over the length of the control horizon. A typical solution to this problem using Lagrangian multipliers can be found [9]:

$$\Delta U = -H^{-1}E - H^{-1}M^T\lambda \tag{4.5}$$

where $\lambda = -(MH^{-1}M^T)^{-1}(N + MH^{-1}E)$.

A rapid approach is necessary, since this problem must be resolved in each time step, and characterizing active limitations in each time step aids in speeding up the calculation process. A technique that implies an iterative approach to identifying the active constraints to solve the problem and find the second term in Eq. (4.5) is used in this research.

As it is applied to a system, a well-designed controller must be stable and have a suitable performance. As laid out in the literature, model predictive control display implying performance and the design considerations can be adjusted to evade controller instability.

4.1.1.2 Stability Analysis

In this section, some theories for MPC stability issues are reviewed. MPC can easily destabilize the system if short prediction horizon is chosen. To avoid this situation, we can consider longer (even infinite) prediction horizons which adversely leads to higher computational effort. Another way to guarantee the stability for any length of prediction horizon is by considering terminal constraints to ensure that the system states converge to definite values at the end of the prediction horizon [12]. Though easiest to handle from the viewpoint of mathematical analysis, a significant drawback of this approach is that the terminal equality constraint can be quite severe; it is hard to satisfy and artificially imposing such a strict constraint can lead to substantial performance loss. For example, for such an approach to work, the underlying system needs to be reachable, instead of just being stabilizable [18]. However, adding constraints might make the optimization infeasible. To overcome this problem, the constraint can be relaxed so that the states of the system converge to a region rather

than specific points. In other words, the terminal constraint points can be changed to terminal constraint sets. Chen and Allgower [4] presented an approach called quasi-infinite horizon MPC, where a quadratic terminal penalty corresponding to the infinite horizon cost of the linearized system is imposed. Because a terminal constraint is used to force the state to lie within a prescribed terminal region, within which the system is stabilized by the linear feedback, feasibility alone implies the asymptotic stability.

Another possible approach is to penalize the terminal constraint points inside the cost function. It has been known for some time (e.g., [1]) that making the horizons infinite in predictive control leads to guaranteed stability, but it was not known how to handle constraints with infinite horizons. The key idea is to re-parameterize the predictive control problem with infinite horizons in terms of a finite number of parameters (instead of the infinite number of control decisions), so that the optimization can still be performed over a finite-dimensional space—in fact, it remains a quadratic programming (QP) problem. In [5], the opinion has been expressed that there is no longer any reason to use finite horizons—at least with linear models. In the case of changing the horizon length to infinity, it is necessary to reformulate the problem for both stable and unstable systems. The essential difference in the latter case is that the unstable modes must be driven to 0 within N_c steps; otherwise, these modes, which are uncontrolled, would become unbounded and result in an infinite cost value.

In [1, 23], it is shown that stability can sometimes be guaranteed with finite horizons even when there is no explicit terminal constraint. The finite horizon predictive control problem is associated with a time-varying Riccati difference equation, which is intimately related to the optimal value of the cost function. The Fake Algebraic Riccati Technique replaces this equation with an algebraic (time-invariant) Riccati equation that resembles that found in infinite-horizon LQ problems.

Besides the approaches using terminal penalty constraints, other approaches have been proposed. One notable alternative approach employs a contraction constraint, that requires the size of the state to shrink over the prediction horizon [27]. More generally, it chooses a positive-definite function of the state and requires this function to decrease over time in the optimization. To ensure feasibility, the size of the prediction horizon is not set a priori but is treated as an optimization variable. The whole computed input sequence can be implemented in open loop until the end of the horizon, as originally suggested, or the optimization can be repeated after some time as suggested in [6]. Extending the MPC formulation for constrained linear systems to nonlinear systems is conceptually straightforward but met with practical difficulties. Most of the stability results for the constrained linear systems apply to nonlinear systems without modification. In fact, many of the earlier stability results for constrained optimal control [12, 22] were developed in the context of a general nonlinear system. However, the implementation is greatly complicated by the computational complexity in finding a globally optimal solution to a non-convex optimization problem. Computational complexity remains as a major obstacle for designing a practically implementable nonlinear MPC algorithm with guaranteed stability. Naturally, the researchers focused on finding a formulation that does not require a globally optimal solution to be found, just a feasible solution. In Mayne

and Michalska [22], once a feasible solution is found, the subsequent calculation preserves the feasibility and tries to merely improve the cost.

However, [45] shows that the closed-loop system with MPC controller is globally asymptotically stable if and only if the optimization problem is feasible.

Therefore, one can prove the stability of MPC by ensuring that the solution is always feasible and the constraints are satisfied all the time.

4.1.1.3 Robustness Analysis

MPC, being a feedback control method, has some inherent robustness, which was analyzed by several researchers [28]. Nonetheless, when a quantitative description of the model uncertainty is available, it may be beneficial to consider all possible future trajectories under the given uncertainty description in the optimal control calculation. Lee and Yu [19] presented an argument indicating the deficiency of the open-loop formulation and presented an alternative formulation based on dynamic programming. With some modifications, they were able to formulate a MPC algorithm that solves a convex program at each time and guarantees robust stability. However, such an approach cannot be implemented directly since the possible control laws do not yield a finite-dimensional parametrization. Kothare, Balakrishnan and Morari [14] presented an interesting formulation where the minimization at each sample time searched overall linear state feedback laws to minimize the worst-case error. The problem was formulated as a linear matrix inequality (LMI), which is convex and can be solved through semi-definite programming. However, MPC integrates performance of optimal control with the robustness of feedback control.

4.1.2 NMPC Performance on the Low-Fidelity Powertrain Model

MPC approach is capable of near-optimal control of HEV powertrains. In this section, different MPC energy management strategies for a PHEV are evaluated, with different levels of trip information available to the controller, based on fuel economy.

4.1.2.1 Control-Oriented Model

In order to design MPC EMS, a simple and sufficiently accurate model of a PHEV powertrain is needed, which is called a control-oriented model. Among the different architectures for a HEV, the power-split configuration seems to be the most efficient one for a limited capacity of battery [25]. In a power-split configuration, the engine, the electric motor, and the generator are connected to each other by means of two planetary gear sets.

For deriving the dynamics of the system, it is assumed that the mass of the pinion gears is small, there is no friction, no tire slip, nor efficiency loss in powertrain. By considering the vehicle longitudinal dynamics and an internal resistance model for the battery, the equation of the system will be written as (4.6)–(4.8):

$$\begin{aligned}(\frac{I_v^{'}(s_1+r_1)^2}{r_1 I_e^{'} K}+\frac{I_v^{'} s_1^2}{r_1 I_g^{'} K}+r_1)(\frac{r_2}{s_2})\dot{\omega}_m &= (\frac{(s_1+r_1)^2}{r_1 I_e^{'}}+\frac{s_1^2}{r_1 I_g^{'}})(\frac{s_2}{r_2})T_m \\ +(\frac{s_1+r_1}{I_e^{'}})T_e+(\frac{s_1}{I_g^{'}})T_g &- (\frac{(s_1+r_1)^2}{r_1 I_e^{'} K}+\frac{s_1^2}{r_1 I_g^{'} K})T_d \end{aligned} \tag{4.6}$$

$$\begin{aligned}(\frac{I_e^{'} r_1^2 K}{(r_1+s_1) I_v^{'}}+\frac{I_e^{'} s_1^2}{(r_1+s_1) I_g^{'}}+r_1+s_1)\dot{\omega}_e &= -(\frac{r_1}{I_v^{'}})T_d \\ (\frac{r_1^2 K}{(s_1+r_1) I_v^{'}}+\frac{s_1^2}{(r_1+s_1) I_g^{'}})T_e+(\frac{RK}{I_v^{'}})T_m &- (\frac{s_1}{I_g^{'}})T_g \end{aligned} \tag{4.7}$$

$$S\dot{O}C = -\frac{i_{batt}}{Q_{batt}} = -\frac{V_{oc}-\sqrt{V_{oc}^2-4(T_m\omega_m\eta_m^{-k}-T_g\omega_g\eta_g^k)R_{batt}}}{2R_{batt}Q_{batt}} \tag{4.8}$$

It should be noted that the parameters are adjusted according to a Toyota Prius plug-in hybrid (see Appendix A). In this system, there are three state variables: ring speed (ω_r) which is proportional to the vehicle velocity, engine speed (ω_e), and battery state of charge (SOC). There are three inputs: Engine torque (T_e), Motor torque (T_m), and Generator Torque (T_g). η_m and η_g represents motor and generator efficiency, respectively, including DC/DC converter and DC/AC inverter efficiencies. T_d is the driver's demanded torque. When the battery is discharged $k = 1$, and $k = -1$ for battery charging. The aforementioned system is considered as the control-oriented model.

4.1.2.2 MPC Formulation for a PHEV

There are three inputs that give flexibility to the control problem. In fact, two of these inputs are independent and the third one can be found through system dynamics. In each prediction window, a cost function needs to be minimized that results in maximum fuel economy and tracking a predefined level of battery charge while following a drive cycle. The cost function is:

$$J(k) = \sum_{i=1}^{N_p}\left[w_1(SOC_{ref}(k+i)-SOC(k+i))^2+w_2(\dot{m}(k+i))^2\right] \tag{4.9}$$

The first term is related to keeping the state of charge around a predefined reference. The second term is intended to minimize the fuel consumption. w_1 and w_2 are weighting parameters that are chosen according to the predicted maximum value of the weighted variables. The effect of these weighting parameters on fuel consump-

tion will be investigated later in this chapter. Fuel economy is estimated according to UDDS drive cycle. Also, there are some constraints on this problem that are defined as follows:

$$\begin{aligned} T_{min-e} < T_e < T_{max-e} \\ T_{min-m} < T_m < T_{max-m} \\ T_{min-g} < T_g < T_{max-g} \\ \omega_{min-e} < \omega_e < \omega_{max-e} \\ \omega_{min-r} < \omega_r < \omega_{max-r} \\ \omega_{min-g} < \omega_g < \omega_{max-g} \\ SOC_{min} < SOC < SOC_{max} \end{aligned}$$

To obtain a simpler form of the controller and also utilizing the linear MPC, the equations of the system are linearized for each time step around the operating point. Moreover, the fuel consumption map of the engine is projected as:

$$\dot{m}_f = \bar{\alpha}\omega_e^2 + \bar{\beta}T_e\omega_e \tag{4.10}$$

where $\bar{\alpha}$ and $\bar{\beta}$ are constants [26].

This leads to the evaluation of different potential control schemes based on the trip information, by verifying an appropriate reference SOC trajectory. This reference SOC is plugged into (4.9) and a quadratic programming problem is solved for each case as follows.

4.1.2.3 No Knowledge of Trip Information

In the event that trip information is missing, the most optimal strategy is to obtain use of the vehicle full electric range at early stages of driving. In this control strategy, the battery energy is utilized until the SOC grasps a predetermined level (charge depletion), and it is independent of the driving cycle and driving distance or any other information like initial SOC [21]. In fact, the engine may be started in charge depletion mode at points, where the necessitated power surpasses what the battery and the motor can provide. The strategy enters a loop run by MPC when the SOC drops deteriorates to the lower limit. This controller attempts to maintain the state of charge near the reference and concurrently minimizes the fuel consumption (charge sustenance). Here, the lower limit is assumed to be 0.3 because of the battery health factors. It should be noted that charge depletion/charge sustenance (CDCS) strategy might be very useful to extend the battery life cycle [24].

4.1.2.4 Known Traveling Distance

In this instance, knowledge of traveling distance to the next charging station exists. The most optimal strategy is proceeding in pure electric mode, if the traveling distance is less than the vehicle all-electric range. As stated earlier, postponing the charge sustaining stage would improve fuel economy. As such, it may be assumed that the battery SOC is linearly decreased with the vehicle traveled distance according to the following relation [46]:

$$SOC_{ref} = \frac{SOC_{high} - SOC_{low}}{Distance_{total}} X + SOC_{low} \tag{4.11}$$

where SOC_{high}, SOC_{low} and X are initial battery state of charge, the lowest possible charge level of battery, and distance traveled.

In order to apply this strategy, the linear trajectory must be plugged into the cost function, and MPC reduces fuel consumption while making state of charge follow the reference.

4.1.2.5 Known Entire Trip Information

In this section, it is presumed that access to some vehicle velocity information determined by road traffic through GPS is available. The objective is to obtain an optimized SOC trajectory that can be determined based on the longitudinal model of the PHEV to reduce fuel consumption up to the next charging station (see [39]). A curve is fit to the vehicle velocity data to reach this goal, as shown in Fig. 4.1. The velocity schedule is then allocated into appropriate segments. The power required in each segment can be calculated based on the initial and final velocities in each segment and traveling time. At this moment, an optimization parameter (PR) defined as in Eq. (4.12) is introduced.

$$P_{mot} = PR \times P_{dem} \tag{4.12}$$

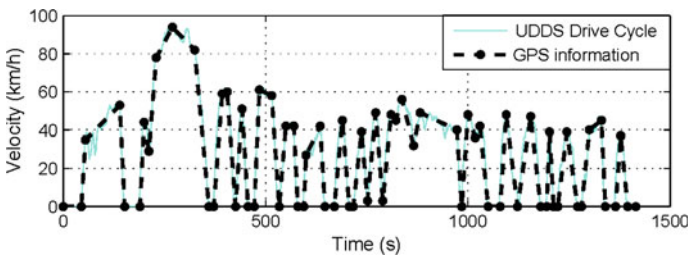


Fig. 4.1 GPS information and segments for one UDDS drive cycle

where P_{dem} and P_{mot} are demanded and motor power, respectively. PR determines the contribution of each power source in drivetrain to provide propulsion power. The optimization parameters are electric power ratios (PR_i), and the cost function is the total fuel consumption which is the summation of fuel consumption in each segment (m_{f_i}).

$$\min_{PR_1, PR_2, \ldots, PR_n} (m_{f_1}, m_{f_2}, \ldots, m_{f_n}) \tag{4.13}$$

Based on the motor speed, that is proportional to the vehicle velocity, the motor torque can be obtained and plugged into Eqs. (4.6) and (4.7).

Some constraints on the problem are considered (like SOC value at the end of the trip) as a nonlinear constraint which is a function of electric power ratio (PR) that could be calculated according to each segment SOC depleting from Eq. (4.8).

It is important to note that segment considerations will enhance the forecast for an optimized SOC trajectory. Nonetheless, computational effort is significantly amplified, making for a compromise between number of segments and fuel economy enhancement.

4.1.2.6 Manual Charge Depletion Charge Sustenance

This strategy appears to be a more rational means to lower fuel consumption, however, its effectiveness is restricted and cannot be compared to the blended mode approach. The driver is able to manually switch between full electric and charge sustenance (SC) mode in this scheme, and is able to find the interchanging occasion based on GPS information, or on his/her experience and road traffic comprehension. As per the engine efficiency, it is more advantageous to drive in full electric mode in urban settings with regular stops and starts, while it is suggested that highway driving is conducted in charge sustenance mode. It is evident that interchanging is feasible when there is sufficient energy stored in the battery; however, switching is no longer possible when the state of charge drops to the lowest permitted threshold. It is at this stage that we attempt to divide the UDDS drive cycle into certain sections where the driver is prepared to switch between modes. There is one micro cycle ($180 < t < 360$) in the UDDS schedule with higher average speed, and it is assumed that this occurs while driving on the highway. The driver switches to CS mode in this section, but the vehicle is driven in EV mode for the rest of the drive cycle, while there are numerous stop and start occurrences, or even micro cycles with reduced average velocity ahead, prior to the SOC drops to its lowest allowed threshold.

The ability to propose difference schedules for interchanging between two modes is evident; however, an attempt is made to demonstrate here that this simple strategy results in substantial improvement in fuel economy without committing to any further complex optimization method. Furthermore, GPS data can be used as a guideline for the driver to identify the appropriate time for interchanging between two modes.

4.1.2.7 Simulation Results

The simulation was completed in the MATLAB environment with the required torque calculated based on the UDDS drive cycle. This torque functions as one of the inputs to the controller. EMS utilizes this input and a linearized model of the powertrain to forecast the future input of each power source on board. Outputs of the controller are applied to the nonlinear model of the powertrain (Eqs. (4.6)–(4.8)) to find the system state variables such as vehicle velocity, battery level of charge, and fuel consumption in particular. Results of simulation are presented in Figs. 4.2, 4.3, 4.4, and 4.5. Figure 4.6 depicts fuel economy for different levels of trip information, for different values of the ratio of weighting parameters ($\frac{w_2}{w_1}$) introduced in Eq. (4.9). Readers may refer to [37] for a study on prediction and control horizon length effects on fuel consumption. It is important to note that the battery delivers the full power required to propel the vehicle while in pure electric driving mode. Consequently, if the requisite propulsion power does not surpass the provision limits of the electric motor or the battery, running the MPC controller is not required.

Control input plots are exhibited in each case to demonstrate that they meet the constrictions of the problem.

As shown in Fig. 4.2a, the battery is fully charged at the commencement of the drive cycle, the vehicle is in pure electric mode and the engine is off (Fig. 4.2b), until the charge decreases to the predefined reference state of charge of 0.3. Fuel consumption in this phase is zero. At this time, the main controller switches to the MPC mode and attempts to retain the state of charge as close to the reference as

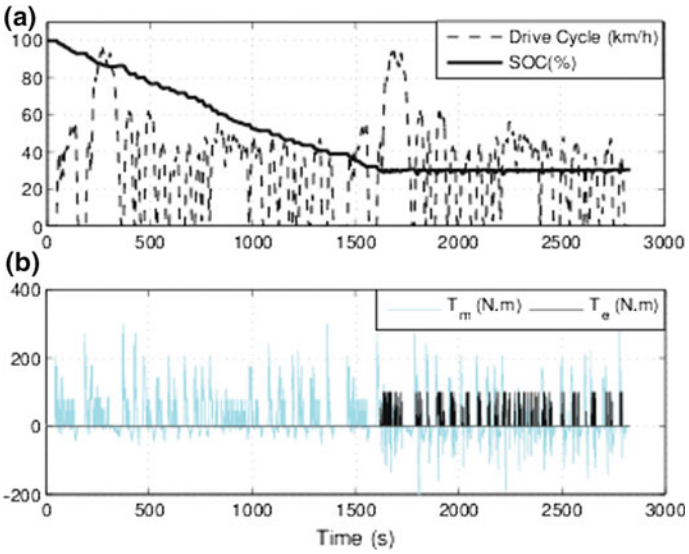


Fig. 4.2 Charge depletion/charge sustenance strategy: **a** SOC along two UDDS drive cycles **b** Power source torques

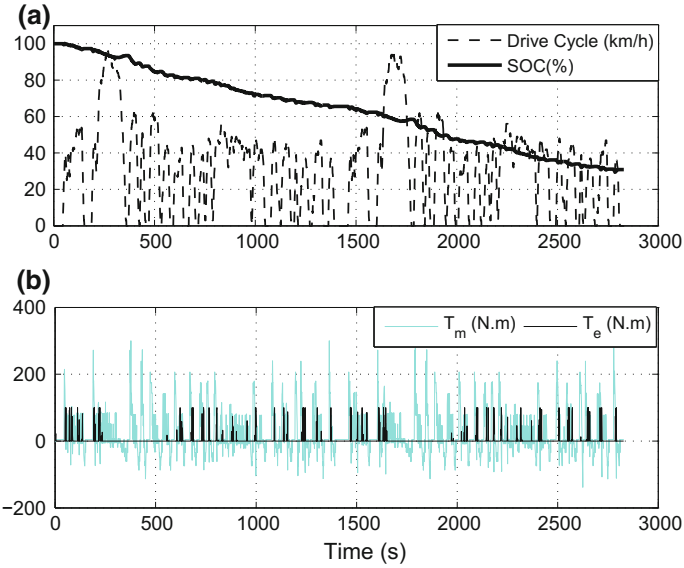


Fig. 4.3 Blended mode strategy (known traveling distance): **a** SOC along two UDDS drive cycles **b** Power source torques

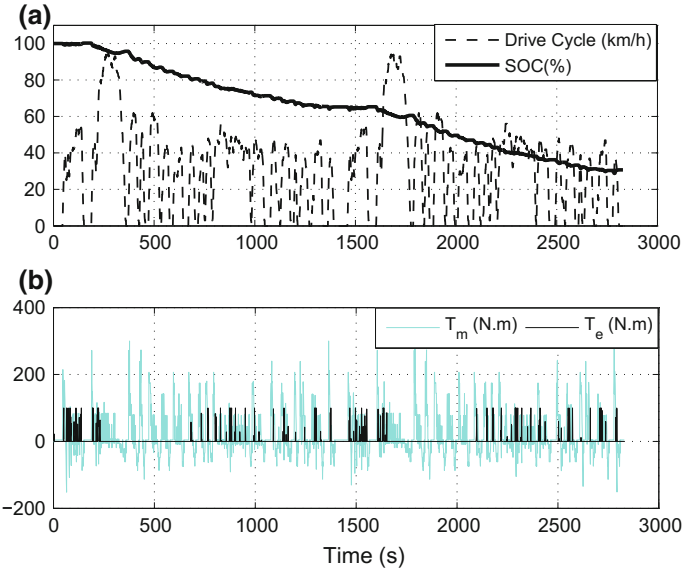


Fig. 4.4 Blended mode strategy (entire trip information): **a** SOC along two UDDS drive cycles **b** Power source torques

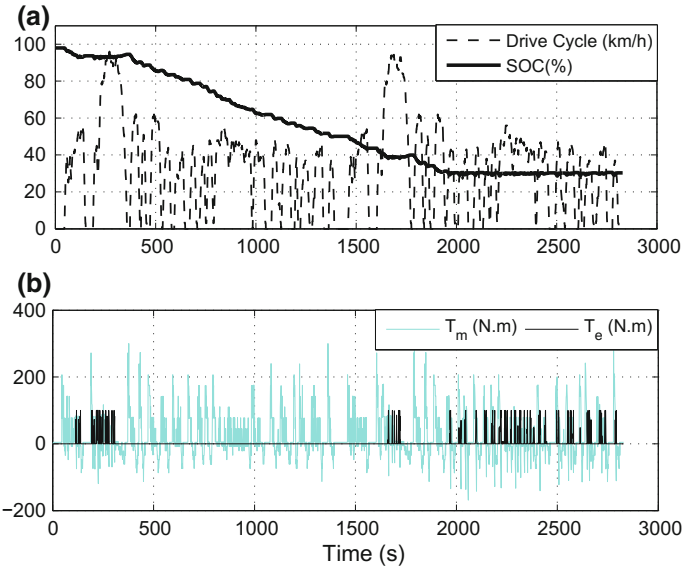


Fig. 4.5 Manually switching between EV and CS: **a** SOC along two UDDS drive cycles **b** Power source torques

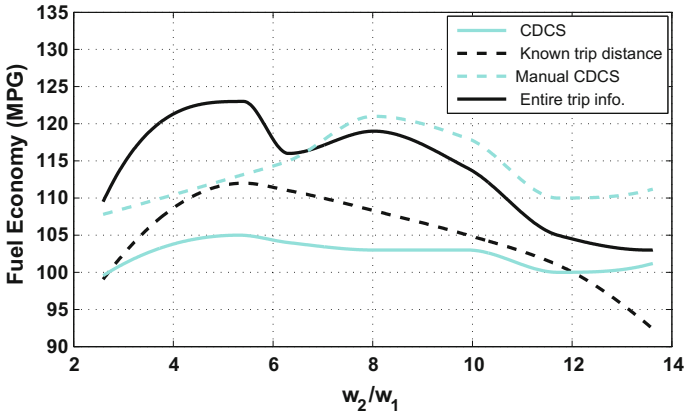


Fig. 4.6 Fuel economy versus ratio of weighting parameters in the cost function

possible, while curtailing fuel consumption. Fuel consumption, in this case, is 105 MPG. (This study references the vehicles fuel economy in miles per gallon (MPG)). Nonetheless, EPA considers miles per gallon gasoline equivalent (MPGe) as a gage of average distance traveled per unit of energy consumed for alternative fuel vehicles, PHEVs and EVs. In this rating, 33.7 kWh of electricity is equivalent to one gallon of gasoline.

In Fig. 4.3a, SOC follows the linear profile versus traveled distance and the engine operation is distributed along entire drive cycle (Fig. 4.3b), which results in 112 MPG. Therefore, the known traveling distance will improve fuel economy by 6.86 %.

In Fig. 4.4a, SOC follows the optimized trajectory. Here, we have better fuel economy since we have access to more information about the trip (9.66% improvement in comparison to known traveling distance case).

Figure 4.5a shows the result of simulation for manually switching between EV and CS mode. First, the vehicle goes in pure electric mode. By getting closer to the high-speed micro cycle, the driver switches to CS mode and MPC keeps the SOC constant around SOC value at the beginning of this micro cycle. After high-speed micro cycle, the vehicle goes in pure electric mode where we have a quick battery depletion and the engine is off (shown in Fig. 4.5b). At $t = 1600\,s$, again we have a high-speed stage and driver switches to CS mode. This time, the state of charge is lower and closer to the reference, so it is obvious that the vehicle can travel less in pure electric mode. Finally, at $t = 1967\,s$, the vehicle goes in CS mode when it is not possible to switch to EV mode any longer.

The performance of the MPC controller is contingent on weighting parameters inside the cost function, as well as predication and control horizon length. The effect of prediction and control horizon values is investigated in [37]. As illustrated in Fig. 4.6 and in comparison to other two strategies, CDCS strategy and the blended mode strategy for known trip distance result in a reduction in fuel economy. When compared with blended with blended mode with entire trip information, manually interchanging EV and CS implies a close or much more improved fuel economy for various weighting parameter values. Nevertheless, the maximum feasible fuel economy reachable through using complete trip information is greater than other strategies The maximum fuel economy for various control strategies is assessed in Table 4.1. Better fuel economy is anticipated by decreasing w_2/w_1, but due to the limitations, particularly sustaining drivability, this enhancement is not reflected in fuel economy for some of w_2/w_1 values.

It took 35.4 s in real time for 2828 s of simulation for two consecutive UDDS drive cycles to be computed. The simulation was carried out on a machine powered by a 3.16 GHz dual-core CPU and 4 GB memory. A faster result is possible with a controller implemented via C-code. In addition, on the same machine, it took 20.2 s in real time for attaining the optimized SOC trajectory based on the trip information.

Table 4.1 Fuel economy for different control strategies

Control strategy	MPG
Charge depletion/Charge sustenance	105
Blended mode (known traveling distance)	112
Blended mode (entire trip information)	123
Manually switching between EV and CS mode	121

4.1.3 NMPC Performance Benchmarking

In order to contrast MPC results, a dynamic programming (DP) problem was solved with the same dynamic and constraints. DP is a widely employed scheme in the literature for obtaining a global solution for HEVs control strategies. DP is simply a benchmark for developing heuristic strategies, since it cannot be implemented online.

Through the detection of all potential control options, the DP method guarantees a global optimal solution in the minimum fuel consumption problem of HEVs.

Figures 4.7, 4.8, 4.9, and 4.10 illustrate the results of DP in charge sustaining mode. As per Fig. 4.9, the average speed of the engine is more than it is in MPC, bringing the operating points closer to the engine sweet spot. As such, the consequential fuel consumption is 326 g (123 MPG), although the engine does not cease to operate. The fuel consumption for MPC without considering input variation inside MPC cost function while $SOC_{ref} = 0.3$ is 381 g (105 MPG). In comparison to the DP model, a 14.4.

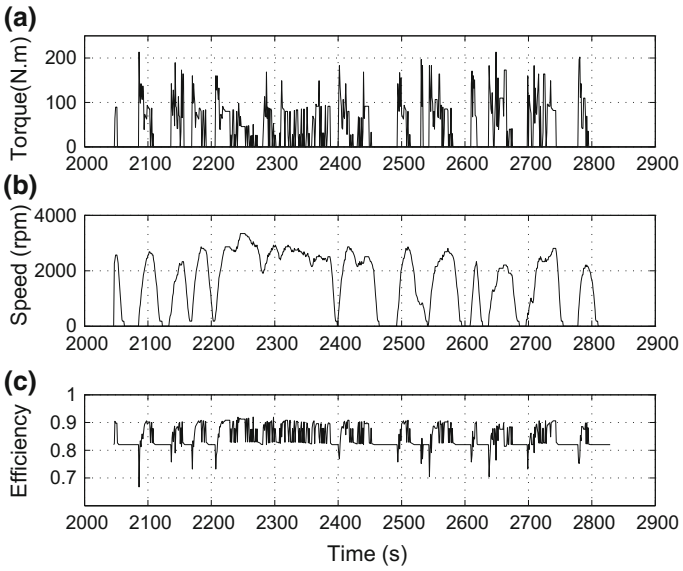


Fig. 4.7 DP result: **a** MG2 torque **b** MG2 speed **c** MG2 efficiency

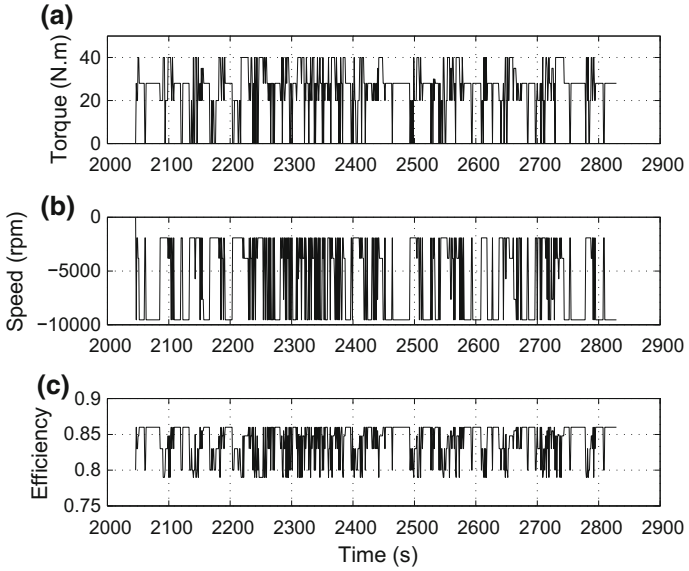


Fig. 4.8 DP result: **a** MG1 torque **b** MG1 speed **c** MG1 efficiency

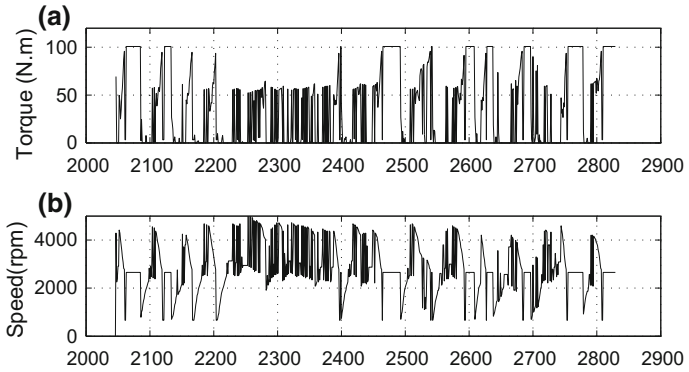


Fig. 4.9 DP result: **a** Engine torque **b** Engine speed

4.1.4 NMPC Performance on the High-Fidelity Powertrain Model

This section, an EMS based on MPC is developed and applied to the high-fidelity simulation model presented in Chap. 3. Low-level controllers are for the engine and electrical dive in order to apply the EMS to the MapleSim model. Engine emission reduction is considered as an objective for designing low-level controls.

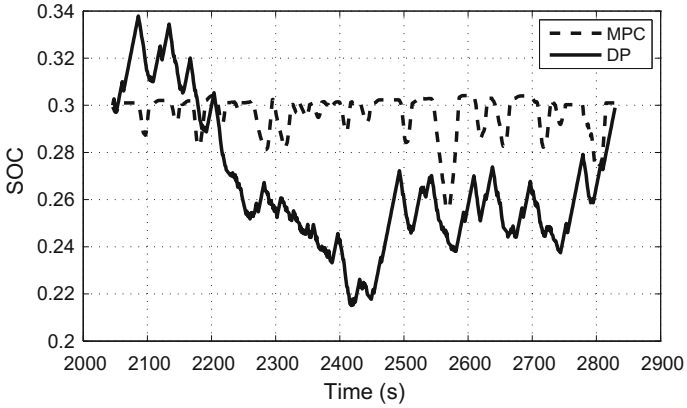


Fig. 4.10 SOC comparison for DP and MPC in charge sustenance mode

As previously stated, PHEVs have the potential for substantial fuel consumption decline, but likely at the expense of amplified tailpipe emissions caused by multiple cold starts and improper use of the engine for PHEV specific operation [34]. Due to the frequent and extensive engine shut down and catalyst cool down, the catalyst temperature management for reduced tailpipe emissions is a complicated problem [15].

Relevant literature and research have addressed the various approaches to the emissions issue. In addition to the catalyst, a separate hardware can potentially be used to resolve the various issues with different engine start events. In [40], the authors investigated the addition of hydrocarbon (HC) absorber traps and activated carbon fiber canister in a traditional exhaust aftertreatment system. These traps can store HC temporarily and release it after the temperature reaches the light-off temperature of exhaust aftertreatment device. Therefore, the HC will be catalyzed by exhaust aftertreatment even at low temperatures. The same happens to NOx that is translated to N_2.

In order to depict the heat/mass transfer in the aftertreatment system and conversion efficiency as a function of the catalyst brick temperature and air-to-fuel ratio [3, 43], the research explored models for a three-way catalytic converter (TWC). These models are overly complex for control algorithm development, and are mainly used for design and assessment. Specific model-based emissions control for the engine are possible to design using simpler models of catalyst conversion efficiency. For example, [35] proposed a simplified model of an internal combustion engine to derive a sliding mode control law for emissions reduction. In [33], the authors utilized a PMP approach to perform real-time optimal control of cold start via an experimentally verified control-oriented model of the engine.

Reducing fuel consumption is the sole objective in the majority of optimization problems for HEVs, with emissions reduction is deemed a process constraint; as long as emissions are within predefined limits, it does not influence the optimization

process [2]. Recent HEV research studied emission reduction as part of the control objective and investigated the trade-off between emissions and fuel economy [13, 20]. Instead of tailpipe emissions, the majority of these studies examined fuel consumption minimization and engine-out emissions [20]. While the reduction of engine emissions may also reduce tailpipe emissions, it is not an essential factor. Although reducing engine emissions can reduce tailpipe emissions as well, it is not the key factor. Rapid catalyst warm-up and sustainment are key factors in the reduction of tailpipe emissions, since the conversion efficiency of a cold catalyst is very low [16]. To develop a fuzzy controller to decrease fuel consumption and emissions of a parallel HEV concurrently, the authors in [8] designed a control scheme using multi-objective genetic algorithms. Sagha et al. [31] proposed a revised equivalent consumption minimization strategy (ECMS) to incorporate fuel consumption and also a NOx reduction control for a lightweight through-the-road architecture HEV. The control strategy is capable of decreasing NOx emissions near Euro 4 restrictions, while maintaining CO and HC emissions below Euro 4 and 5 restriction standards. Gao et al. [10] investigated the effect of an absorber that is capable of significantly reduce the hydrocarbon and nitrogen oxide emissions by provisionally storing them until the three-way catalyst is adequately warm to remove them from the exhaust. The absorber referenced can significantly influence the emission reduction for a PHEV in comparison to a HEV.

Smith et al. [34] experimentally validate a supervisory control system for a pre-transmission parallel PHEV powertrain architecture that reduces tailpipe emissions from a PHEV platform via the developing and refining vehicle supervisory control methods. Commencing with clean electric motor torque via various rule-based methods, the objective of the development was to substitute high engine torque demands. The approach was highly effective in the reduction of NOx emissions, but the model-based control schemes imply much better performance for various operating conditions. Through applying the dynamic programming method, the authors in [15] produced an adaptive supervisory powertrain controller (SPC) that optimally regulates the engine on/off, gear shift, and power-split strategies under several energy-to-distance ratios (EDRs) and catalyst temperature conditions and catalyst temperature conditions for a pre-transmission parallel plug-in hybrid electric compact SUV to attain near-optimal fuel economy and emission performance. The authors employed a simple simulation model of the powertrain to assess their proposed method. They also presumed that air-to-fuel ratio and spark ignition timing of the engine are controlled for ideal performance.

The control design procedure is similar to Sect. 4.3, but is has a different control-oriented model and the cost function is modified.

4.1.4.1 Control-Oriented Model Inside the High-Level Controller

In a power-split configuration, the engine, the electric motor, and the generator are joined by two planetary gear sets (see Fig. 3.1). By overlooking the loss of power

in planetary gear sets, the relation between torques will be [11]:, since the driveline dynamics is faster than other dynamic in the entire system.

$$\begin{aligned} zT_e &= (1+z)T_g \\ T_g &= z(T_f - T_r) \end{aligned} \tag{4.14}$$

where T_f is load as seen before the final drive and $T_r = \frac{r_2}{s_2} T_m$. Also $z = s_1/r_1$, where r_1 and s_1 are the number of ring and sun gear teeth of PSG1, respectively. A simple circuit model of the battery with a voltage source and an internal resistance is used instead of a high-fidelity chemistry-based battery model, i.e., a linearized form of Eq. (4.8). In brief, the equation of controller model is:

$$\begin{aligned} S\dot{O}C &= A\ SOC + B \begin{bmatrix} T_m \\ \omega_e \end{bmatrix} + \tilde{B} \begin{bmatrix} T_d \\ V \\ 1 \end{bmatrix} \\ y &= C\ SOC + D \begin{bmatrix} T_m \\ \omega_e \end{bmatrix} + \tilde{D} \begin{bmatrix} T_d \\ V \\ 1 \end{bmatrix} \end{aligned} \tag{4.15}$$

This model will be discretized before plugging into the controller equations. A, B, $\tilde{B}$, C, D, and $\tilde{D}$ can be found after linearization for each time step around the operating point. The control inputs of system are motor torque (T_m) and engine speed (ω_e). The only state is the battery SOC and disturbances to this system are driver demanded torque (according to gas pedal) (T_d) and vehicle velocity (V) which can be found according to demanded torque separately:

$$a_1 \frac{dV}{dt} + a_2 V^2 + a_3 = T_d \tag{4.16}$$

where $a_1 = mR_{tire}$, $a_2 = 0.5.\rho_{air}.A_d.c_d.R_{tire}$, and $a_3 = f_r.m.g.R_{tire}$

The parameters m, R_{tire}, ρ_{air}, A_d, c_d, and f_r are vehicle mass, tire radius, air density, vehicle frontal area, drag coefficient, and tire rolling resistance. Furthermore, a receding horizon control principle is utilized where the actual plant control input only takes the first sample of the control input signal, while disregarding the remainder of the trajectory.

4.1.4.2 NMPC Energy Management System

The modified cost function is:

$$\begin{aligned} J(k) = \sum_{i=1}^{N_p} ((w_1(SOC_{ref}(k+i) - SOC(k+i))^2 + w_2(\dot{m}_f(k+i))^2)) \\ + \sum_{j=1}^{N_c} (w_3(\Delta T_g(k+j))^2) + w_4(\Delta T_m(k+j))^2). \end{aligned} \tag{4.17}$$

The first term is related to keeping the state of charge around reference. The second term is for minimizing the fuel consumption. Two last terms are considered for making control inputs as smooth as possible, where ΔT_m and ΔT_g are the variation of motor and generator torque. w_1, w_2, w_3, and w_4 are weighting parameters chosen according to the predicted maximum value of the weighted variables. These weighting parameters have a significant effect on controller performance. Readers may refer to [39] for more details on determining weighting parameters. To solve this problem, we keep the constraints on (4.9) unchanged:

$$
\begin{aligned}
T_{min-e} < T_e < T_{max-e} \\
T_{min-m} < T_m < T_{max-m} \\
T_{min-g} < T_g < T_{max-g} \\
\omega_{min-e} < \omega_e < \omega_{max-e} \\
\omega_{min-r} < \omega_r < \omega_{max-r} \\
\omega_{min-g} < \omega_g < \omega_{max-g} \\
SOC_{min} < SOC < SOC_{max}
\end{aligned}
$$

Now, different possible control strategies can be investigated, based on trip information, by determining an appropriate reference SOC trajectory. Here, SOC_{ref} for CDCS and known traveling distance are considered to evaluate controls performance.

4.2 Low-Level Controls Design

Setpoints for the engine, the motor, and the generator torque were presented in the previous section. The low-level controls are required to be tuned in order for these sources to follow the setpoints. A standard PI control can be used for electric devices. However, Appendix B evaluates the design technique of a direct torque controller for high-fidelity simulation model of the electric drive and a different approach are required for the engine.

An appropriate method is required for robustness and to ensure stability, as well as to sustain optimal system performance under adverse operating conditions, since the engine torque setpoint originated from MPC is applied to a mean value gasoline engine model.

Sliding mode control (SMC) can be designed for optimal disturbance rejection and trajectory tracking, rapid dynamic response, and sound stability [36].

Because it is capable of handling the model ambiguities, SMC is a dependable model-based control method for engine torque management in practical cases.

Emission control is also examined here; while minimizing engine emissions, the engine low-level controller should be capable of tracking the designated engine torque. The throttle angle is the principle control input, while other inputs such as injected fuel rate and ignition timing, highly influence the transient behavior of the

engine. In order to obtain the best fuel economy and emission for the engine, the legislation does not permit changes to these two parameters away from their optimum values for an extended period of time [17]. As such, throttle angle is more reliable in general and in this case, dominates engine input to alter the steady-state response. However, the amount of injected fuel is required to be determined as well, in order to control air–fuel ratio, which contributes to engine emissions.

Engine torque management can be done in two ways: The first method uses the engine torque sensor to gage crankshaft torque for feedback control. The effects of engine combustion torque, friction torque, pumping torque, and accessory loads are studied by following this approach. As previously stated, spark timing and air–fuel ratio influences transient engine torque response; however, using torque control based on these two parameters as the major inputs keeps them away from the optimal ranges. As such, torque control in this manner cannot assure low emission, but does moderate ambiguities, particularly in engine aging.

The second method measures and control manifold pressure, as the engine torque is a function of cylinder air flow, which is subsequently a function of the manifold pressure. In fact, air–fuel ratio is an additional parameter that determines engine combustion torque. The control goal can be altered in order to make the manifold pressure follow the desired value, in the case of assuming constant air-to-fuel ratio and ignition timing. If the throttle is used to control manifold pressure, the ignition timing and air-to- fuel ratio on combustion torque do influence the torque control. As such, the drawback of this approach is a larger amount of calibration required to obtain an appropriate conversion from desired torque to desired manifold pressure for all engine operating conditions. Nonetheless, this strategy does not require a torque sensor [17].

The second approach is employed in this research, along with controlling the air to fuel ratio in order to attain a suitable emission performance [41].

4.2.1 Engine Control-Oriented Model

In order to reduce emissions, this research focuses on maximizing catalytic converter efficiency for various operating conditions of the engine. The conversion efficiencies are generally measured over a range of air-to-fuel ratios and catalytic converter body temperatures that require extensive data fitting and lookup tables. The conversion efficiency can be illustrated by the S-shaped Wiebe function, as proposed in [33]:

$$\eta_{conv} = (1 - exp\{-c_1(\frac{AFR - \lambda_0}{\Delta\lambda})^{m_1}\})(1 - exp\{c_2(\frac{T_{exh} - T_0}{\Delta T})^{m_2}\}) \qquad (4.18)$$

where c_1, c_2, m_1, m_2, λ_0, $\Delta\lambda$, T_0, and ΔT are constants that are determined by curve fitting to experimental data. AFR and T_{exh} stand for air-to- fuel ratio and exhaust gas temperature.

For the engine torque control, a simplified model of a mean value engine is required (see (3.1)–(3.5)). In these equations, P_m, $\dot{m}_{act,fuel}$, and T_{exh} are the state variables that define the manifold pressure, actual fuel rate, and exhaust temperature.

The amount of fuel going into the engine needs to be determined in order to estimate the combustion torque. In fact, due to vaporization, none of the injected fuel can enter the cylinders. In order to consider this, an alternate dynamic equation for wall-wetting effect is considered:

$$\ddot{m}_{act,fuel} = \frac{1}{\tau_f}(-\dot{m}_{act,fuel} + \dot{m}_{inj,fuel}) \tag{4.19}$$

where τ_f is a constant. The amount of injected fuel ($\dot{m}_{inj,fuel}$) is an addition control input in the problem. At this point, the engine generated torque can be projected via:

$$T_{ind} = \frac{\dot{m}_{act,fuel}.H_f.\eta_\Delta.\eta_{AFR}.\eta_i}{\omega_e} \tag{4.20}$$

where η_i, and H_f are engine thermal efficiency (approximately a function of engine speed and manifold pressure) and gasoline heat of combustion. Readers may refer to [30] for a table of numeric parameters used in this engine model. Figure 4.11 shows the air-to-fuel ratio and ignition timing efficiencies (η_{AFR} and η_Δ).

Air-to-fuel ratio can be described as:

$$AFR = \frac{\dot{m}_{air}}{\dot{m}_{act,fuel}} \tag{4.21}$$

According to [32], the third dynamic equation to estimate the exhaust gas temperature can be written as:

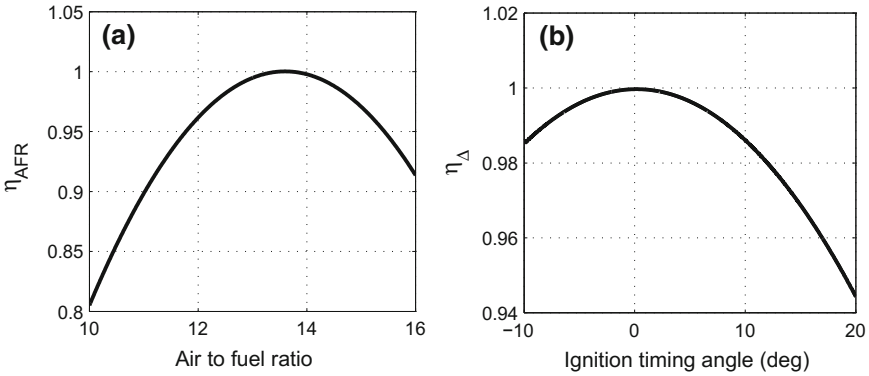


Fig. 4.11 Engine torque change with air-to-fuel ratio and ignition timing

$$\dot{T}_{exh} = \frac{\omega_e}{2\pi}[-T_{exh} + ST.AFI] \tag{4.22}$$

where $AFI = cos(0.13(AFR - 13.5))$, $ST = 7.5\Delta + 600$ and Δ is the angle of crankshaft in degree at which the ignition occurs. Here, it is assumed that the catalytic converter body temperature is proportional to exhaust temperature [44].

4.2.2 Engine Controls Design

To control the engine torque and catalytic converter efficiency using sliding mode control, three different sliding surfaces should be defined [29]: S_1, S_2, and S_3 for torque, air-to-fuel ratio, and exhaust temperature control.

Let $S_1 = T_{ind} - T_{des}$, where T_{des} is the reference engine torque dictated by the high-level controller. By taking the derivative of S_1, we have:

$$\dot{S}_1 = \dot{T}_{ind} - \dot{T}_{des} = \frac{\eta_v.N_{cyl}.V_D.H_f.\eta_\Delta.\eta_{AFR}.\eta_i}{60N_{eng}.R_{air}T_{man}}.\frac{P_m}{AFR} - \dot{T}_{des} \tag{4.23}$$

By using Eq. (4.20) and rearranging the terms of Eq. (4.23), we can find the control input for the first sliding surface:

$$A_{th} = \frac{V_m}{R_{air}.T_{man}.C_D.MA.PRI}[\frac{60N_{eng}.R_{air}T_{man}}{\eta_v.N_{cyl}.V_D.H_f.\eta_\Delta.\eta_{AFR}.\eta_i}(\dot{S}_1 + \dot{T}_{des}).AFR + (\frac{\eta_v N_{cyl} V_D \omega_e}{60N_{eng}V_m} + \frac{A\dot{F}R}{AFR}P_m)] \tag{4.24}$$

For the second sliding surface, we assume $S_2 = AFR - AFR_{des}$, where AFR_{des} is the desired air-to-fuel ratio value. By differentiating S_2 and taking time derivative of AFR, we can write:

$$\dot{S}_2 = A\dot{F}R - A\dot{F}R_{des} = \frac{\ddot{m}_{air} - AFR.\ddot{m}_{act,fuel}}{\dot{m}_{act,fuel}} \tag{4.25}$$

By using Eqs. (4.20) and (4.26), the injected fuel rate can be found as another manipulated input:

$$\dot{m}_{inj,fuel} = \frac{\tau_f}{AFR}\{\ddot{m}_{air} - \dot{m}_{act,fuel}(\dot{S}_2 + A\dot{F}R_{des})\} + \dot{m}_{act,fuel} \tag{4.26}$$

The desired air-to-fuel ratio is constant (stoichiometry) so $A\dot{F}R_{des} = 0$.

The last sliding surface belongs to exhaust temperature control. By taking time derivative of $S_3 = T_{exh} - T_{exh,des}$ and (4.22), we can find appropriate ignition timing as the third control input:

$$\Delta = \frac{1}{7.5}[\frac{T_{exh} + (2\pi/\omega_e)(\dot{S}_3 + \dot{T}_{exh,des})}{cos(0.13(AFR - 13.5))} - 600] \tag{4.27}$$

where $T_{exh,des}$ is the desired exhaust gas temperature. Indeed, the exhaust temperature should not be too high to prevent damaging the catalytic converter.

Now, $\dot{S}_1$, $\dot{S}_2$ and $\dot{S}_3$ can be designed to satisfy reachability condition ($\dot{S}S < 0$) and find an acceptable torque and emissions generation for the engine, accordingly. Actually, the two first sliding surfaces are coupled and make it more difficult to control the engine torque and AFR separately. Also, ignition timing and AFR changing can deteriorate engine torque performance (according to Eq. (4.20)). But, using appropriate functions and coefficients will make it possible to have a good performance, as the results show.

4.2.3 Results of Simulation

Figure 4.12 illustrates the simulation procedure with a driver who follows the predefined drive cycle with gas and brake pedals. Command from the gas pedal is calibrated to provide the required torque, which directly goes to the control system. In addition, the mechanical brake is a standard PI controller. The demanded torque and the vehicle velocity (found through Eq. 4.16), are fed to the control system, while the high-level controller calculates the control inputs every 2 s. The generator speed and torque engine torque, battery SOC, and its variation are additional information required for the high-level controller in order to project appropriate controls for drivability, fuel economy, and to sustain battery SOC near the predefined level. New setpoints can be calculated for engine, motor and generator torque, according to control inputs. Low-level controllers are responsible for tracking these setpoints as accurately as possible, while reducing emissions as elaborated below.

The high-fidelity MapleSim model is transformed into an optimized S-function in order to evaluate controls in the MATLAB environment.

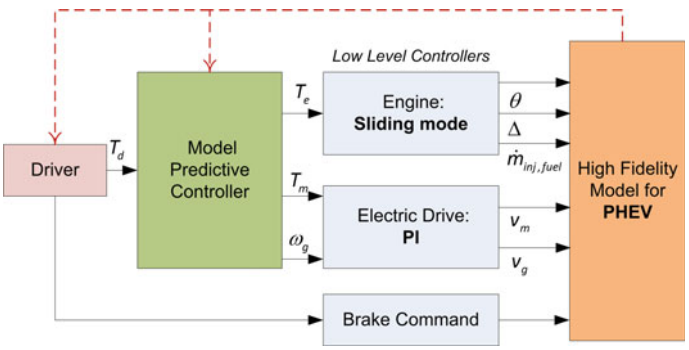


Fig. 4.12 Simulation procedure

4.2.3.1 Without Emissions Control

Figure 4.13 shows the vehicle follows two consecutive UDDS drive cycles in the blended mode strategy without controlling the emissions. As demonstrated, at the end of the trip the battery charge depletes to the lowest possible level. In fact, if the horizontal axis exhibits the traveled distance, the depletion trajectory will be a near linear profile. Therefore, the high-level controller has maintained drivability while keeping the final SOC above the minimum level. Figure 4.13b and c show that the low-level controllers have made the engine and MG2 torque follow the

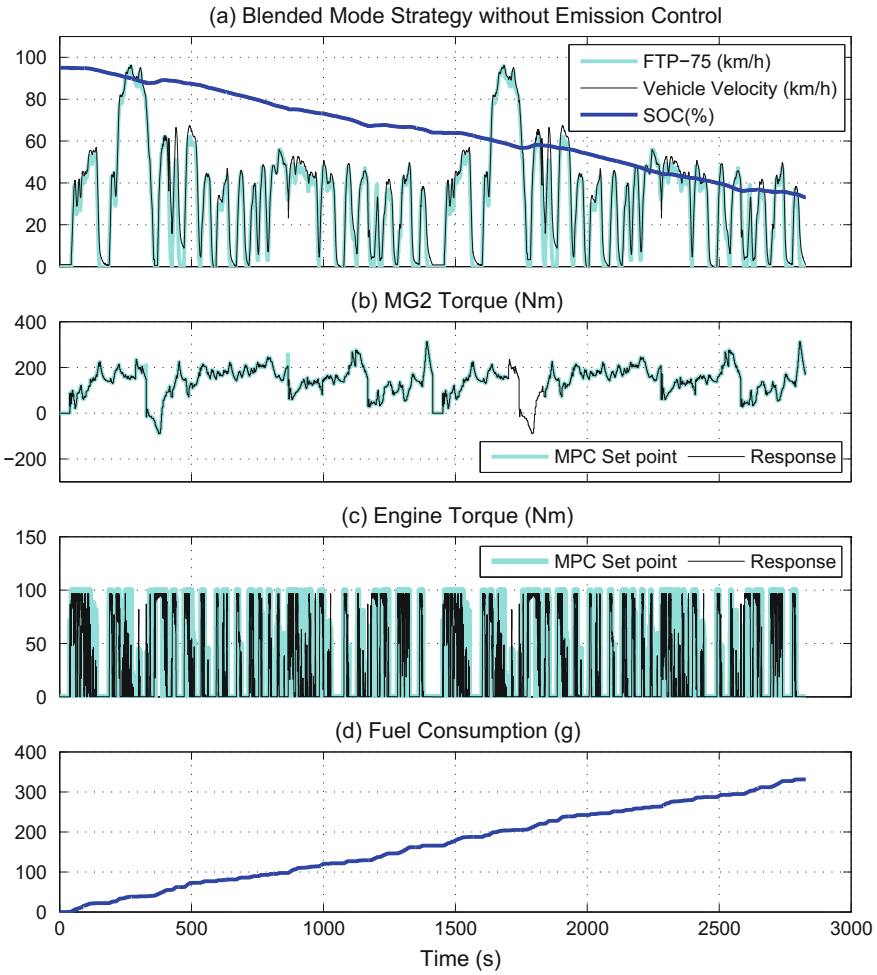


Fig. 4.13 Blended mode MPC strategy without emission control: **a** Velocity and Battery SOC **b** MG2 Torque **c** Engine Torque **d** Fuel Consumption

MPC-prescribed trajectory. According to Fig. 4.13d, the resultant fuel consumption is 1.66 l/100km (142 MPG).

Figure 4.14a demonstrates the drivability of vehicle in CDCS strategy, where the vehicle goes in pure electric mode at the beginning of its trip. When the battery state of charge drops to $SOC = 0.3$ at $t = 1714$ s, the engine kicks in and maintains the SOC around SOC_{ref}. Figure 4.14b shows that the vehicle is propelled only by MG2 up to 15.83 km. In this part of the trip, the engine is off (Fig. 4.14c). This strategy results in higher fuel consumption of 1.83 l/100km (128 MPG) in comparison to the blended mode strategy, according to Fig. 4.14d.

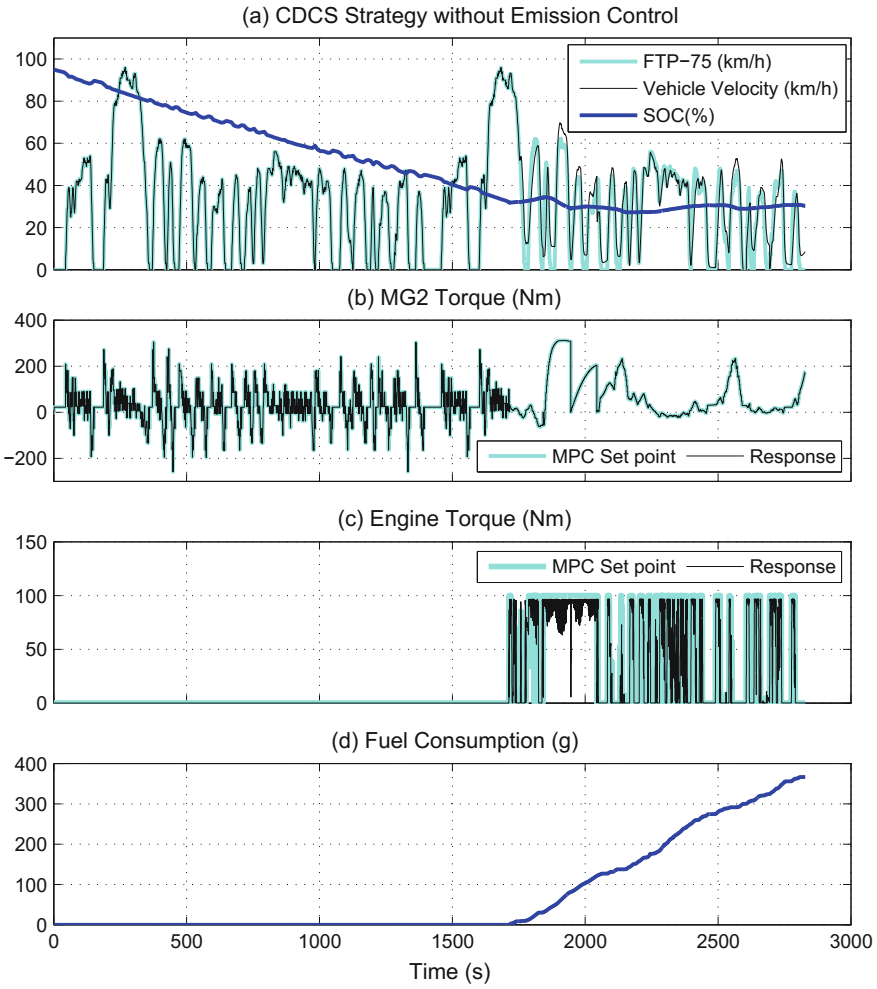


Fig. 4.14 CDCS MPC strategy without emission control: **a** Velocity and Battery SOC **b** MG-2 Torque **c** Engine Torque **d** Fuel Consumption

4.2.4 With Emissions Control

Figure 4.15a shows the drivability of the vehicle in blended mode while controlling the engine emissions. Figure 4.15b shows more frequent operation of MG2 in comparison to the case where the engine emission is not controlled (Fig. 4.13b). The reason is the failure of the engine to provide adequate torque to propel the vehicle for all time steps. This failure is due to the extra heat loss that occurs inside the engine. As previously stated, one approach to reduced emission is to maintain the catalyst temperature high. In order to reach this, the gas temperature inside the exhaust manifold must be increased. Altering the ignition timing allows for additional heat loss and

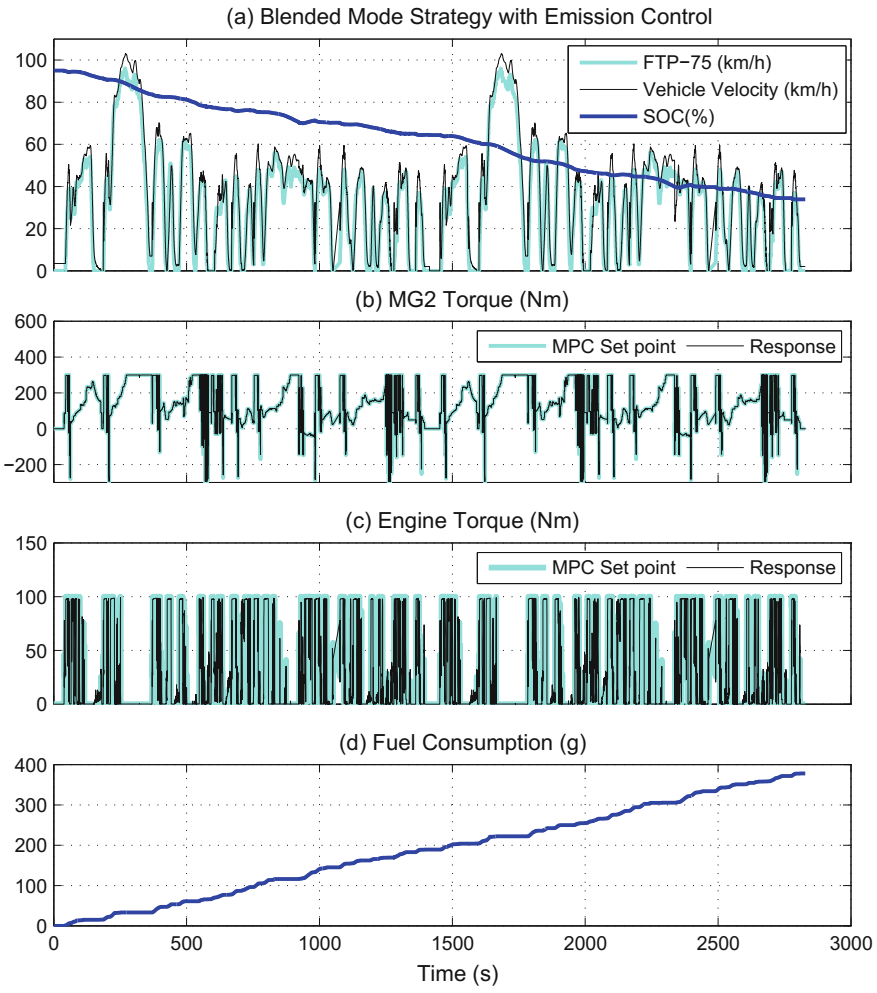


Fig. 4.15 Blended mode MPC strategy with emission control: **a** Velocity and battery SOC **b** MG-2 torque **c** Engine torque **d** Fuel consumption

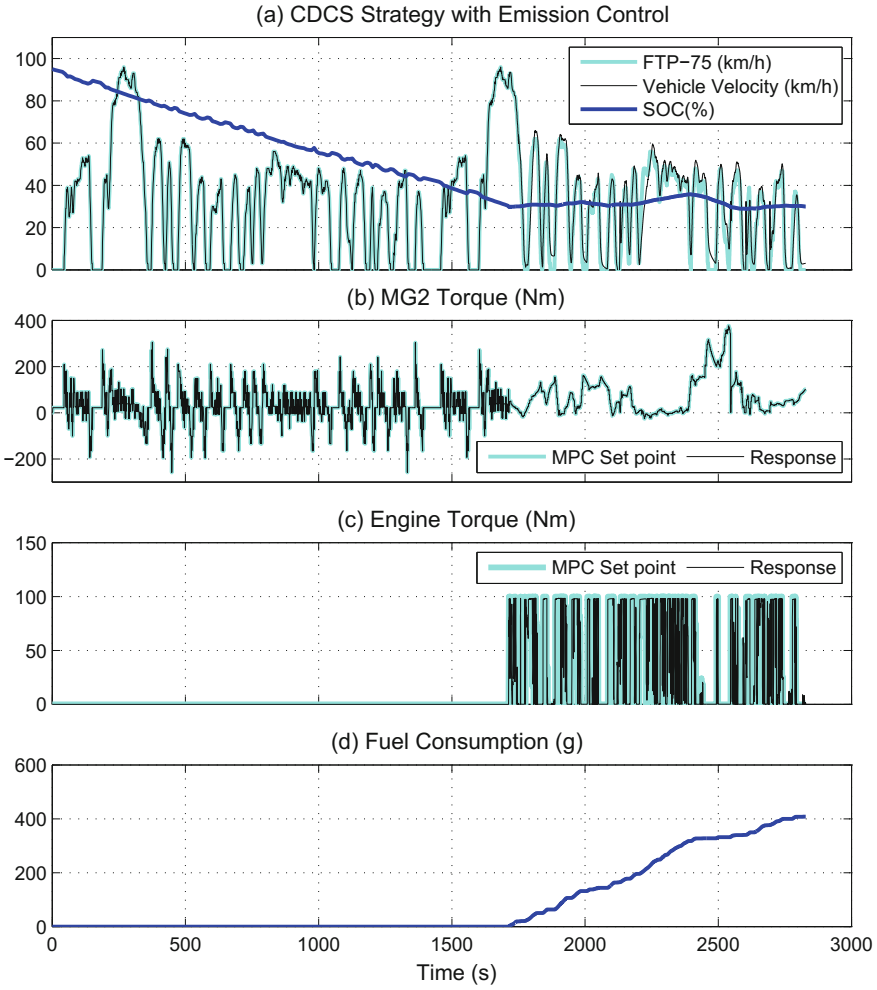


Fig. 4.16 CDCS MPC strategy with emission control: **a** Velocity and battery SOC **b** MG-2 torque **c** Engine torque **d** Fuel consumption

exhaust gas with higher temperature, to the detriment of higher fuel consumption. In short, a larger amount of the combustion energy must disperse as heat to warm-up the catalyst, instead of generating mechanical energy inside the engine. Figure 4.15b and c shows the SMC and PI controllers are successful in making the engine and MG2 follow the MPC-prescribed trajectories. Changing ignition timing and AFR result in a decrease of engine combustion torque which is shown by corresponding efficiencies in Eq. (4.20). Therefore, Fig. 4.15d indicates more fuel consumption of 1.89 l/100km (124 MPG).

Figure 4.16 shows the results for the CDCS strategy. Here, the fuel consumption has risen to 2.05 l/100km (115 MPG).

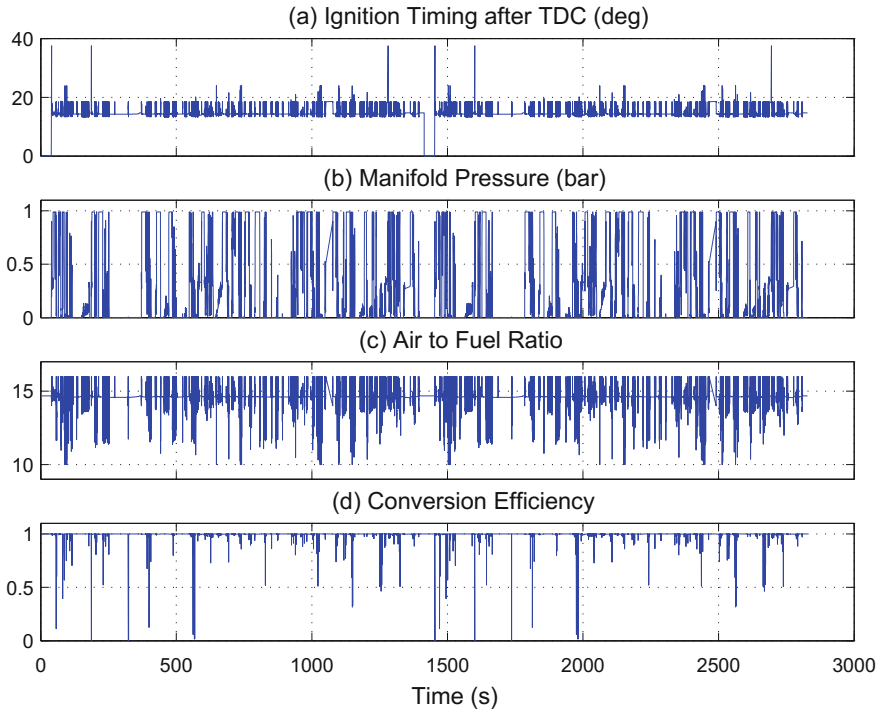


Fig. 4.17 Blended mode MPC strategy with emission control: **a** Ignition timing **b** Manifold pressure **c** AFR **d** Conversion efficiency

As previously stated, exhaust temperature is closely linked to the catalytic converter body temperature that contributes to conversion efficiency. Ignition timing significantly impacts the determination of exhaust temperature. Figure 4.17a shows ignition timing throughout the vehicle trip. For the sake of combustion stability, we confine the ignition timing within $\Delta \in [0, 20]$. As a result, exhaust temperature (T_{exh}) remains high enough for the whole trip.

Figure 4.17b shows the manifold pressure as compared with ignition timing and AFR versus throttle angle for different time steps. Figure 4.17c shows AFR which is determined by the second level of sliding mode control. As seen, AFR is alternating around the stoichiometry ratio and is confined within an acceptable range. Figure 4.17d shows the conversion efficiency that mostly remains at its maximum level.

Figure 4.18 demonstrates different results for engine emission control in the CDCS strategy. It is evident that the engine emission is zero for the full electric mode of driving.

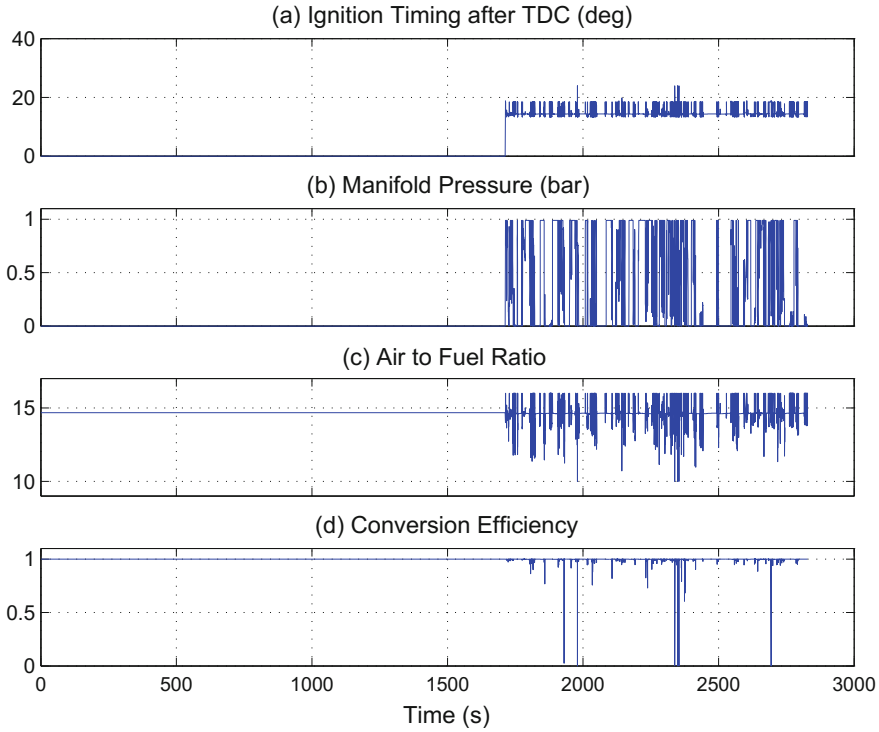


Fig. 4.18 CDCS MPC strategy with emission control: **a** Ignition timing **b** Manifold pressure **c** AFR **d** Conversion efficiency

4.3 Summary

In the first part of this chapter, an energy management strategy for a PHEV was designed using a discrete MPC concept for different levels of trip information. The system was chosen close to the specification of Toyota Prius plug-in hybrid. The model inside the controller was linearized and discretized, and the simulation time was comparable to the time needed for implementing the controller online in a practical situation. According to the simulation and in comparing to charge depletion/charge sustenance strategy, the fuel economy can be enhanced by up to 17.

The results are summarized in Table 4.2. Figure 4.19 depicts the dispersal of catalyst operating point for the two strategies described. Furthermore, performance assessment of the of the current high-level controls and the adaptive ECMS power management system are laid out in [41].

An energy management scheme was designed in the second part, including high- and low-level controllers, to reduce fuel consumption and engine emission of a power-split PHEV. The preceding model predictive control EMS was altered with a different control oriented model. The sliding mode control approach was used to design a low-

Table 4.2 Fuel consumption comparison

Strategy	Fuel consumption	(l/100km)	*Consumption increase by*
	w/o e-control	*w e-control*	*e-control (%)*
CDCS	1.83 (128 MPG)	2.05 (115 MPG)	12.02
Blended mode	1.66 (142 MPG)	1.89 (124 MPG)	13.86

e-control = emissions control, w = with, w/o = without

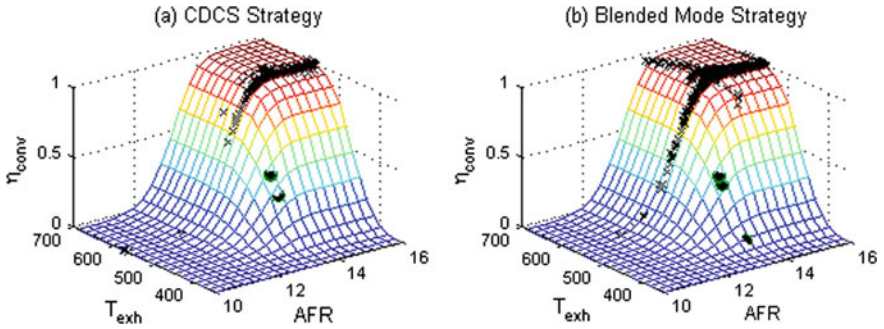


Fig. 4.19 Catalyst conversion efficiency for **a** CDCS **b** Blended mode strategies: without emission control (bullet marker)/with emission control (cross marker)

level controller for the gasoline engine in order for the engine to follow a desirable torque and emission performance. For more accurate results, the control scheme was applied to a high-fidelity simulation model of the vehicle, including a chemistry-based model of the lithium-ion battery developed in MapleSim 6.1. The simulation took place for both charge depletion/charge sustenance and blended mode strategies, with the results showing a promising fuel consumption of 1.89 l/100km (124 MPG) and 2.05 l/100km (115 MPG) for blended and CDCS strategies, respectively, while the engine emissions were controlled during the vehicle trip. It was demonstrated that engine emissions control can expand fuel consumption by 13.

References

1. Bitmead, R.R., Gevers, M., Wertz, V.: Adaptive Optimal Control: The Thinking Man's GPC. Prentice Hall, Upper Saddle River (1990)
2. Ceraolo, M., di Donato, A., Franceschi, G.: A general approach to energy optimization of hybrid electric vehicles. IEEE Trans. Veh. Technol. **57**, 1433–1441 (2008)
3. Chan, S.H., Hoang, D.L., Zhou, P.L.: Heat transfer and chemical kinetics in the exhaust system of a cold-start engine fitted with a three way catalytic converter. Proc. Inst. Mech. Eng. Part D J. Autom. Eng. **214**, 765–777 (2000)
4. Chen, H., Allgower, F.: A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. Automatica **14**, 1205–1217 (1998)

5. Clarke, D.W.: Advances in Model-Based Predictive Control. Oxford University Press, Oxford (1994)
6. de Oliveira Kothare, S.L., Morari, M.: Contractive model predictive control for constrained nonlinear systems. IEEE Trans. Autom. Control **45**, 1053–1071 (2000)
7. Del Re, L., Allgower, F., Glielmo, L., Guardiola, C., Kolmanovsky, I.: Automotive Model Predictive Control: Models Methods and Applications. Springer, Heidelberg (2010)
8. Dorri, M., Shamekhi, A.: Design of an optimal control strategy in a parallel hybrid vehicle in order to simultaneously reduce fuel consumption and emissions. SAE Technical Paper No. 2011-01-0894 (2011)
9. Ferris, C.M., Mangasarian, O.L., Wright, S.J. : Linear programming with MATLAB. In: The Society for Industrial and Applied Mathematics and the Mathematical Programming Society, Philadelphia (2007)
10. Gao, Z., Kim, M., Choi, J., Daw, C.S., Parks II J.E., Smith, D.E.: Cold-start emissions control in hybrid vehicles equipped with a passive adsorber for hydrocarbons and nitrogen oxides. Proc. Inst. Mech. Eng. Part D (J. Automob. Eng.). **226**, 1396–1407 (2012)
11. Guzzella, L., Sciarretta, A.: Vehicle Propulsion Systems: Introduction to Modeling and Optimization. Springer, Heidelberg (2007)
12. Keerthi, S.S., Gilbert, E.G.: Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: stability and moving-horizon approximations. J. Optim. Theory Appl. **57**, 265–293 (1988)
13. Kolmanovsky, I., Nieuwstadt, M., Sun, J.: Optimization of complex powertrain systems for fuel economy and emissions. IEEE Int. Conf. Control Appl. **32**, 833–839 (1999)
14. Kothare, M.V., Balakrishnan, V., Morari, M.: Robust constrained model predictive control using linear matrix inequalities. Automatica **32**, 1361–1379 (1996)
15. Kum, D., Peng, H., Bucknor, N.K.: Optimal catalyst temperature management of plug-in hybrid electric vehicles. In: American Control Conference (ACC), pp. 2732–2738 (2011)
16. Kum, D., Peng, H., Bucknor, N.K.: Supervisory control of parallel hybrid electric vehicles for fuel and emission reduction. J. Dyn. Syst. Meas. Control **133**, 833–839 (2011)
17. Lamberson, D.M.: Torque management of gasoline engines. M.Sc. Dissertation, University of California at Berkeley, Berkeley (CA) (2003)
18. Lee, J.H.: Model predictive control: review of the three decades of development. Int. J. Control Autom. Syst. **9**, 415–424 (2011)
19. Lee, J.H., Yu, Z.: Worst-case formulation of model predictive control for systems with bounded parameters. Automatica **33**, 415–424 (2011)
20. Lin, C.C., Peng, H., Grizzle, J.W., Kang, J.M.: Power management strategy for a parallel hybrid electric truck. IEEE Trans. Control Syst. Technol. **11**, 839–849 (2003)
21. Marano, V., Tulpule P., Stockar, S., Onori, S., Rizzoni, G.: Comparative study of different control strategies for plug-in hybrid electric vehicles. SAE World Congress and Exhibition Detroit MI USA (2004)
22. Mayne, D., Michalska, H.: Receding horizon control of nonlinear systems. IEEE Trans. Autom. Control **35**, 763–781 (1997)
23. Mosca, E.: Optimal, Predictive, and Adaptive Control. Prentice Hall, Upper Saddle River (1995)
24. Moura, S.J., Stein, J.L., Fathy, H.K.: Battery health-consious power management for plug-in hybrid vehicles via stochastic control. In: ASME Dynamic Systems and Control Conference (2010)
25. Muta K., Yamazaki, M., Tokieda, J.: Development of new-generation hybrid system THS IIDrastic improvement of power performance and fuel economy. SAE Paper No. 2004-01-0064 (2004)
26. Nicholas, K., Hedrick, K., Borrelli, F.: Integrating traffic data and model predictive control to improve fuel economy. In: IFAC Symposium on Control in Transportation Systems (2009)
27. Polak, E., Yang, T.H.: Moving horizon control of linear systems with input saturation and plant uncertainty, parts 1 and 2. Int. J. Control **58**, 613–663 (1993)
28. Primbs, J.A., Netistic, V.: A framework for robustness analysis of constrained finite receding horizon control. In: American Control Conference (ACC), pp. 2718–2722 (1998)

29. Razavian, R.S., Taghavipour, A., Azad, N.L., McPhee, J.: Design and evaluation of a real-time optimal control system for series hybrid electric vehicles. Int. J. Electr. Hybrid Veh. **4**, 260–288 (2012)
30. Saeedi, M.: A mean value internal combustion engine model in MapleSim. M.Sc. Dissertation, Waterloo (ON) (2010)
31. Sagha, H., Farhangi, S., Asaei, B.: Modeling and design of a nox emission reduction strategy for lightweight hybrid electric vehicles. In: 35th Annual Conference of IEEE Industrial Electronics (IECON '09), pp. 334–339 (2009)
32. Sanketi, P.R., Zavala, J.C., Hedrick, J.K.: Dynamic surface control of engine exhaust hydrocarbons and catalyst temperature for reduced cold start emissions. In: Proceedings of the International Federation of Automatic Control (IFAC) Conference (2005)
33. Shaw, B., Fischer, G.D., Hedrick, J.K.: A simplifed coldstart catalyst thermal model to reduce hydrocarbon emissions. In: Proceedings of 15th Triennial World Congress of the International Federation of Automatic Control, pp. 1915–1921 (2002)
34. Smith, D., Lohse-Busch, H., Irick, D.: A preliminary investigation into the mitigation of plug-in hybrid electric vehicle tailpipe emissions through supervisory control methods. SAE Technical Paper No. 2010-01-1266 (2010)
35. Souder, J.S., Hedrick, J.K.: Adaptive sliding mode control of airfuel ratio in internal combustion engines. Int. J. Robust Nonlinear Control **14**, 525–541 (2011)
36. Iordanou, H.N., Surgenor, B.: Experimental evaluation of the robustness of discrete sliding mode control versus linear quadratic control. IEEE Trans. Control Syst. Technol. **5**, 254–260 (1997)
37. Taghavipour, A., Azad, N.L., McPhee, J.: An optimal power management strategy for power split plug-in hybrid electric vehicles. Int. J. Veh. Des. **60**, 286–304 (2012)
38. Taghavipour, A., Azad, N.L., McPhee, J.: Design and evaluation of a predictive powertrain control system for a plug-in hybrid electric vehicle to improve fuel economy and emissions. IMechE Part D J. Autom. Eng. **229**, 624–640 (2015)
39. Taghavipour, A., Vajedi, M., Azad, N.L., McPhee, J.: Predictive power management strategy for a phev based on different levels of trip information. In: IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling (E-COSM'12), vol. 60, pp. 326–333 (2012)
40. Tian, Y., Sun, W., Qu, D., Wang, L.: Study of a exhaust after-treatment system applied to hybrid vehicle. In: Asia-Pacific Power and Energy Engineering Conference (APPEEC), pp. 1–4 (2010)
41. Vajedi, M., Taghavipour, A., Azad, N.L., McPhee J.: A comparative analysis of route-based power management strategies for real-time application in plug-in hybrid electric vehicles. In: American Control Conference (ACC), pp. 2612–2617 (2014)
42. Wang, L.: Model Predictive Control System Design and Implementation Using MATLAB. Springer, Heidelberg (2009)
43. Wurzenberger, J.C., Auzinger, G., Heinzle, R., Wanker, R.: 1D modeling of reactive fluid dynamics, cold start behavior of exhaust systems. SAE Technical Paper No. 2006-01-1544 (2006)
44. Zavala, J.C. : Engine modeling and control for minimization of hydrocarbon coldstart emissions in SI engines. Ph.D. Dissertation, University of California, Berkeley, Berkeley (CA) (2007)
45. Zheng, A., Morari, M.: Stability of model predictive control with soft constraints. IEEE Trans. Autom. Control **40**, 1818–1823 (1995)
46. Zheng, C., Vahidi A., Li, X., Essenmacher, D.: Role of trip information preview in fuel economy of plug-in hybrid vehicles. In: ASME Dynamic Systems and Control Conference (2009)

Chapter 5
Multi-parametric Predictive Control

This chapter discusses the design procedure of EMS using the explicit model predictive control (eMPC) approach in order to reduce optimal control computational time. Also, a control-oriented parameter estimation method is introduced to improve the control-oriented model accuracy which leads to better performance of the eMPC controller.

Despite the demonstrated benefits of MPC, its capabilities are limited due to the computational effort required for solving the on-line optimization problem. This MPC shortcoming can be overcome by using the so-called explicit/multi-parametric MPC (eMPC or mp-MPC) methods. In eMPC, the on-line optimization problem involved in the MPC is solved off-line using multi-parametric programming approaches and the control variables and the value function of the optimization problem are derived as explicit functions of the system state variables, as well as the critical regions of the state-space where these functions are valid. This significantly reduces the computational effort required for the MPC implementation [1].

In this chapter, a near-optimal, real-time implementable solution for a PHEV EMS is proposed, using explicit model predictive control. Di Cairano et al. [2] utilized eMPC solution for a series HEV; however, this research is considered the first endeavor at designing an explicit model predictive controller and implementing it for the Toyota Prius plug-in hybrid power-split architecture. There are some challenges associated with finding a suitable control-oriented model. Due to the exponential increase of the size of the controls database by the number of state variables, utilizing eMPC is only useful for relatively small problems. In order to capture the complex dynamics of a power-split PHEV powertrain, the control-oriented model must be very simple, yet precise. In addition, the control-oriented model and the optimization cost function should be selected in a manner that assures a feasible solution aside from optimality, stability and desirable performance for the controller. A switched discrete-time control system is proposed in this section and a stability analysis is required to ensure that the control system maintains its performance for

A. Taghavipour et al., *Intelligent Control of Connected Plug-in Hybrid Electric Vehicles*,
Advances in Industrial Control, https://doi.org/10.1007/978-3-030-00314-2_5

all possible PHEV operating points. As such, an innovative control-oriented model is presented for simplicity and to tackle the above-mentioned issues.

The eMPC controller performance is analogous to the accuracy of the control-oriented model. To attain an enhanced control oriented model that is both simple and captures more dynamic relations of the powertrain, a control-relevant parameter estimation method is proposed in this chapter. The EMS is reformatted to establish the efficiency of the control-oriented model improvement, based on the new model.

By developing a suitable control-oriented model, the energy management strategy and application is examined first. The subsequent prototypes from solving eMPC are presented in Sect. 5.2, in addition to the physical interpretation of the distinctive regions.

The stability of the closed-loop system is discussed in Sect. 5.3, and the designed controller is applied to the simulation model, with the result compared to the MPC approach results in Sect. 5.4.

5.1 eMPC Energy Management Strategy Design

An energy management scheme for the Toyota Prius plug-in hybrid is developed using the eMPC approach in this section. Bemporad et al. [3] presented a technique to determine the linear quadratic regulator for constrained systems through off-line multi-parametric linear programming (mp-LP) and multi-parametric quadratic programming (mp-QP). Piecewise, the control law appeared as linear and continuous and its implementation as a look-up table was achievable, i.e.: different linear state feedback laws were applied to different polyhedral regions. As such, the online control computation is condensed to the determination of the region connected with the current state, followed by the application of the stored control law associated with that region.

The controller design process can be divided into two distinct stages: offline and online procedures. The purpose of the offline procedure is to populate various look-up tables through a control-oriented model encompassing the appropriate control actions for varying system operating points. A multi-parametric programming issue is resolved with a preliminary condition in this method, resulting in a polytope with a specific control action. This is followed by the investigation of the entire state space to locate the other polytopes and control actions. Solving the optimization issue may result in large quantities of look-up tables. In this case, to downsize the look-up tables and enhance controller speed in the execution stage, region reduction methods can be utilize to eliminate some redundant constrictions in the optimization problem. The online procedure occurs during controls implementation and requires a fast and efficient search algorithm to look up the tables and determine which of the polytopes includes the preliminary state variable. The algorithm is referred to as the point location and the eMPC law can be acquired once the corresponding polytope is found [4]. These steps are illustrated in Fig. 5.1.

Each step is detailed separately in the proceeding subsections.

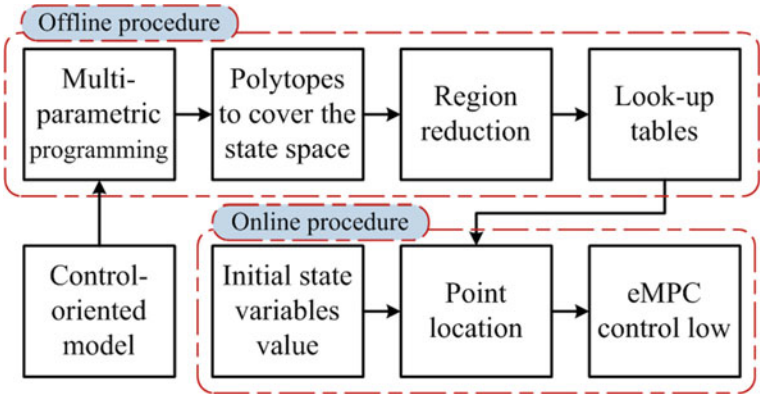


Fig. 5.1 eMPC design procedure

5.1.1 Control-Oriented Model

A comparatively plain model is necessary in order to take advantage of the explicit model predictive control approach. The below model inside the controller is utilized to attain smaller look-up tables and enable the application of the controller to a commercial control hardware with limited memory and computational power:

$$Z(k+1) = \mathbf{A}Z(k) + \mathbf{B}U(k) \tag{5.1}$$

where $Z = \left[\, SOE,\ E \,\right]^T$ and $U = \left[\, P_{BAT},\ P_{ENG},\ P_{BRK} \,\right]^T$.

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} a_1 & 0 & a_2 \\ a_3 & a_4 & a_5 \end{bmatrix}$$

There are two state variables in this model: battery state of energy (SOE) and tractive energy E. Battery SOE is defined as the ratio of battery stored/released energy to the battery total usable energy. E is defined as the tractive energy that is required to propel the vehicle, i.e. the power required in a time step. Two sources of energy are available in the powertrain: the battery and the fuel. Two control inputs are considered to address the mentioned sources as P_{BAT} and P_{ENG} which are the battery and the engine power, respectively. In fact, the battery power is the power summation of two electrical motors power on board. Braking power is added to the array of control actions to stop the vehicle. The coefficients are a function of efficiencies (electrical and mechanical) as well as the control period that is considered in the design procedure.

5.1.2 Optimization Problem Formulation

To hybridize a vehicle, it is imperative to enhance the fuel economy and emissions performance, while preserving the vehicle drivability, in contrast to the baseline vehicle.

Fuel economy is closely correlated to the battery depletion trajectory in a PHEV. As such, a tracking term inside the cost function is considered that SOE should follow at each time step (SOE_{ref}). To address drivability, another term is added inside the cost function to ensure that the hybrid powertrain provides adequate propulsion power to the driver's request (E_{ref}). Equally importantly, the fuel consumption must be decreased; the engine fuel consumption is presumed to be proportional to the power generated by the engine, and therefore, as one of the presumed control actions, the engine power can be curtailed. The cost function and constraints along the prediction horizon can be written as follows:

$$\begin{gathered}\min_U \sum_{j=1}^{N_p} \left\{ Y^T(j)\mathbf{Q}Y(j) + U^T(j)\mathbf{R}U(j) \right\} \\ s.t. \\ \begin{bmatrix} SOE_{min} \\ E_{min} \end{bmatrix} \leq \begin{bmatrix} SOE \\ E \end{bmatrix} \leq \begin{bmatrix} SOE_{max} \\ E_{max} \end{bmatrix} \\ U_{min} \leq U \leq U_{max} \end{gathered} \tag{5.2}$$

where $Y = \left[SOE - SOE_{ref},\ E - E_{ref} \right]^T$, and N_p is the prediction horizon length.

$$\mathbf{Q} = \begin{bmatrix} \omega_1 & 0 \\ 0 & \omega_2 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} \omega_3 & 0 & 0 \\ 0 & \omega_4 & 0 \\ 0 & 0 & \omega_5 \end{bmatrix}$$

The above cost optimization is subjected to the constraints on state variables and control actions. $\omega_i's$ are the weighting parameters that should be tuned for the best performance.

We have 4 unknowns in the cost function: 2 state variables and 2 setpoints. In order to separate the unknowns from each other, Eq. (5.2) should be rewritten in the following way:

$$\begin{gathered}\min_U \sum_{j=1}^{N_p} \left\{ X^T(j)\mathbf{H}X(j) + U^T(j)\mathbf{R}U(j) \right\} \\ s.t. \\ \mathbf{G}U \leq \mathbf{W} + \mathbf{S}X \end{gathered} \tag{5.3}$$

where $X = \left[SOE,\ E,\ SOE_{ref},\ E_{ref} \right]^T$, $X_{min} = \left[SOE_{min},\ E_{min} \right]^T$, and $X_{max} = \left[SOE_{max},\ E_{max} \right]^T$,

$$\mathbf{H} = \begin{bmatrix} \omega_1 & -\omega_1 & 0 & 0 \\ 0 & 0 & \omega_2 & -\omega_2 \\ -\omega_1 & \omega_1 & 0 & 0 \\ 0 & 0 & -\omega_2 & \omega_2 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} -\mathbf{I}_{3\times 3} \\ \mathbf{I}_{3\times 3} \\ \mathbf{O}_{4\times 4} \end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix} -U_{min} \\ U_{max} \\ -X_{min} \\ X_{max} \end{bmatrix}$$

$$\mathbf{S} = \begin{bmatrix} \mathbf{O}_{6\times 4} \\ \begin{matrix} 1 & 0 & 0\,0 \\ 0 & 1 & 0\,0 \\ -1 & 0 & 0\,0 \\ 0 & -1 & 0\,0 \end{matrix} \end{bmatrix}$$

In multi-parametric programming, the objective is to find the optimizer U^* for a whole range of parameters X, i.e. $U^*(X)$ as an explicit function of the parameter X. The cost function is quadratic, so a multi-parametric quadratic programming (mp-QP) problem is solved. As shown in [3], we wish to solve problem (5.3) for all X within the polyhedral set of feasible values X_N. According to [5], if the multi-parametric quadratic program (5.3) is considered, then the set of feasible parameters X_N is convex, the optimizer U^* is continuous and piecewise affine (PWA), and the optimal value function J^* is continuous, convex and piecewise quadratic.

$$U^*(X) = f_i X + g_i, X \in T_i = \{X \mid h_i X \leq k_i\}; i = 1, \ldots, N \tag{5.4}$$

Each $\{T_i\}_{i=1}^N$ defines a polytope which will be referred to as a region. Note that the evaluation of the PWA solution (5.4) of the mp-QP provides the same result as solving the quadratic program, i.e. for any given parameter X, the optimizer $U^*(X)$ is identical to the optimizer obtained by solving the quadratic program (5.3) for X.

To solve the mp-QP problem, we need to solve the active constraint identification problem. A feasible parameter $\hat{X}$ is determined and the associated QP (5.3) is solved. This will yield the optimiser U^* and active constraints defined as inequalities that are active at solution. The rows indexed by the active constraints are extracted from the constraint matrices $\mathbf{G}$, $\mathbf{W}$ and $\mathbf{S}$ to form the matrices $\mathbf{G}_A$, $\mathbf{W}_A$ and $\mathbf{S}_A$.

It is possible to use the Karush-Kuhn-Tucker (KKT) conditions to obtain an explicit representation of the optimiser $U_N(x)$ which is valid in some neighborhood of $\hat{X}$:

$$\begin{gathered} \mathbf{H}U + \mathbf{G}^T\lambda = 0 \\ \lambda^T\left(\mathbf{G}U - \mathbf{W} - \mathbf{S}\hat{X}\right) = 0 \\ \lambda \geq 0 \\ \mathbf{G}U \leq \mathbf{W} + \mathbf{S}\hat{X} \end{gathered} \tag{5.5}$$

We can find the optimized variable $U = -\mathbf{H}^{-1}\mathbf{G}^T\lambda$. For inactive constraints, it holds that $\lambda = 0$. For active constraints with the corresponding Lagrange multipliers λ_A, inequality constraints are changed to equalities. Substituting for U from (5.4) into equality constraints gives:

$$\begin{gathered} -\mathbf{G}_A\mathbf{H}^{-1}\mathbf{G}_A^T\lambda_A + \mathbf{W}_A + \mathbf{S}_A\hat{X} = 0 \\ \Longrightarrow \lambda_A = -(\mathbf{G}_A\mathbf{H}^{-1}\mathbf{G}_A^T)^{-1}(\mathbf{S}_A\hat{X} + \mathbf{W}_A) \end{gathered} \tag{5.6}$$

The optimal control trajectory U are given as affine functions of $\hat{X}$

$$U^*(\hat{X}) = \mathbf{H}^{-1}\mathbf{G}_A^T(\mathbf{G}_A H^{-1}\mathbf{G}_A^T)^{-1}(\mathbf{S}_A\hat{X} + \mathbf{W}_A) = f_i\hat{X} + g_i \tag{5.7}$$

In the next step, the set of states is determined where the optimizer U^* satisfies the same active constraints and is optimal. Such a region is characterized by two inequalities written compactly as $h_i X \leq k_i$ where

$$\begin{aligned} h_i &= \begin{bmatrix} \mathbf{G}f_i - \mathbf{S} \\ (\mathbf{G}_A\mathbf{H}^{-1}\mathbf{G}_A^T)^{-1}\mathbf{S}_A \end{bmatrix} \\ k_i &= \begin{bmatrix} \mathbf{W} - \mathbf{G}g_i \\ -(\mathbf{G}_A\mathbf{H}^{-1}\mathbf{G}_A^T)^{-1}\mathbf{W}_A \end{bmatrix} \end{aligned}$$

Once the controller region is computed, the algorithm proceeds iteratively until the entire feasible state space X_N is covered with controller regions T_i, i.e. $X_N = \cup_{i=1,\ldots,N} T_i$ in order to explore the whole state space.

5.1.3 Region Reduction

A small amount of constraints demarcating a region is preferred at the implementation stage, since the controller rapidly inspects the constraints to attain the proper control action. Consequently, computation of the minimal representations of the controller regions T_i where h_i and k_i are given according to (5.8) can significantly reduce the computational load in most multi-parametric programming solvers [6].

Commencing with a prearranged piecewise affine solution, in [7] the authors present an method to relegate the number of partitions by optimally merging regions with the same affine gain to preserve the original solution, but equivalently expressed with a minimal number of partitions. Nevertheless, techniques for attaining a more

drastic decline of complexity necessitate altering the solution by accepting a certain level of sub-optimality with respect to the original problem formulation [8].

In [9, 10] the authors propose recursive rectangular partitions of the parameter space to determine a sub-optimal solution to general classes of multi-parametric programming problems. Based on a dynamic programming formulation of the finite-horizon optimal control problem, in [11], the authors propose an approach to relax optimality within a prescribed bound in favor of the reduced complexity of the solution, which in the case of linear systems and piecewise affine convex costs leads to another form of computing approximate mp-LP (multi-parametric linear programming) solutions.

In this way, there are a couple of approaches to identify redundant constraints and remove them in order to reduce the number of regions. An ordinary way to address this problem is to solve n LPs (in the worst-case of $n - 1$ constraints) for each region to detect and remove all redundant constraints according to [12]. Another approach is called ray shooting [13], which is suitable for the cases where the fraction of redundant constraints is low. On the other hand, the bounding box approach is most useful for polytopes with many easily detected redundant constraints. The region reduction that is used here is a combination of ray shooting and bounding box in order to find the redundant constraints even faster [14].

5.1.4 Point Location Problem

In this part, the point-location or set membership problem is addressed for the class of discrete-time control problems with linear state and input constraints for which an explicit time-invariant piecewise state feedback control law over a set of overlapping polyhedral regions is given. The point-location problem comes into play on-line when evaluating the control law. One must identify the state space region in which the measured state lies at the current sampling instance. As the number of defining regions grows, a purely sequential search through the regions is too lengthy to achieve high sampling rates. Hence, it is important to find an efficient on-line search strategy to evaluate the control action in time.

Baotic et al. [15] offers two novel algorithms to evade storing the redundant polyhedral regions by utilizing the properties of multi-parametric linear and quadratic solutions, subsequently lowering the online storage demands and computational complexity of the evaluation control. In [16] the authors propose to classify the hyperplanes defining the regions on a binary search tree (potentially subdividing some of the regions further), so that the time to locate the state vector on-line within the partition becomes logarithmic in the number of stored cells, and memory space is saved.

In this case the popular concept of interval trees [17] is applied to obtain a list of candidates that are potential solutions for the point-location problem. Standard interval trees are efficiently ordered binary search trees used to resolve a set of potentially overlapping one-dimensional line segments that comprise a particular

point or line segment. These line segments can be located through the bounding box method. A local search on the list of candidates must follow in order to establish the polytope that contains the current state variable [9].

The optimization problems have been solved in multi-parametric toolbox [18]. Proceeding the resolution of the mp-QP problem, various look-up table and control actions have been produced. The next section focuses on the details of these lookup tables.

5.2 Energy Management Polytopes

Based on the drive cycle maximum demanded power and also the battery state of charge we can discretize the SOE_{ref} and E_{ref} range. Here, we define 9 levels for each of them. By solving the mp-QP problem, we end up with 81 different sets of polytopes; each contains a definite control action. The total number of the mentioned polytopes is 3153. Figure 5.2 shows how the polytopes are distributed for different SOE_{ref} and E_{ref} levels. By doing region reduction, one can reduce the total number of polytopes to 3123.

To get more insight to the problem, we can consider the set which belongs to $E_{ref} = 0$ and $SOE_{ref} = 60$. It consists of 33 polytopes. As shown in Fig. 5.3, the number of polytopes around the reference setpoints is higher. The reason is that the eMPC controller is supposed to track a predefined level of SOE and E.

Figure 5.4 shows the control action versus different measured values of SOE and E (initial conditions) at current sampling instance. It contains 33 polytopes. We can analyze Fig. 5.3 in 4 different regions:

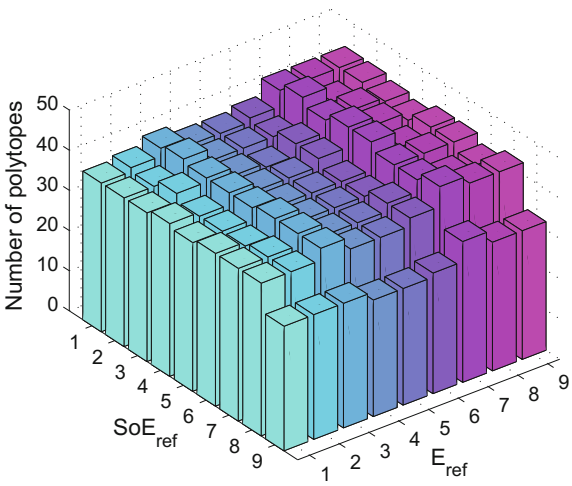


Fig. 5.2 Number of polytopes for different levels of E_{ref} and SOE_{ref}

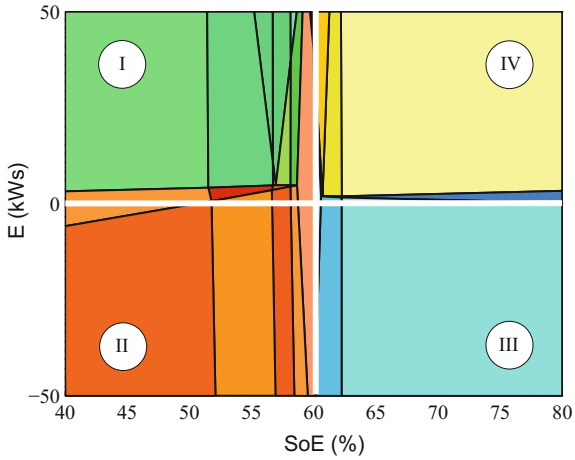


Fig. 5.3 Polytope set for $E_{ref} = 0$ and $SOE_{ref} = 60\%$

Region(I) $E > E_{ref}$ and $SOE < SOE_{ref}$: In this part, the controller has to increase the battery state of charge and slow down the vehicle. There are two ways to do that. One is to increase the engine power to charge the battery. The other one is to increase the braking power to get use of regenerative braking. Figure 5.4 shows that the controller uses both ways to get to the objective in region (I). The battery power should be negative indicating that it is being charged (Fig. 5.4a). Moreover, if $\delta E = |E - E_{ref}|$ is high, the braking power will be more (as shown in Fig. 5.4c)

Region(II) $E < E_{ref}$ and $SOE < SOE_{ref}$: Since $E < E_{ref}$, the powertrain is required to provide propulsion power from the engine and/or the electric drive. But, in this case, the battery state of energy is less than the reference value ($SOE < SOE_{ref}$), so we cannot use electric drive to assist the engine to propel the vehicle by depleting the battery further. On the other hand, we cannot use the regenerative braking for charging the battery (zero in Fig. 5.4c), since we cannot stop the vehicle. As a result, the engine plays a key role in this case. As shown in Fig. 5.4a, $P_{BAT} < 0$, because SOE is less than the reference value. Region (II) is the worst case for fuel consumption among all other propulsion scenarios.

Region(III) $E < E_{ref}$ and $SOE > SOE_{ref}$: In this region, we should accelerate the vehicle ($P_{BRK} = 0$). The electric drive can assist the engine since we have enough charge in the battery. Another objective of the controller is to minimize the engine power (in order to reduce fuel consumption). In region (III), there is no need to increase SOE so the electric drive can take care of propelling the vehicle. Moreover, the engine power is changing based on the magnitude of δE (as shown in Fig. 5.4b)

Region(IV) $E > E_{ref}$ and $SOE > SOE_{ref}$: In this case, we neither need propulsion power nor battery charging. So, there is no need to run the engine (for the sake of fuel consumption); as a result the engine power is zero throughout this region. But, we need to stop the vehicle so $P_{BRK} \neq 0$. On the other hand $P_{BAT} > 0$ to deplete the battery to return SOE closer to SOE_{ref}.

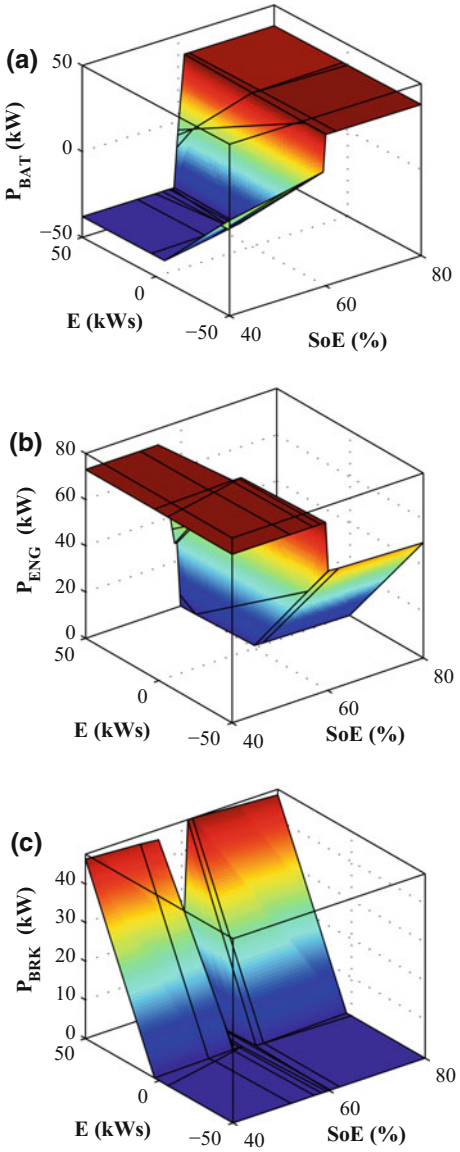


Fig. 5.4 Control actions for $E_{ref} = 0$ and $SOE_{ref} = 60\%$ based on different initial conditions (**a**) battery power (**b**) engine power (**c**) braking power

Figure 5.5 shows the continuous cost function over the mentioned set of polytopes which is piece-wise quadratic. In the next section, this function is used for stability analysis.

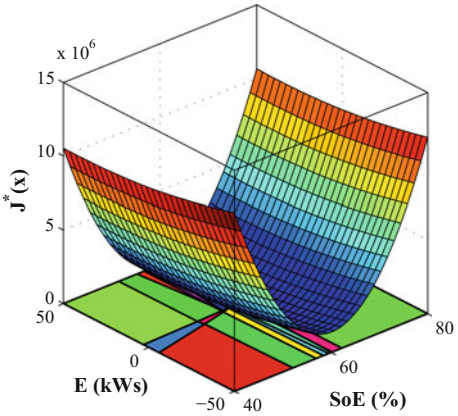


Fig. 5.5 Cost function for $E_{ref} = 0$ and $SOE_{ref} = 60\%$ based on different initial conditions

5.3 Stability Notes

As stated previously, the closed-loop system with MPC controller is globally asymptotically stable, exclusively when the optimization problem is feasible.

For the eMPC problem, feasibility of the solution is not adequate for proving stability. Since we have a switched discrete-time system, the stability of the closed-loop system should be investigated in 3 levels. Firstly, the local stability of the closed-loop system around the equilibrium point in each of 81 sets of polytopes should be proven. Secondly, the global stability of the mentioned controller throughout that specific set of polytopes is proven. Finally, the stability of the closed-loop system must be investigated, while the controller switches between different sets of polytopes based on reference SOE and E.

In each set of polytopes which belongs to a definite SOE_{ref}, E_{ref}, the controller drives the state variables to the mentioned reference values in finite time steps and $Z_0 = [SOE_{ref}, E_{ref}]^T$ is the equilibrium point in each set. To prove the local stability of the closed-loop system, we pick the polytope which contains Z_0. The control corresponding to that polytope is:

$$\hat{U} = f_0 \hat{Z} + g_0 \tag{5.8}$$

By applying the above control to the control-oriented model we can find the closed-loop system equation as:

$$Z(k+1) = (\mathbf{A} + \mathbf{B} f_0) Z(k) + \mathbf{B} g_0 \tag{5.9}$$

By defining $\tilde{Z} = Z - Z_0$, we transfer the state variables to the equilibrium point. As a result we have:

$$\tilde{Z}(k+1) = (\mathbf{A} + \mathbf{B} f_0)\tilde{Z}(k) + \mathbf{B} g_0 + (\mathbf{A} + \mathbf{B} f_0 - I_{2\times 2}) Z_0 = \tilde{\mathbf{A}}\tilde{Z}(k) + \tilde{\mathbf{B}} \tag{5.10}$$

Now, we can investigate the stability of (5.10) around $\tilde{Z} = 0$. First, we show that $\tilde{\mathbf{A}}$ is locally and asymptotically stable for all 81 sets of polytopes. We have a discrete switching system and need to make sure that the spectral radius of $\tilde{\mathbf{A}}$ is less than unity. Figure 5.6 shows that the spectral radius of $\tilde{\mathbf{A}}$ is less than unity.

We show that if $\tilde{\mathbf{A}}$ is stable and $\tilde{\mathbf{B}}$ is bounded then the closed-loop system (5.10) is stable. For a discrete system, if $V_1(\tilde{Z}_k) > 0$ exists and $\Delta V_1(\tilde{Z}_{k+1}, \tilde{Z}_k) = V_1(\tilde{Z}_{k+1}) - V_1(\tilde{Z}_k) < 0$ then the system is exponentially stable in the sense of Lyapunov [19]. Since $\tilde{\mathbf{A}}$ is stable, we can find $P_1 > 0$ and $Q > 0$ such that:

$$\tilde{\mathbf{A}}^T P_1 \tilde{\mathbf{A}} - P_1 + Q = 0 \tag{5.11}$$

We assume that $V_1(\tilde{Z}_k) = \tilde{Z}_k^T P_1 \tilde{Z}_k$

$$\Delta V_1(\tilde{Z}_{k+1}, \tilde{Z}_k) = \tilde{Z}_{k+1}^T P_1 \tilde{Z}_{k+1} - \tilde{Z}_k^T P_1 \tilde{Z}_k$$
$$= \tilde{Z}_k^T (\tilde{\mathbf{A}}^T P_1 \tilde{\mathbf{A}} - P_1) \tilde{Z}_k + \tilde{\mathbf{B}}^T P_1 \tilde{Z}_{k+1} + \tilde{\mathbf{B}}^T P_1 \tilde{Z}_k$$

If $Q = \mathbf{I}_{2\times 2}$ in (5.11) we can write:

$$\Delta V_1(\tilde{Z}_{k+1}, \tilde{Z}_k) = -\tilde{Z}_k^T \tilde{Z}_k + \tilde{\mathbf{B}}^T P_1 \tilde{Z}_{k+1} + \tilde{\mathbf{B}}^T P_1 \tilde{Z}_k \tag{5.12}$$

Suppose that in (5.11) we take $P_1 = \mathbf{I}_{2\times 2}$ and $Q > 0$, then we can say $\tilde{\mathbf{A}}^T \tilde{\mathbf{A}} \leq I_{2\times 2}$ and $\|\tilde{Z}_k\|$ is monotonically convergent:

$$\|\tilde{Z}_{k+1}\| \leq \|\tilde{\mathbf{A}}\| \|\tilde{Z}_k\| \leq \|\tilde{Z}_k\| \tag{5.13}$$

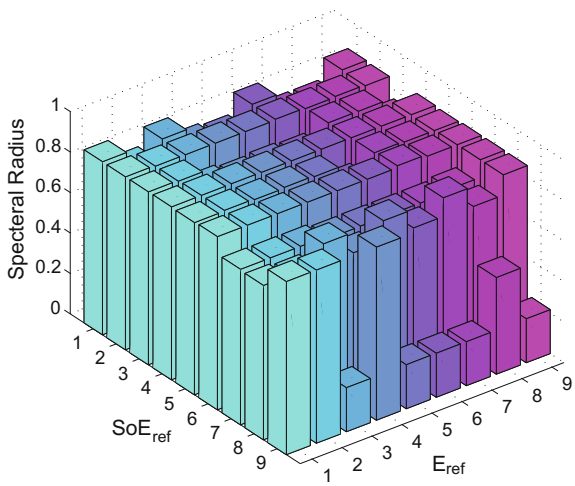


Fig. 5.6 Spectral radius of $\tilde{A}$ for different levels of E_{ref} and SOE_{ref}

Therefore:

$$\Delta V_1(\tilde{Z}_{k+1}, \tilde{Z}_k) \leq -\|\tilde{Z}_k\|^2 + 2\|\tilde{\mathbf{B}}\|\|P_1\|\|\tilde{Z}_k\| \tag{5.14}$$

If $\tilde{\mathbf{B}}$ is bounded, there is a $\beta > 0$ such that $\|\tilde{\mathbf{B}}\| < \beta\|\tilde{Z}_k\|^2$. As a result:

$$\Delta V(\tilde{Z}_{k+1}, \tilde{Z}_k) \leq \|\tilde{Z}_k\|^2(1 - 2\beta\|P_1\|\|\tilde{Z}_k\|) \tag{5.15}$$

For $\|\tilde{Z}_k\| < (1/2\beta\|P_1\|)$, $\Delta V_1(\tilde{Z}_{k+1}, \tilde{Z}_k) < 0$ and (5.10) would be stable. In this problem, $\beta = 10^{-8}$.

Now, we have to investigate the global stability of the closed-loop system for each set of polytopes.

Theorem. The equilibrium $x = 0$ is exponentially stable on sets of polytopes if there exist a function $\bar{V}(x)$ where ($\alpha, \varpi > 0$):

$$\alpha\|x\|^2 < \bar{V}(x) < \varpi\|x\|^2 \tag{5.16}$$

with a negative forward difference $\Delta\bar{V}(x_{k+1}, x_k) = \bar{V}(x_{k+1}) - \bar{V}(x_k) < 0$ when $x_k \in T_j \backslash 0$ and $x_{k+1} \in T_i$ [20].

We introduce the following function as a positive definite candidate (since $\mathbf{Q} > 0$) for V:

$$\bar{V}(\tilde{Z}) = \sum_{j=1}^{N_p} \{\tilde{Z}^T(j)\mathbf{Q}\tilde{Z}(j)\} \tag{5.17}$$

which is a part of the cost function. We previously proved that $\|\tilde{Z}_{k+1}\| \leq \|\tilde{Z}_k\|$ so we can easily get $\Delta\bar{V}(\tilde{Z}_{k+1}, \tilde{Z}_k) < 0$. As a result, the closed-loop system is globally and exponentially stable.

Up to now, the stability of the closed-loop system of each set of polytopes is investigated, whereas the controller switches between different sets of polytopes to cover all operating points. A switched system is stable if all individual subsystems are stable and the switching is sufficiently slow, so as to allow the transient effects to dissipate after each switch. In [21], this property is formulated and justified using multiple Lyapunov techniques. In this work, the switching frequency depends on the dynamics of (SOE_{ref}, E_{ref}). As mentioned before, (SOE_{ref}, E_{ref}) are bounded values. As a result, we assume the following equations govern the dynamics of those reference values.

$$\begin{bmatrix} SOE_{ref}(k+1) \\ E_{ref}(k+1) \end{bmatrix} = \begin{bmatrix} 1/\zeta & 0 \\ 0 & 1/\zeta \end{bmatrix} \begin{bmatrix} SOE_{ref}(k) \\ E_{ref}(k) \end{bmatrix} + \Gamma \tag{5.18}$$

where ζ should be chosen in such a way that guarantees (5.19) stability and also make the switching system slower than the control-oriented model. For stability, ζ should be greater than unity, so that the poles of (5.18) are located inside the unity circle in z-plane. On the other hand, these poles should be far enough from the center of unity circle to slow down the system (5.18) response. We assume that ρ is the largest

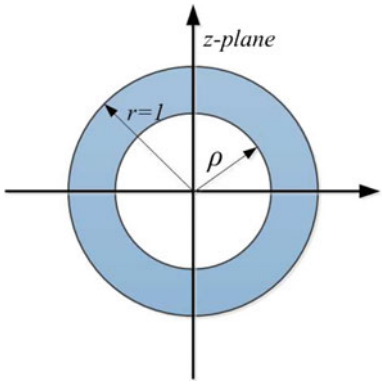


Fig. 5.7 The locus of switching system poles in z-plane

spectral radius of $\tilde{A}$ for all 81 sets of polytopes. The driving behavior determines ζ. If we choose $1/\zeta > \rho$, the switching system will be slower than the control-oriented model. As a result, if ζ is determined in such a way that the poles of the switching system are located inside the dark ring of Fig. 5.7, the switched system will be stable.

We append the control-oriented model to the switching system:

$$X(k+1) = \begin{bmatrix} \mathbf{I}_{2\times 2} & \mathbf{O}_{2\times 2} \\ \mathbf{O}_{2\times 2} & (1/\zeta)\mathbf{I}_{2\times 2} \end{bmatrix} X(k) + \begin{bmatrix} \mathbf{B} \\ \mathbf{O}_{2\times 3} \end{bmatrix} U(k) + \begin{bmatrix} \mathbf{O}_{2\times 1} \\ \Gamma \end{bmatrix} \tag{5.19}$$

where $1 < \zeta < 1/\rho$. For the closed-loop system, (5.20) can be transformed to:

$$\begin{bmatrix} \tilde{Z}_{k+1} \\ \tilde{Z}_{0,k+1} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{A}} & \mathbf{O}_{2\times 2} \\ \mathbf{O}_{2\times 2} & (1/\zeta)\mathbf{I}_{2\times 2} \end{bmatrix} \begin{bmatrix} \tilde{Z}_k \\ \tilde{Z}_{0,k} \end{bmatrix} + \begin{bmatrix} \tilde{\mathbf{B}} \\ \Gamma \end{bmatrix} \tag{5.20}$$

Since the spectral radius of $(1/\zeta)\mathbf{I}_{2\times 2}$ is less than unity and Γ is bounded ($\|\Gamma\| < 115$), according to the above discussion there is a $V_2(\tilde{Z}_{0,k}) = \tilde{Z}_{0,k}^T P_2 \tilde{Z}_{0,k} > 0$ such that $\Delta V_2(\tilde{Z}_{0,k+1}, \tilde{Z}_{0,k}) \leq 0$ where $P_2 > 0$.

For the whole system, we introduce a positive definite V ($P_1, P_2 > 0$) such that:

$$\begin{aligned} V\left(\begin{bmatrix} \tilde{Z}_k \\ \tilde{Z}_{0,k} \end{bmatrix}\right) &= \begin{bmatrix} \tilde{Z}_k & \tilde{Z}_{0,k} \end{bmatrix} \begin{bmatrix} P_1 & \mathbf{O}_{2\times 2} \\ \mathbf{O}_{2\times 2} & P_2 \end{bmatrix} \begin{bmatrix} \tilde{Z}_k \\ \tilde{Z}_{0,k} \end{bmatrix} \\ &= \tilde{Z}_k^T P_1 \tilde{Z}_k + \tilde{Z}_{0,k}^T P_2 \tilde{Z}_{0,k} = V_1 + V_2 \end{aligned} \tag{5.21}$$

We proved that $\Delta V_1 < 0$ and $\Delta V_2 < 0$, so that $\Delta V = \Delta V_1 + \Delta V_2 < 0$. Now, we can say that the closed-loop system (5.19) is stable.

5.4 eMPC Performance Simulation

After finding the polytopes and the corresponding control actions, we need to implement the controller to the simulation model by using low-level controls. Basically, we have to change the provided power to torque and speed for different components. Figure 5.8 shows the procedure that is done at each control time step. At the beginning, we have E_{ref} and SOE_{ref} as well as initial SOE and demanded energy that are given to the eMPC controllers. By using the mentioned point location algorithm, the appropriate controls among the polytopes can be found. We are looking for T_e, T_m, and ω_g. By having T_e we can control the engine throttle to the desired engine torque. On the propulsion side, we have P_{BAT} and P_{ENG}. Once we got P_{ENG}, we can use the optimal operating line of the engine, which gives us the most efficient operating point for the given P_{ENG}. Now, we have the engine speed and torque for the optimum operating point. The engine torque setpoint can directly be given to the engine low-level controller. If we measure the vehicle velocity, we will be able to get the MG1 speed setpoint by using the speed constraint relation on the first planetary gear set ($z = \frac{s_1}{r_1}$). Meanwhile, if we use static torque relation on the planetary gear set, we can find the MG1 torque based on the engine torque. Now MG1 power is calculated and we can find the MG2 power, since we have got the P_{BAT} from eMPC controller. By measuring the MG2 speed at the current time step, we are able to find the last setpoint value which is MG2 torque. Now, we can implement the controller to the simulation model.

In this section, the eMPC EMS is applied to the high-fidelity simulation model including the low-level controls. The results of MIL simulation are reviewed in Figs. 5.9, 5.10, 5.11, 5.12, 5.13 and 5.14.

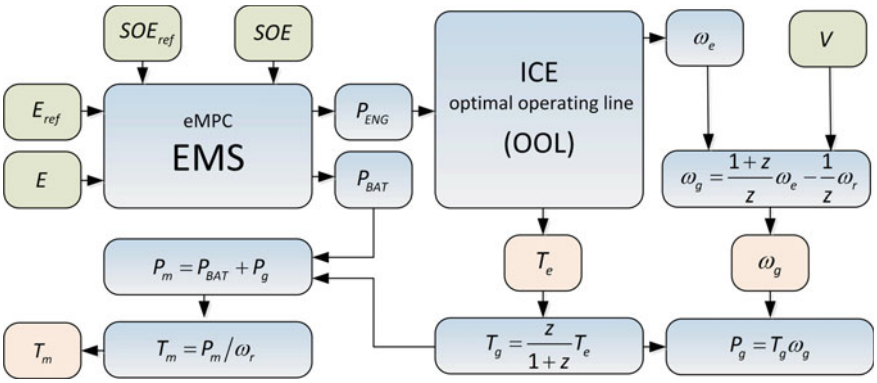


Fig. 5.8 Low-level controls implementation

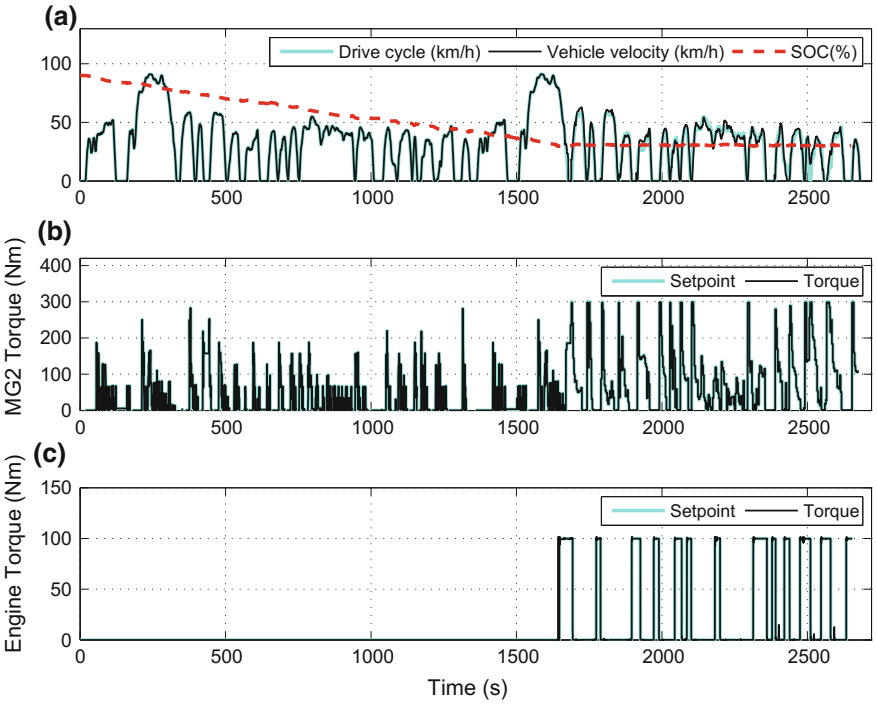


Fig. 5.9 CDCS eMPC strategy with emission control: **a** Velocity and battery SOC **b** MG2 torque **c** Engine torque

5.4.1 No Knowledge of Trip Information

Figure 5.9a shows that the vehicle follows 2 UDDS drive cycles for CDCS strategy. Figure 5.9b, c show the performance of low-level controls in tracking the setpoints determined by the eMPC EMS. Figure 5.10 demonstrates the emissions control performance in terms of maximizing HC conversion of catalytic converter along with engine transients. The sliding mode controller keeps the HC conversion efficiency around unity by controlling the air-fuel ratio and ignition timing as shown in Fig. 5.11. PHEV fuel consumption will be reduced to $2.09\,\mathrm{L}/100\,\mathrm{km}(113MPG)$ by using eMPC EMS.

5.4.2 Known Travelling Distance

Figures 5.12, 5.13 and 5.14 show similar results for the blended mode strategy. Fuel consumption for this strategy is $1.81\,\mathrm{L}/100\,\mathrm{km}(130MPG)$ by considering engine emissions control.

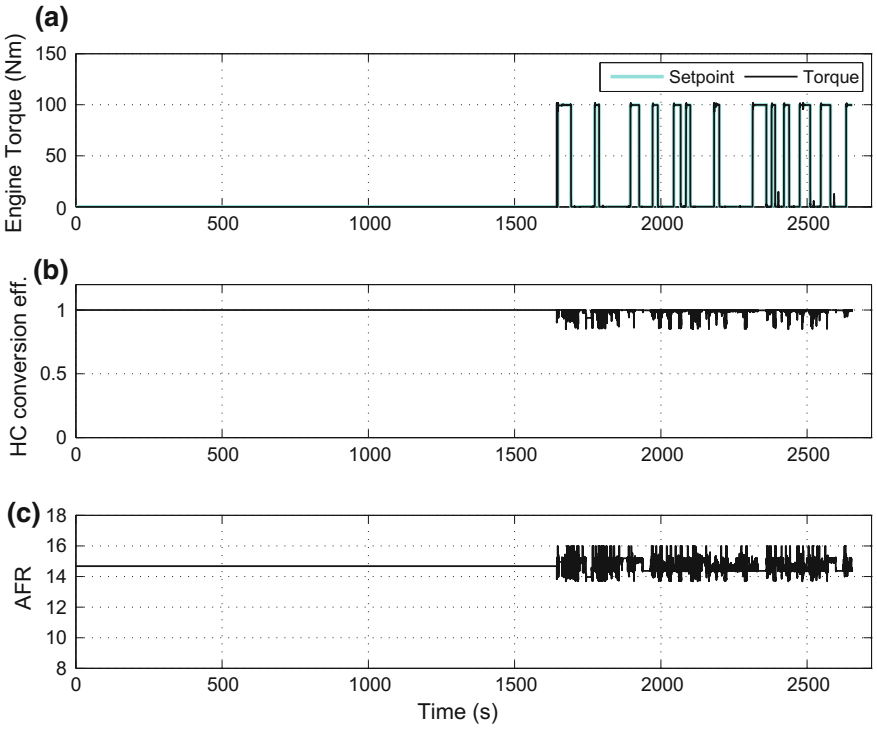


Fig. 5.10 CDCS eMPC strategy with emission control: **a** Engine torque **b** HC conversion efficiency **c** Air-to-fuel ratio (AFR)

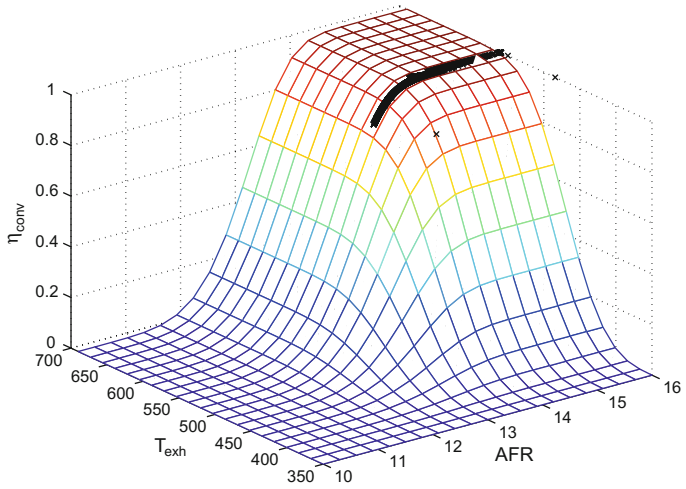


Fig. 5.11 Catalyst conversion efficiency for CDCS eMPC strategy

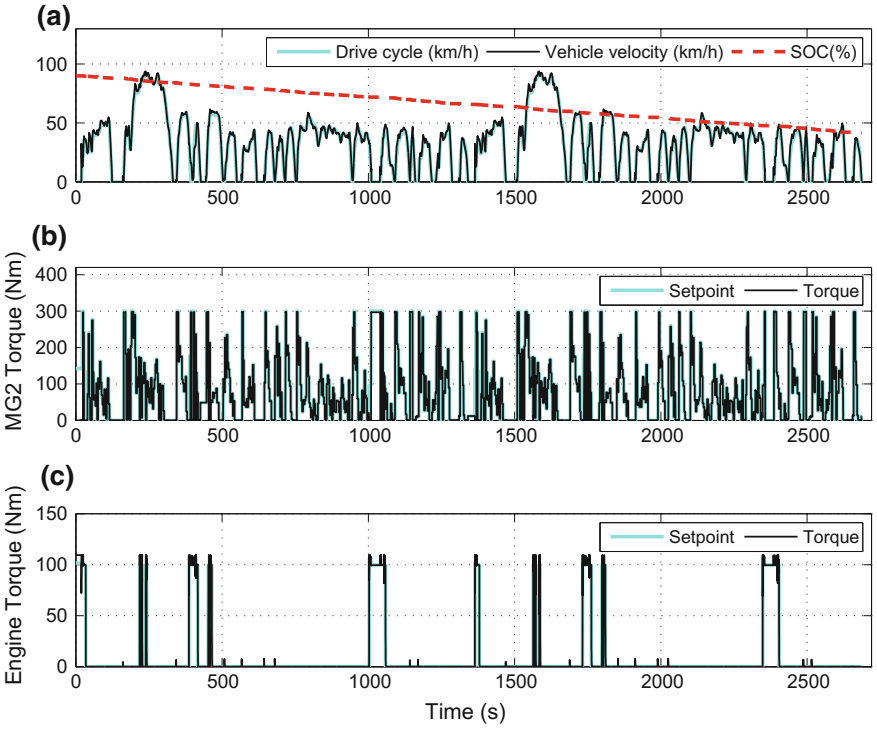


Fig. 5.12 Blended eMPC strategy with emission control: **a** Velocity and battery SOC **b** MG2 torque **c** Engine torque

5.4.3 Discussions

Fuel economy for MIL testing using the high-fidelity simulation model is compared in Table 5.1. Table 5.1 shows that explicit model predictive control reduces fuel economy by 1.74% and improves it by 4.84% for CDCS and blended mode strategies, respectively when compared to the MPC approach. But note that the primary purpose of considering explicit model predictive control was to maintain the performance of MPC while make it faster during implementation stage. The MIL simulation takes 1885s for simulating 2828s, which is three times faster than MIL using the MPC EMS on the average. This shows that eMPC has a superior potential to be implemented to a commercial control hardware with limited computational power. As a measure of drivability performance, eMPC EMS can follow the designated drive cycle with the root mean square error of 0.87 km/h and 0.89 km/h for CDCS and Blended mode strategies, respectively.

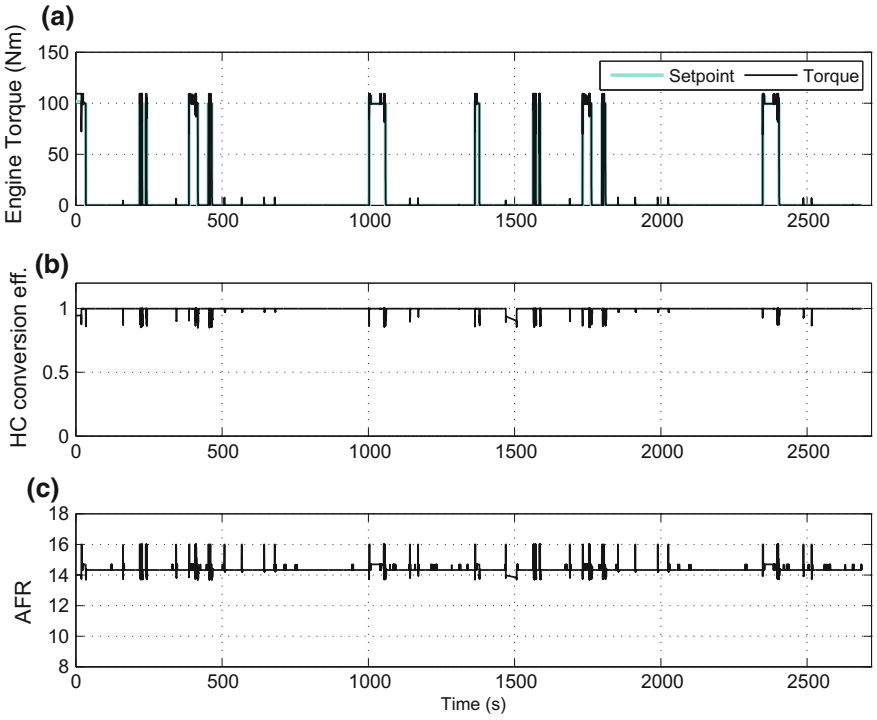


Fig. 5.13 Blended eMPC strategy with emission control: **a** Engine torque **b** HC conversion efficiency **c** Air-to-fuel ratio (AFR)

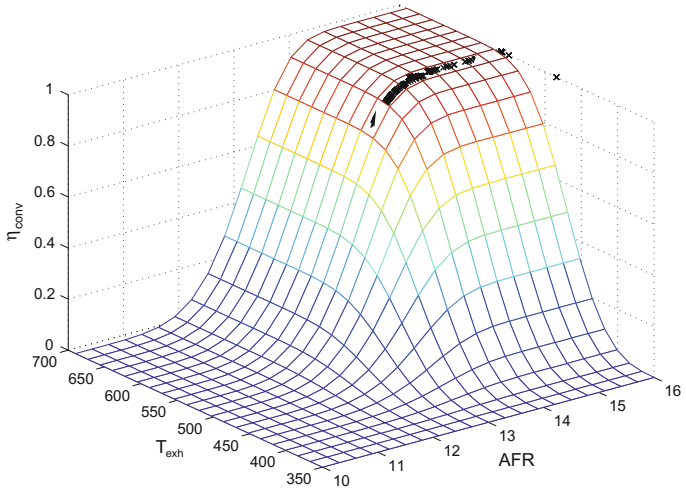


Fig. 5.14 Catalyst conversion efficiency for blended eMPC strategy

Table 5.1 MIL with the high-fidelity powertrain model: fuel economy for different control strategies

Control strategy	MPC (MPG)	eMPC (MPG)	Improvement (%)
Charge depletion/charge sustenance	115	113	−1.74
Linear blended mode	124	130	4.84

5.5 eMPC Performance Benchmarking via HIL

For implementing the eMPC EMS onto the ECU, a database with the size of 1.5 MB plus the eMPC search algorithm should be stored in the hardware memory. The search algorithm code is not in-lined, and cannot be compiled to the MotoTron ECU. Unfortunately, by in-lining the algorithm code, the size of code plus eMPC database exceeds 2MB flash memory size of the ECU.

To solve this problem, the eMPC energy management was modified. The control action surfaces versus state variables were approximated with some new look-up tables. Using this technique, we reduced the size of the controller from 2 MB to 143 kB for the larger CRPE-eMPC EMS.

Figure 5.15 shows different parts of the high-fidelity simulation model inside the real-time computer.

The ECU passes 3 control actions: P_{BAT}, P_{ENG}, and P_{BRK} to the real-time target via CAN bus at every 5 milliseconds. zThe real-time target runs the higher-fidelity model and corresponding low-level controls at every 1 ms. As shown in Fig. 6.3, P_{BAT} and P_{ENG} are fed into the setpoint configurer where the setpoints for the low level controllers are determined. Then, T_m, ω_g, and T_e are transferred to the electric

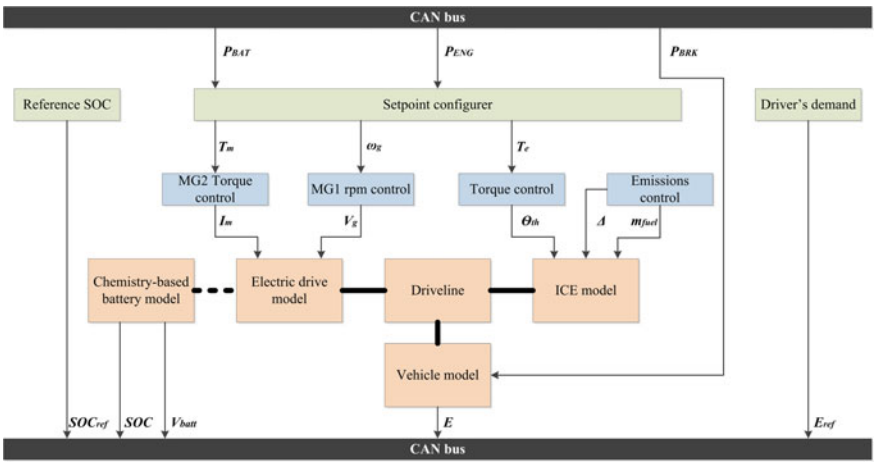


Fig. 5.15 High-fidelity model inside the real-time computer

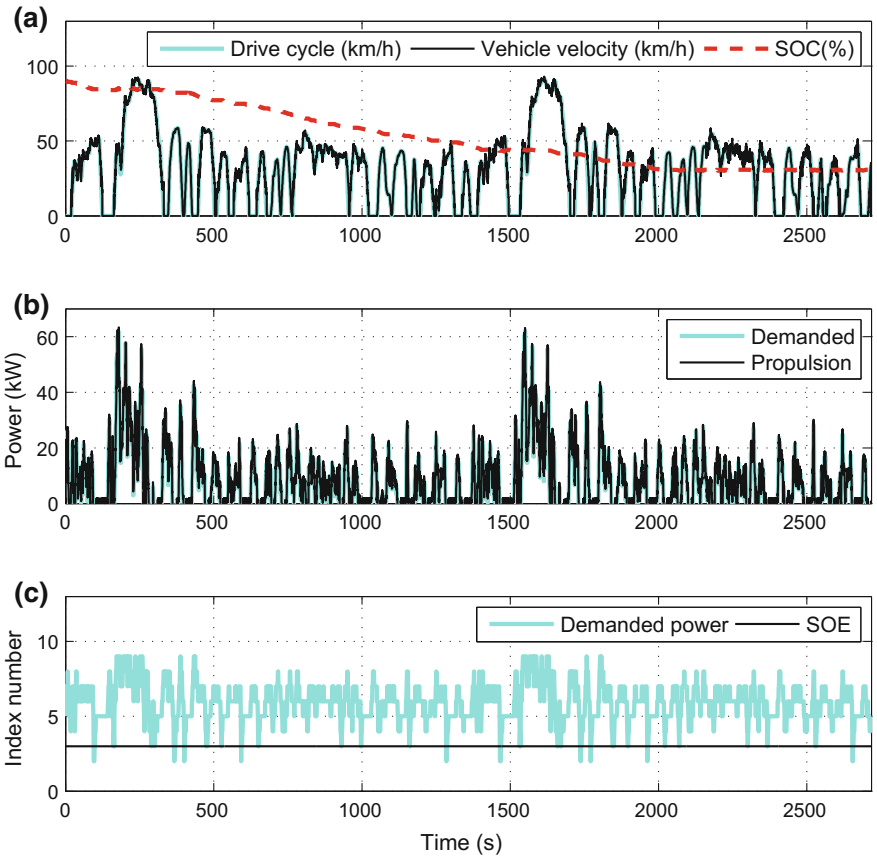


Fig. 5.16 Charge depletion/charge sustenance strategy (**a**) Vehicle velocity and battery depletion profile (**b**) Demanded and propulsion power (**c**) Demanded power and SOE indices

drive and engine low-level controls. By applying the low-level controls commands to each component of the powertrain model, SOC, E, V_{batt}, SOC_{ref}, and E_{ref} are measured at each 1 ms and passed to the ECU via CAN bus.

In Fig. 5.16a, the vehicle drivability performance and battery state of charge for CDCS strategy are demonstrated. In Fig. 5.16b, we can see that the driver's demanded power is followed by propulsion power. This shows that the powertrain is able to provide the required propulsion power, so the vehicle velocity can follow the predefined UDDS schedule. Figure 5.16c shows the index of demanded power as well as SOE index which are determined according to the number of polytopes set. Figure 5.16 shows these results for the blended mode strategy. Note that the engine operation has reduced the battery SOC depletion slope which results in better fuel economy as compared to CDCS strategy.

Table 5.2 eMPC MIL and HIL test using low-fidelity powertrain model: fuel economy for different control strategies

Control strategy	MIL (MPG)	HIL (MPG)
Charge depletion/charge sustenance	119	116
Linear blended mode	133	127

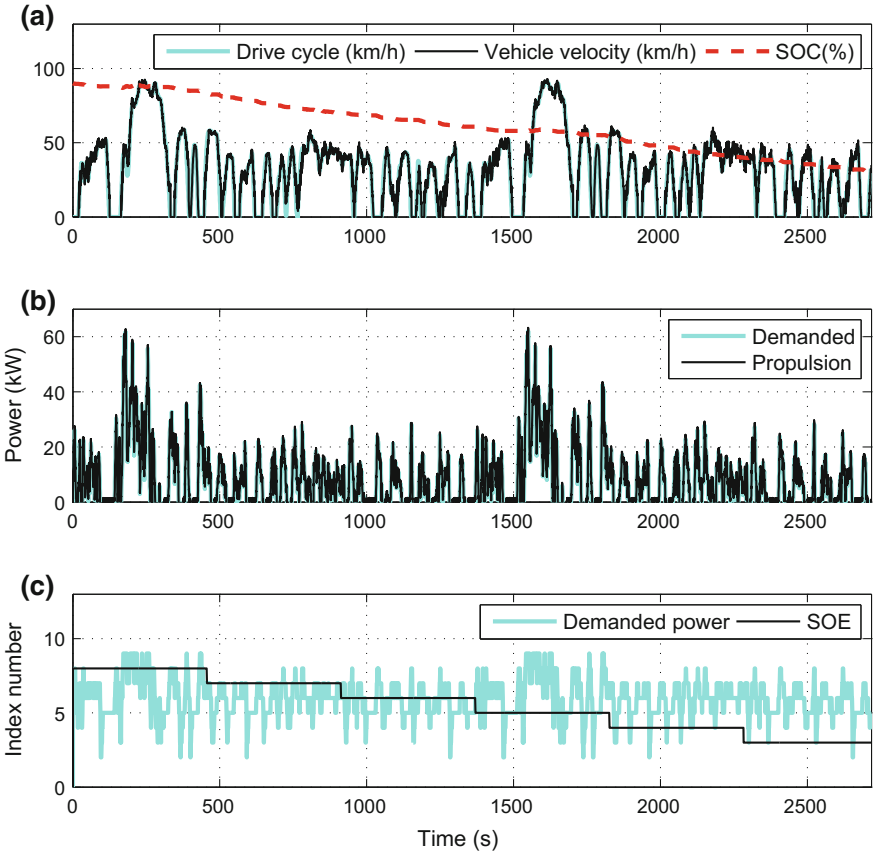


Fig. 5.17 Blended mode strategy (**a**) Vehicle velocity and battery depletion profile (**b**) Demanded and propulsion power (**c**) Demanded power and SOE indices

Table 5.2 shows the HIL fuel economy for applying eMPC EMS to low-fidelity powertrain model.

Note that if we use the same controller and simulation model for MIL and HIL test, the simulation results should be the same. By comparing the results in this section to what was discussed in Sect. 5.4, some discrepancies are seen in terms of vehicle drivability and fuel economy. The oscillations of the vehicle velocity shown

in Figs. 5.16 and 5.17 as compared to Figs. 5.9 and 5.10, is due to switching between different polytope sets found by considering 9 divisions for E_{ref} and SOE_{ref} along the drive cycle. Fuel economy for CDCS and blended mode strategies in HIL testing are worsened by 2.5 and 4.5% as compared to MIL test. This error is due to replacing the eMPC data base and its search algorithm with the approximated look-up tables. As a measure of drivability performance, eMPC EMS can follow the designated drive cycle with the root mean square error of 0.91 and 1.12 km/h for CDCS and Blended mode strategies, respectively. In brief, the difference between MIL and HIL simulation results is due to the difference between the originally designed and modified CRPE-eMPC EMS.

5.6 Summary

In this chapter, the explicit model predictive control approach was used to design an energy management strategy for a plug-in hybrid powertrain. In this way, a new control-oriented model was proposed with two state variables. We implemented the developed controller to a PHEV simulation model and reduced the simulation time by 44% and improved fuel economy by 16% on average in comparison to MPC. According to the model-in-the-loop simulation results, the designed eMPC energy management can be applied to the high-fidelity simulation model three times faster that its implicit MPC counterpart, while maintaining the expected performance.

References

1. Kouramasa, K.I., Panosa, C., Fascab, N.P., Pistikopoulos, E.N.: An algorithm for robust explicit/multi-parametric model predictive control. Automatica **49**, 381–389 (2013)
2. Di Cairano, S., Liang, W., Kolmanovsky, I.V., Kuang, M.L., Phillips, A.M.: Engine power smoothing energy management strategy for a series hybrid electric vehicle. In: American Control Conference (ACC), pp. 2101–2106 (2011)
3. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N.: The explicit linear quadratic regulator for constrained systems. Automatica **38**, 3–20 (2002)
4. Taghavipour, A., Azad, N.L., McPhee, J.: Real-time predictive control strategy for a plug-in hybrid electric powertrain. Mechatronics **29**, 13–27 (2015)
5. Borrelli, F.: Constrained Optimal Control Of Linear And Hybrid Systems. Lecture Notes in Control and Information Sciences, vol. 290. Springer (2003)
6. Tondel, P., Johansen, T.A., Bemporad, A.: An algorithm for multiparameteric quadratic programming and explicit mpc solution. Automatica **39**, 489–497 (2003)
7. Geyer, T., Torrisi, F.D., Morari, M.: Optimal complexity reduction of polyhedral piece-wise affine systems. Automatica **44**, 1728–1740 (2008)
8. Alessio, A., Bemporad, A.: A survey on Explicit Model Predictive Control. Lecture Notes in Control and Information Sciences, vol. 384, pp. 345–369 (2009)
9. Johansen, T.A.: On multi-parametric nonlinear programming and explicit nonlinear model predictive control. In: 41th IEEE Conference on Decision and Control (CDC), pp. 2768–2773 (2002)

10. Johansen, T.A.: Approximate explicit receding horizon control of constrained nonlinear systems. Automatica **40**, 293–300 (2004)
11. Lincoln, B., Rantzer, A.: Relaxing dynamic programming. IEEE Trans. Autom. Control **51**, 1249–1260 (2006)
12. Fukuda, K.: Polyhedral computation faq (2000). http://www.ifor.math.ethz.ch/staff/fukuda/
13. Cheng, S.W., Janadan, A.: Algotithms for ray-shooting and intersection searching. J. Algorithms **13**, 670–692 (1992)
14. Kvasnica, M.: Efficient software tools for control and analysis of hybrid systems. Ph.D dissertation, ETH Zurich (2008)
15. Baotic, M., Borrelli, F., Bemporad, A., Morari, M.: Efficient on-line computation of constrained optimal control. SIAM J. Control Optim. **47**, 2470–2489 (2008)
16. Tndel, P., Johansen, T.A., Bemporad, A.: Evaluation of piecewise affine control via binary search tree. Automatica **39**, 945–950 (2003)
17. De Berg, M., Schwarzkopf, O., Van Kreveld, M., Overmars, M.: Computational Geometry: Algorithms and Applications, 2nd edn. Springer (2000)
18. Kvasnica, M., Grieder, P., Baotic, M.: Multi-parametric toolbox (MPT) (2004). Available from: http://control.ee.ethz.ch/mpt/
19. Slotine, J.J., Li, W.: Applied Nonlinear Control. Prentice Hall (1991)
20. Ferrari-Trecate, G., Cuzzola, F.A., Mignone, D., Morari, M.: Analysis of discrete-time piecewise affine and hybrid systems. Automatica **38**, 2139–2146 (2002)
21. Liberzon, D.: Switching in Systems and Control. Birkhauser Boston (2003)

Chapter 6
Control-Relevant Parameter Estimated Strategy

The real-time applicability of a model predictive controller is closely correlated to the simplicity of the control-oriented model. Simple models are not as precise at capturing the principal dynamics of the main plant. As such, it is preferred that simple model parameters are projected in a manner that attains all dynamics of the plant. Nevertheless, locating the parameters required in the control-oriented model to cover all of the operating points of the plant may not be feasible.

Alternatively, parameter identification within a particular range of the plant operating points can be done, which is indispensable according to control design requirements. The derived control-oriented model is only valid for an active frequency range required by the controls. The control-oriented model may not denote the plant for the entire range of its operating points, but its simplicity and accuracy are optimal in the process of passing the real-time application requirements needed for designing predicting controls.

6.1 Control-Relevant Parameter Estimation (CRPE)

Parameter estimation is a critical step in modeling. Provided that the anticipated purpose of the model is control system design, control relevance in parameter estimation represents the optimal selection of design variables in the estimation algorithm. It is based on the premise that if the bias distribution is properly shaped, the low-order representations of the plant will be obtained, capturing all necessary plant dynamics for control system design [1].

A. Taghavipour et al., *Intelligent Control of Connected Plug-in Hybrid Electric Vehicles*,
Advances in Industrial Control, https://doi.org/10.1007/978-3-030-00314-2_6

6.1.1 Battery Thevenin Model

As previously a stated, the performance of the model predictive controller is closely connected to the accuracy of the control-oriented model inside it. As such, a system to enhance PHEV control-oriented model accuracy is presented, while maintaining the size of eMPC lookup tables minimal for real-time implementation.

A more comprehensive model of the component inside the control-oriented model is considered, because the battery packs role is essential in determining the full electric range and PHEV fuel economy and emission performance.

The preceding control-oriented model (5.1) contains two state variables. An additional extra state variable is added to the previous model.

One added state variable belongs to the battery component. In the new control-oriented model, we replace the power-based model of the battery with the Thevenin's equivalent model of it (Fig. 6.1).

According to the definition of battery state of charge:

$$S\dot{O}C = -\frac{i_{batt}}{Q} = -\frac{1}{R_{batt}Q}(V_{oc} - V - V_{batt}) \tag{6.1}$$

where Q and V are battery pack capacity and the voltage across RC-component of Fig. 6.1, respectively. One can find a differential equation for V as:

$$\dot{V} = -\frac{V}{RC} + \frac{1}{R_{batt}C}(V_{oc} - V - V_{batt}) \tag{6.2}$$

The control input in (6.1) and (6.2) is V_{batt}. On the other hand, we find an expression which can relate V_{batt} to P_{BAT} as battery power is directly related to the total propulsion power. Therefore,

$$P_{BAT} = i_{batt}V_{batt} = \frac{V_{batt}}{R_{batt}}(V_{oc} - V - V_{batt}) \tag{6.3}$$

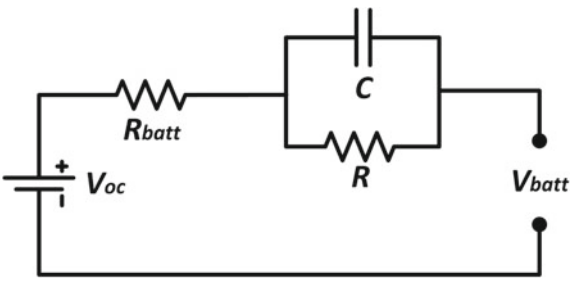


Fig. 6.1 Thevenin's equivalent model of the battery

which can be rewritten as:

$$V_{batt} = \frac{1}{2}\{(V_{oc} + V) - \sqrt{(V_{oc} + V)^2 + 4R_{batt}P_{BAT}}\} = f(V, P_{BAT}) \tag{6.4}$$

In order to get a linear control-oriented model, $f(V, P_{batt})$ can be linearized within operating range of V and P_{BAT}. If $f(V, P_{BAT}) = b_1 V + b_2 P_{BAT} + b_3$, the battery control-oriented model can be written as

$$\begin{aligned} S\dot{O}C &= -\tfrac{1}{R_{batt}Q}\{V_{oc} - b_3 - (1 - b_2)V - b_2 P_{BAT}\} \\ \dot{V} &= -\tfrac{V}{RC} + \tfrac{1}{R_{batt}C}\{V_{oc} - b_3 - (1 - b_2)V - b_2 P_{BAT}\} \end{aligned} \tag{6.5}$$

Equation (6.5) in combination with (6.6) makes the new control-oriented model with three state variables

$$\dot{E} = P_{BAT} + P_{ENG} - P_{BRK} \tag{6.6}$$

In order to get a more accurate behavior of the battery, the four parameters of Thevenin's equivalent model of battery should be estimated at this stage, according to controls requirements [2, 3]. The next section will focus on this.

6.1.2 Battery Parameters Estimation

This section focuses on estimating the battery parameters within the operating range of the EMS. The battery active frequency range is found based on a large number of simulations already completed. Accordingly, the experimental data required for parameter estimation results from that particular frequency range.

The new control-oriented model can be established as the battery model parameters are identified, which is followed by a novel eMPC EMS design. The new EMS is anticipated to perform better than the previous one.

Considering battery power and state of the charge as input and output to the model, respectively, a global parameter estimation scheme is needed to assess the parameters for the equivalent battery model. To this end, the power associated with the high-fidelity battery model along with its state of charge are estimated using eMPC, assuming a UDDS driving cycle for the PHEV.

As illustrated in Fig. 6.2, the power signal contains high-frequency oscillations, making the parameter estimation process much more difficult.

A dominant frequency range of the power signal can be extracted through a spectrum analysis in order to expedite the identification process. A low-pass filter should be designed afterward to regenerate the model signal.

Based on both Welch and Fast Fourier Transform (FFT) power spectral estimate (as shown in Fig. 6.3), the frequencies higher than 5 Hz have less contribution in representing the battery power signal.

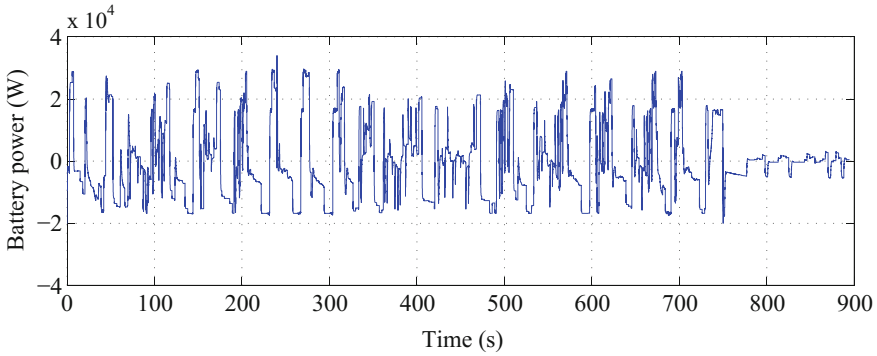


Fig. 6.2 Time history of the battery power, applying eMPC to the PHEV for UDDS driving cycle

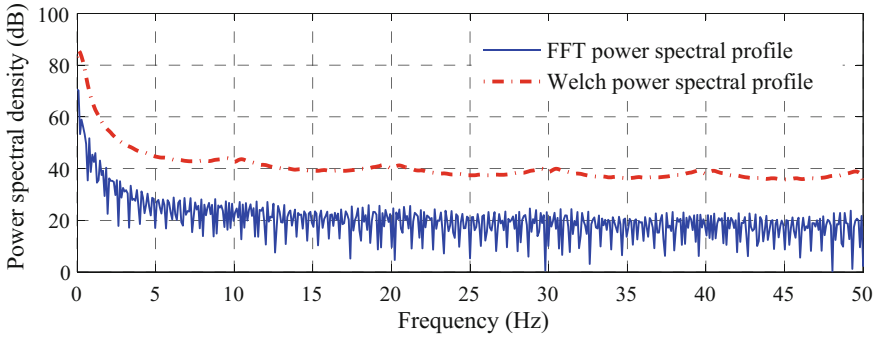


Fig. 6.3 Power spectral density analysis of the chemistry-based battery power signal, when an eMPC scheme is applied to the PHEV

Consequently, a digital filter with finite-duration impulse response (least square approaches, etc.), appears to be a promising candidate, by which the refined power signal to the battery can be regenerated.

The state of charge is then extracted, applying the refined power signal to the high-fidelity battery model. Assuming the acquired input–output to the battery as the reference data, the parameters for the equivalent circuit model can be estimated [4].

A dual-polarization equivalent circuit model with a double-RC circuit is utilized in this work (Fig. 6.4), with two essential physical phenomena in lithium-ion batters incorporated in the equivalent system.

In terms of dynamic performance and state-of-the-charge estimation, the model has been experimentally assessed for lithium-ion batteries, and it contains polarizations of different time scales [5].

Although Thevenin's equivalent model is suitable for control-oriented modeling, it is not capable of representing the physics of lithium-ion battery as effectively as a dual-polarization model, and based on this, the parameters of the dual-polarization model are estimated first. This is followed by the approximation of

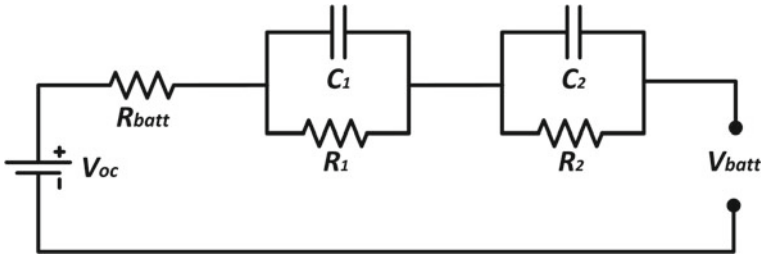


Fig. 6.4 Dual-polarization model of the battery

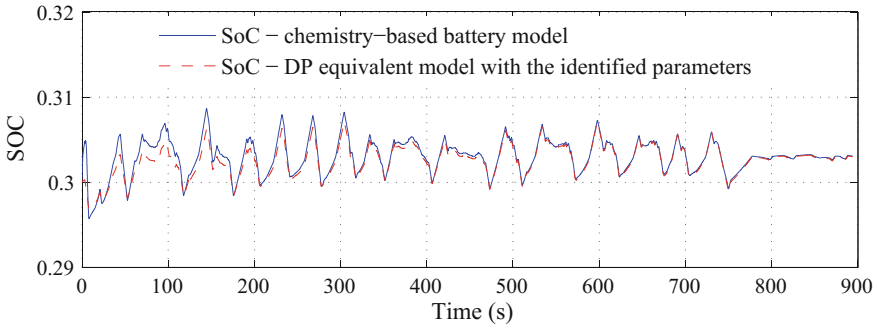


Fig. 6.5 Time history of the SOC profile for the dual-polarization equivalent and chemistry-based battery model

the dual-polarization model by a Thevenin's equivalent model. In other words, the impendence of two RC circuits are approximated with only one RC circuit:

$$\frac{R_1}{R_1C_1s+1}+\frac{R_2}{R_2C_2s+1}\cong\frac{R}{RCs+1} \tag{6.7}$$

Figure 6.5 illustrates the simulation results for the state of charge of the equivalent battery model, as compared to the experimental full chemistry-based battery model derivations.

A great match signifies the efficacy of the homotopy optimization procedure in identifying the model parameters, which was used earlier in validating the PHEV simulation model.

A homotopy gain of 200 ($K_i, i = 1$) and a decremental step of 0.25 in the homotopy parameter ($\Delta\upsilon$) show a reasonable match between the original data and the results obtained from the dual-polarization model with identified parameters. The value for the identified parameters is listed in Table 6.1.

Table 6.1 Identified values for the parameters of the equivalent dual-polarization circuit model

Parameter	Parameter description	Identified value (unit)
V_{oc}	Open-circuit voltage	388.6 (V)
R_{batt}	Battery resistance	0.31 (Ω)
$R_c(R_1)$	Concentration polarization resistance	0.13 (Ω)
$R_e(R_2)$	Electrochemical polarization resistance	0.052 (Ω)
$\tau_c(R_1C_1)$	Concentration polarization time constant	101.5 (s)
$\tau_e(R_2C_2)$	Electrochemical polarization time constant	11.3 (s)

6.1.3 CRPE Control-Oriented Model

We found the battery model parameters as $V_{oc} = 388.64$ V, $R_{batt} = 0.3147\Omega$, $Q_{batt} = 75428$ A.s, $R = 0.0831\Omega$, and $C = 168.5173$ F, so the discrete version of the new control-oriented model is

$$\bar{Z}(k+1) = \bar{\mathbf{A}}\bar{Z}(k) + \bar{\mathbf{B}}U(k) + \bar{F} \tag{6.8}$$

where $\bar{Z} = \left[\, SOC,\ V_{batt},\ E \,\right]^T$ and $U = \left[\, P_{BAT},\ P_{ENG},\ P_{BRK} \,\right]^T$.

$$\bar{\mathbf{A}} = \begin{bmatrix} 1 & \bar{a}_1 & 0 \\ 0 & \bar{a}_2 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$
$$\bar{\mathbf{B}} = \begin{bmatrix} \bar{a}_3 & 0 & 0 \\ \bar{a}_4 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$
$$\bar{\mathbf{F}} = \begin{bmatrix} \bar{a}_5 \\ \bar{a}_6 \\ 0 \end{bmatrix} \tag{6.9}$$

Note that SOC, V, and E are in percent, volt, and kWs, respectively. In the new control-oriented model, we use battery state of charge instead of battery state of energy.

6.2 CRPE-eMPC Energy Management Polytopes

We can rewrite (6.10) in the way of (5.3) but here we added one parameter to the multiparametric quadratic programming problem. So the new problem has five parameters, because of one added state variable in the new control-oriented model. Now,

the control-relevant parameter estimated eMPC (CRPE-eMPC) energy management can be designed.

$$\begin{gathered}\min_U \sum_{j=1}^{N_p} \left\{ \bar{Y}^T(j)\bar{\mathbf{Q}}\bar{Y}(j) + U^T(j)\bar{\mathbf{R}}U(j) \right\} \\ s.t. \\ \begin{bmatrix} SOC_{min} \\ V_{batt,min} \\ E_{min} \end{bmatrix} \leq \begin{bmatrix} SOC \\ V_{batt} \\ E \end{bmatrix} \leq \begin{bmatrix} SOC_{max} \\ V_{batt,max} \\ E_{max} \end{bmatrix} \\ U_{min} \leq U \leq U_{max} \end{gathered} \tag{6.10}$$

where $\bar{Y} = \left[SOC - SOC_{ref},\ E - E_{ref} \right]^T$, and N_p is the prediction horizon length.

$$\bar{\mathbf{Q}} = \begin{bmatrix} \bar{\omega}_1 & 0 \\ 0 & \bar{\omega}_2 \end{bmatrix}$$

$$\bar{\mathbf{R}} = \begin{bmatrix} \bar{\omega}_3 & 0 & 0 \\ 0 & \bar{\omega}_4 & 0 \\ 0 & 0 & \bar{\omega}_5 \end{bmatrix}$$

6.2.1 CRPE-eMPC Controls Regions

By solving the multiparametric quadratic problem, the number of polytopes can be seen in Fig. 6.6:

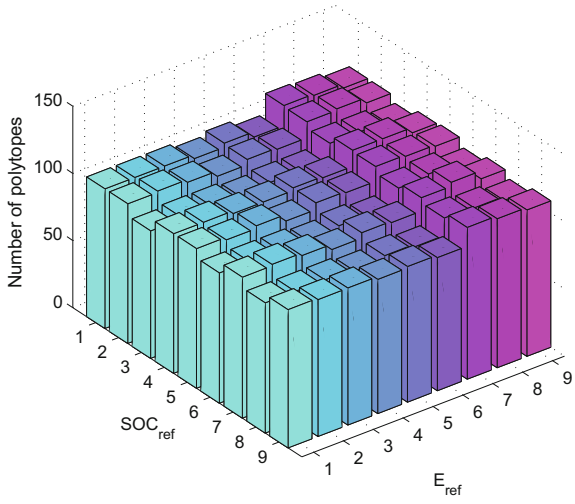


Fig. 6.6 Number of polytopes for different levels of E_{ref} and SOC_{ref}

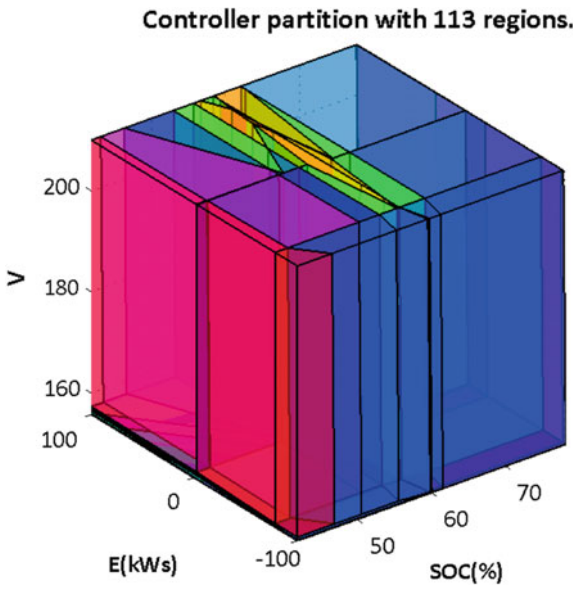


Fig. 6.7 Polytope set for $E_{ref} = 0$ and $SOC_{ref} = 60\%$

Total number of polytopes is 8439. The resultant polytopes set for $E_{ref} = 0$ and $SOC_{ref} = 60\%$ is demonstrated in Fig. 6.9.

As it is seen, there is one added dimension to the polytope set in Fig. 6.7 as compared to Fig. 5.3. For any given V_{batt}, any cross section which contains $(SOC - E)$ generally looks similar to Fig. 5.3.

In this case, we cannot visualize the control actions like Fig. 5.4. But for any cross section of Fig. 5.14 for any given V_{batt}, the analysis on E and SOC remains valid. Figure 6.8 shows the control actions versus different measured values of SOC and E (initial conditions) at current sampling instance for $V_{batt} = 180V$.

6.2.2 CRPE-eMPC Stability Notes

Here, we can exactly follow what is mentioned in Sect. 5.3. In each set of polytopes which belongs to a definite SOC_{ref}, E_{ref}, the controller drives the state variables to the mentioned reference values in finite time steps. To prove the local stability of the closed-loop system, we pick the polytope which contains $\bar{Z}_0$. The control corresponding to that polytope is:

$$\hat{\bar{U}} = \bar{f}_0 \hat{\bar{Z}} + \bar{g}_0 \tag{6.11}$$

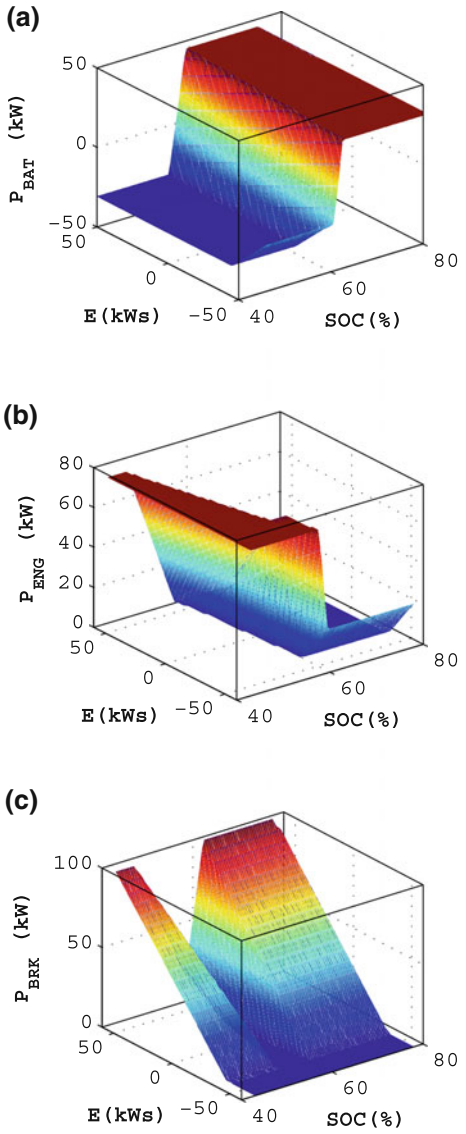


Fig. 6.8 Control actions for $V_{batt} = 180V$, $E_{ref} = 0$, and $SOC_{ref} = 60\%$ based on different initial conditions **a** battery power **b** engine power **c** braking power

By applying the above control to the control-oriented model, we can find the closed-loop system equation as:

$$\bar{Z}(k+1) = (\bar{\mathbf{A}} + \bar{\mathbf{B}}\bar{f}_0)\bar{Z}(k) + \bar{\mathbf{B}}\bar{g}_0 \tag{6.12}$$

By defining $\tilde{\bar{Z}} = \bar{Z} - \bar{Z}_0$, we transfer the state variables to the equilibrium point. As a result we have:

$$\begin{aligned}\tilde{\bar{Z}}(k+1) &= (\bar{\mathbf{A}} + \bar{\mathbf{B}}\bar{f}_0)\tilde{\bar{Z}}(k) \\ &+\bar{\mathbf{B}}\bar{g}_0 + (\bar{\mathbf{A}} + \bar{\mathbf{B}}\bar{f}_0 - I_{2\times 2})\bar{Z}_0 = \tilde{\bar{\mathbf{A}}}\tilde{\bar{Z}}(k) + \tilde{\bar{\mathbf{B}}}\end{aligned} \tag{6.13}$$

First, we show that $\tilde{\bar{\mathbf{A}}}$ is locally and asymptotically stable for all 81 sets of polytopes. We have a discrete switching system and need to make sure that the spectral radius of $\tilde{\bar{\mathbf{A}}}$ is less than unity. Also, $\tilde{\bar{\mathbf{B}}}$ is bounded, so the closed-loop system (5.34) is locally stable.

On the other hand, by using the cost function over each set of polytopes, it is proven that there is a $\bar{\mathbf{Q}} > 0$ the closed-loop system is globally and exponentially stable:

$$\bar{V}(\tilde{\bar{Z}}) = \sum_{j=1}^{N_p} \{\tilde{\bar{Z}}^T(j)\bar{\mathbf{Q}}\tilde{\bar{Z}}(j)\} > 0 \Rightarrow \Delta\bar{V}(\tilde{\bar{Z}}_{k+1}, \tilde{\bar{Z}}_k) < 0 \tag{6.14}$$

For investigating the stability of the switched system, we assume that (6.14) remains valid and $\bar{\rho}$ is the largest spectral radius of $\tilde{\bar{A}}$ for all 81 sets of polytopes. If $\bar{\zeta}$ is chosen greater than $\bar{\rho}^{-1}$, the switched system will be stable.

$$\begin{bmatrix} SOC_{ref}(k+1) \\ E_{ref}(k+1) \end{bmatrix} = \begin{bmatrix} 1/\bar{\zeta} & 0 \\ 0 & 1/\bar{\zeta} \end{bmatrix} \begin{bmatrix} SOC_{ref}(k) \\ E_{ref}(k) \end{bmatrix} + \Gamma \tag{6.15}$$

6.3 CRPE-eMPC Performance Simulation

In this part, the CRPE-eMPC EMS is applied to the high-fidelity simulation model considering the low-level controls (see Sect. 4.6). The results of MIL simulation are reviewed in Figs. 6.9, 6.10, 6.11, 6.12, 6.13, and 6.14.

6.3.1 No Knowledge of Trip Information

Figure 6.9a shows the vehicle drivability performance along two UDDS drive cycles for the CDCS strategy. Figure 6.9b, c show the performance of low-level controls in tracking the setpoints determined by CRPE-eMPC EMS. Figure 6.10 demonstrates the emissions control performance. The sliding mode controller keeps the HC conversion efficiency around unity by controlling the air–fuel ratio and ignition timing as shown in Fig. 6.11. PHEV fuel consumption is reduced to 1.97 l/100 km (119 MPG) by using the CRPE-eMPC EMS.

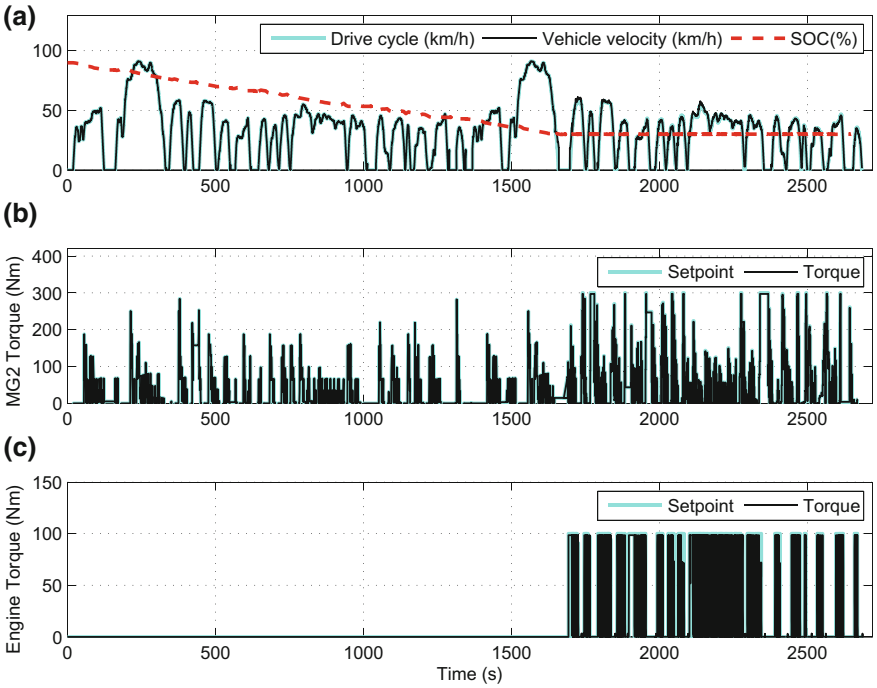


Fig. 6.9 CDCS CRPE-eMPC strategy with emission control: **a** Velocity and battery SOC **b** MG2 torque **c** Engine torque

6.3.2 Known Traveling Distance

Figures 6.12, 6.13, and 6.14 show the results for the blended mode strategy. Fuel consumption in this simulation is 1.68 l/100 km (140 MPG) by considering engine emissions control.

6.3.3 Discussions

Fuel economy for MIL testing by using CRPE-eMPC EMS is reviewed in Table 5.3. It shows that CRPE-eMPC EMS improves fuel economy by 5.31% and 7.69% for CDCS and blended mode strategies, respectively, in comparison to the eMPC high-level controller which was designed based on a control-oriented model with two state variables. The MIL simulation can still be performed 1.5 times faster than

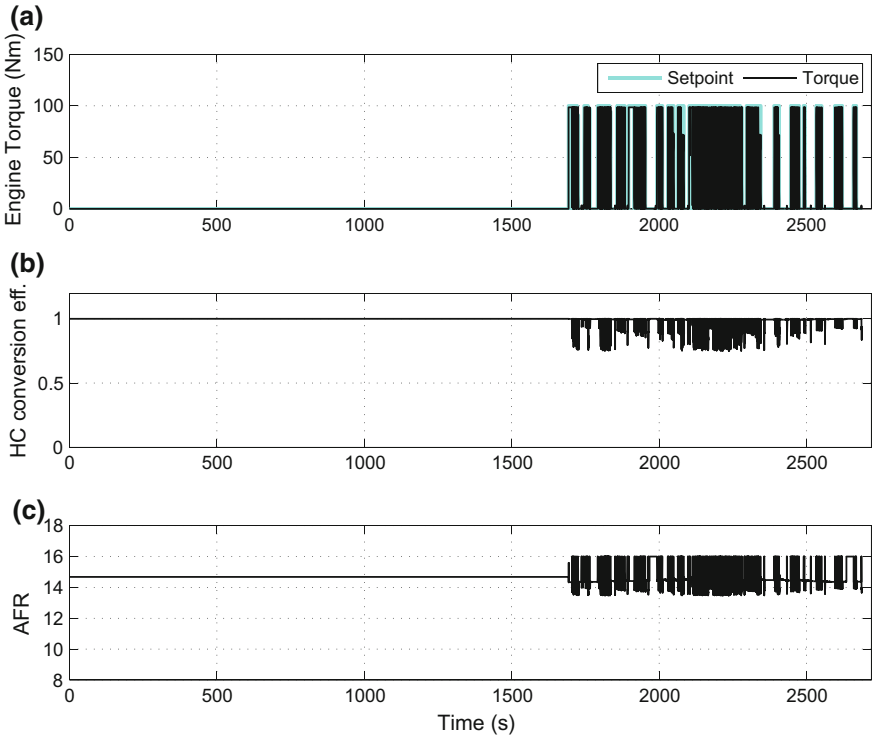


Fig. 6.10 CDCS CRPE-eMPC strategy with emission control: **a** Engine torque **b** HC conversion efficiency **c** Air-to-fuel ratio (AFR)

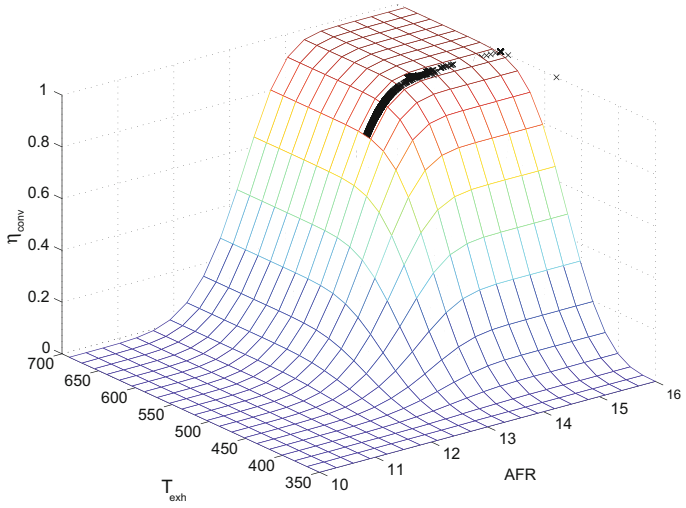


Fig. 6.11 Catalyst conversion efficiency for CDCS CRPE-eMPC strategy

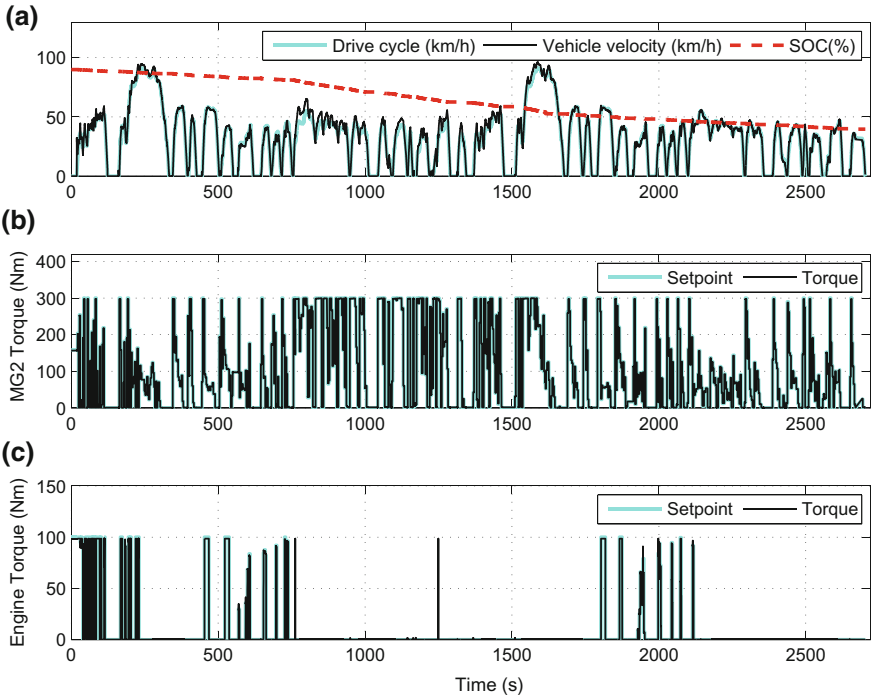


Fig. 6.12 Blended CRPE-eMPC strategy with emission control: **a** Velocity and battery SOC **b** MG2 torque **c** Engine torque

real time. Therefore, adding one state variable to the previous control-oriented model (Sect. 5.2.1) does not slow down the simulation. In fact, the performance of the controller is also improved in terms of fuel economy while maintaining the emissions performance. As a measure of drivability performance, CRPE-eMPC EMS can follow the designated drive cycle with the root mean square error of 0.45 km/h and 0.97 km/h for CDCS and blended mode strategies, respectively (Table 6.2).

Figure 6.15 shows why the fuel economy has been improved by using CRPE-eMPC approach for designing EMS. More accurate battery model inside the control-oriented model has led to greater P_{BAT} which has reduced the P_{ENG} and consequently reduced fuel consumption.

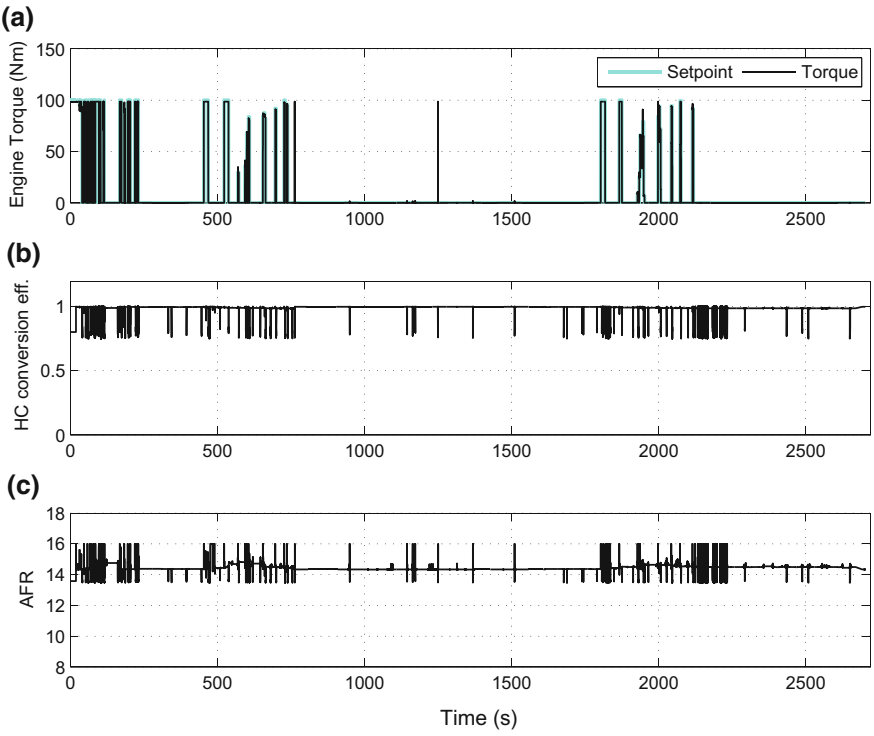


Fig. 6.13 Blended CRPE-eMPC strategy with emission control: **a** Engine torque **b** HC conversion efficiency **c** Air-to-fuel ratio (AFR)

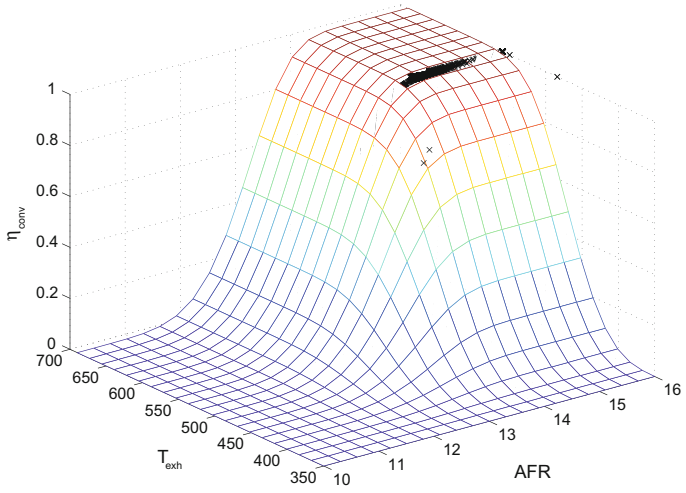


Fig. 6.14 Catalyst conversion efficiency for Blended CRPE-eMPC strategy

Table 6.2 MIL with the high-fidelity powertrain model: Fuel economy for different control strategies

Control strategy	eMPC (MPG)	CRPE-eMPC (MPG)	improvement (%)
Charge depletion/Charge sustenance	113	119	5.31
Linear blended mode	130	140	7.69

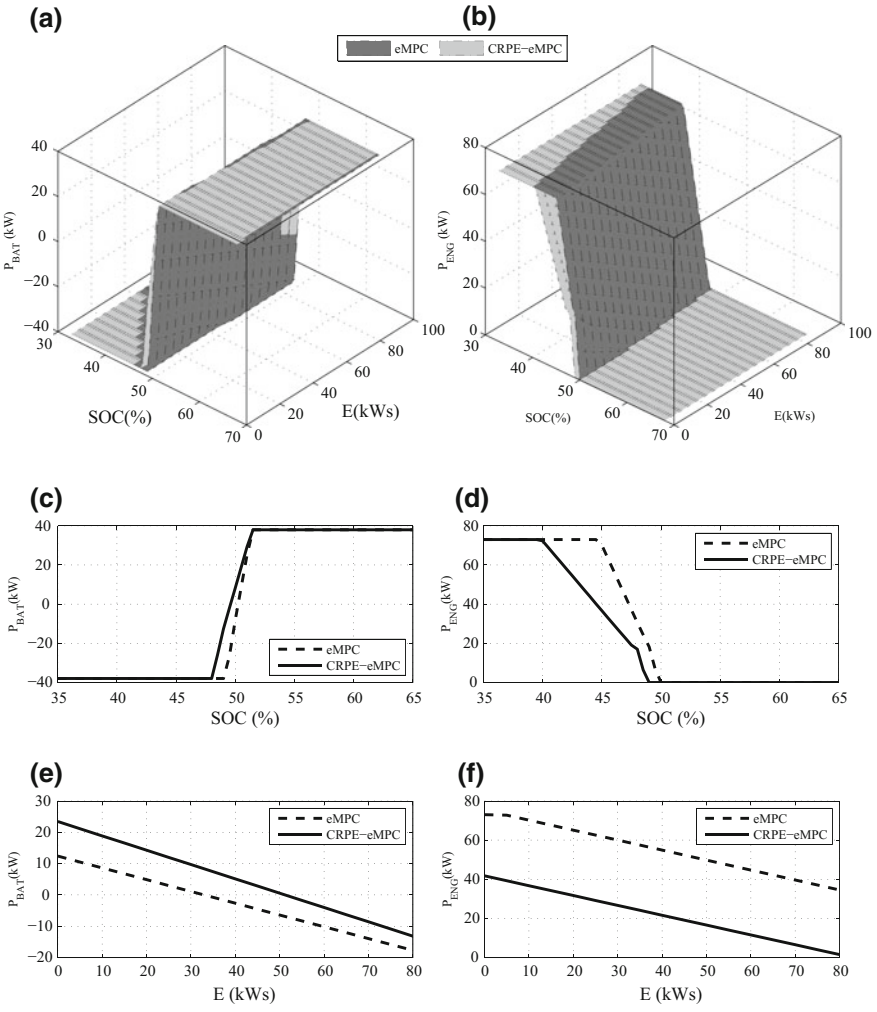


Fig. 6.15 eMPC versus CRPE-eMPC: **a** Battery power **b** Engine power **c** Battery power for constant $E = 10$ kWs **d** Engine power for constant $E = 10$ kWs **e** Battery power for constant $SOC = 50\%$ **e** Engine power for constant $SOC = 45\%$

6.4 CRPE-eMPC Performance Benchmarking via HIL

Figure 6.16a shows the vehicle drivability performance along two UDDS drive cycles for CDCS strategy. The performance of the low-level controls in tracking the CRPE-eMPC EMS setpoints is shown in Fig. 6.16b, c. The emissions control performance is demonstrated in Figs. 6.17 and 6.18. PHEV fuel consumption is reduced to 2.06 l/100 km by using CRPE-eMPC EMS.

Figures 6.19, 6.20, and 6.21 show the results for blended mode strategy. Fuel consumption in this simulation is 1.73 l/100 km by considering engine emissions control.

Table 6.3 shows the HIL predicted fuel economy for applying CRPE-eMPC EMS to the high-fidelity powertrain model.

Presumably, the results of MIL test which was discussed in Sect. 6.3 should be the same as what we get from HIL test. But fuel economy for CDCS and blended mode strategies in HIL testing are worsened by 5 and 3.6% as compared to the MIL test.

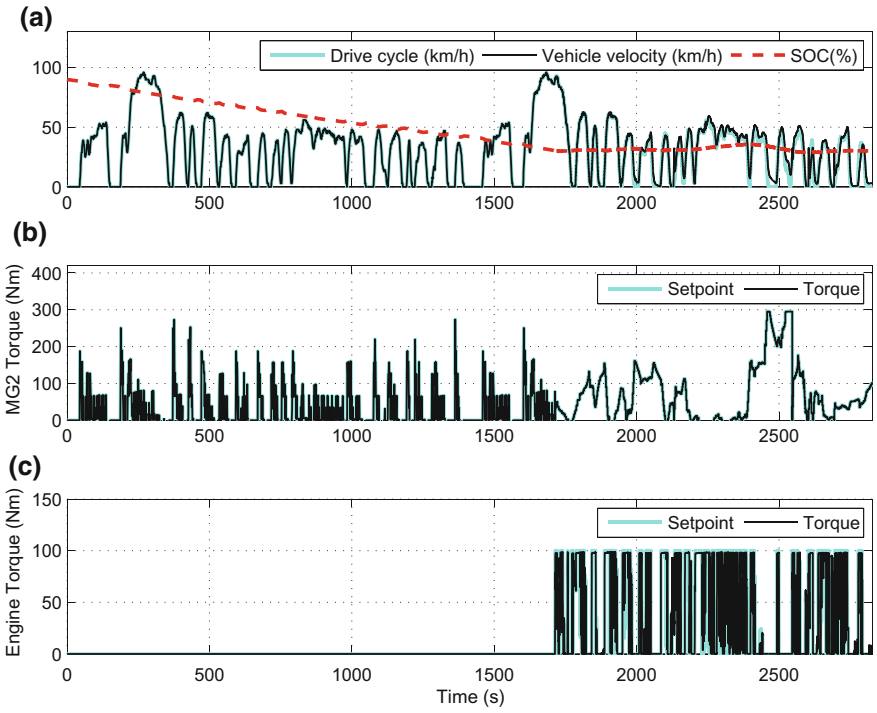


Fig. 6.16 HIL simulation, CDCS CRPE-eMPC strategy with emission control: **a** Velocity and battery SOC **b** MG2 torque **c** Engine torque

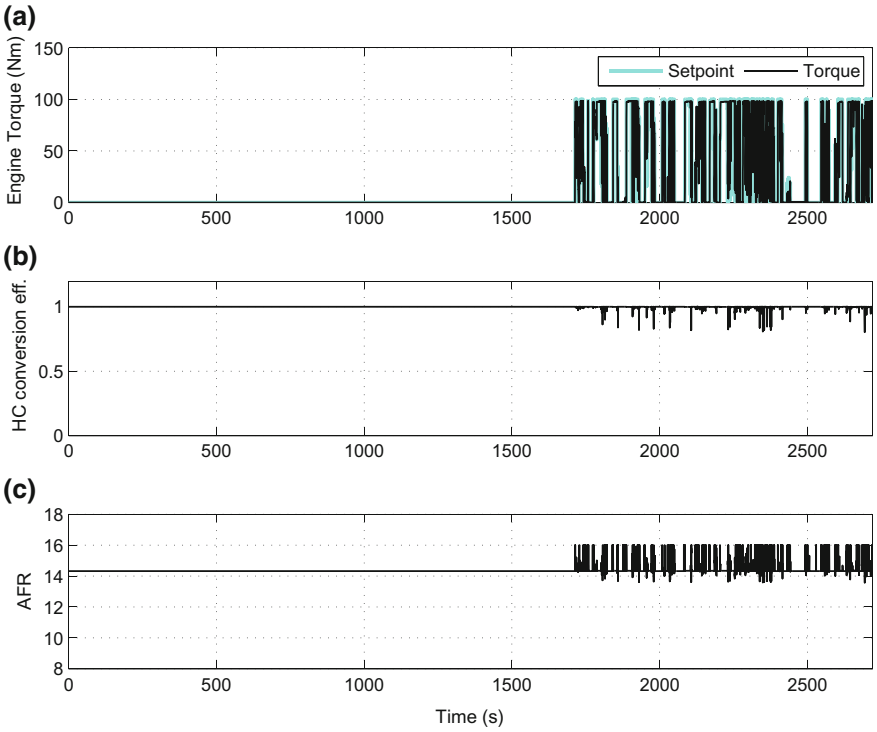


Fig. 6.17 HIL simulation, CDCS CRPE-eMPC strategy with emission control: **a** Engine torque **b** HC conversion efficiency **c** Air-to-fuel ratio (AFR)

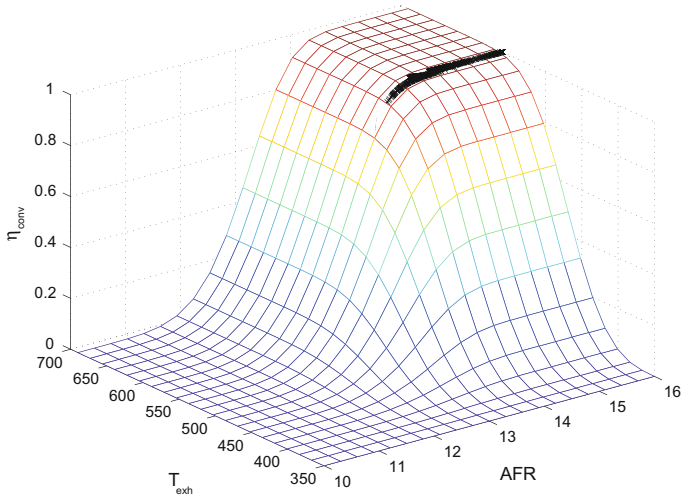


Fig. 6.18 HIL simulation, catalyst conversion efficiency for CDCS CRPE-eMPC strategy

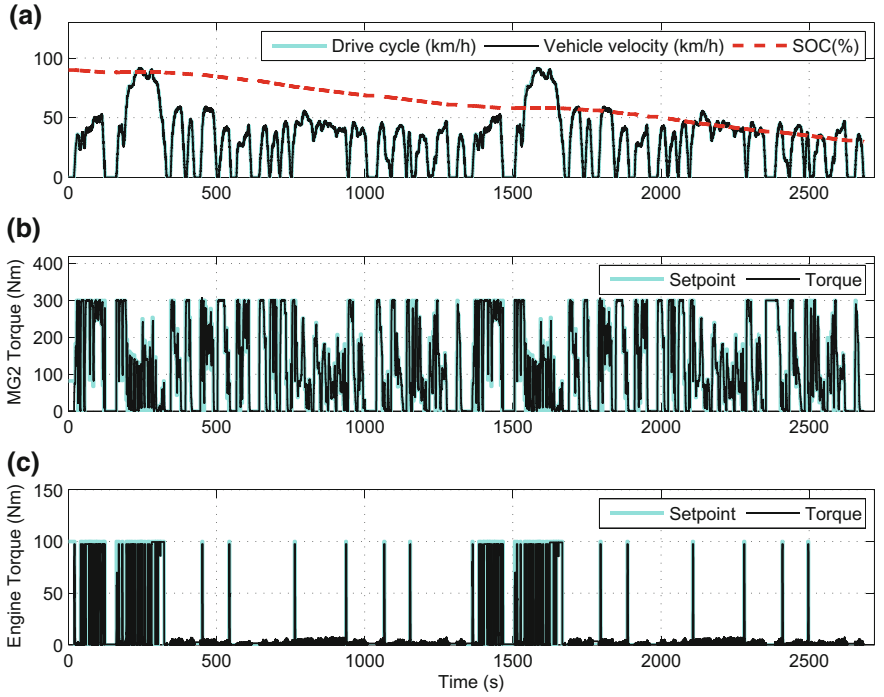


Fig. 6.19 HIL simulation, blended CRPE-eMPC strategy with emission control: **a** Velocity and battery SOC **b** MG2 torque **c** Engine torque

As mentioned earlier, the current controller has been changed in comparison to the EMS validated in Sect. 6.3, since the CRPE-eMPC controller was modified in order to implement the EMS into MotoTron hardware. Note that if the emissions control is not considered in the control scheme, the fuel economy improvement is expected to reach up to 132 MPG and 156 MPG for CDCS and blended mode strategies, respectively.

Table 6.4 summarizes the resulted fuel economy by applying different EMSs to the PHEV high-fidelity model. We can compare the fuel economy with the baseline rule-based EMS of Autonomie software which is 97 MPG. As a measure of drivability performance, CRPE-eMPC EMS can follow the designated drive cycle with the root mean square error of 1.27 km/h and 0.21 km/h for CDCS and blended mode strategies, respectively.

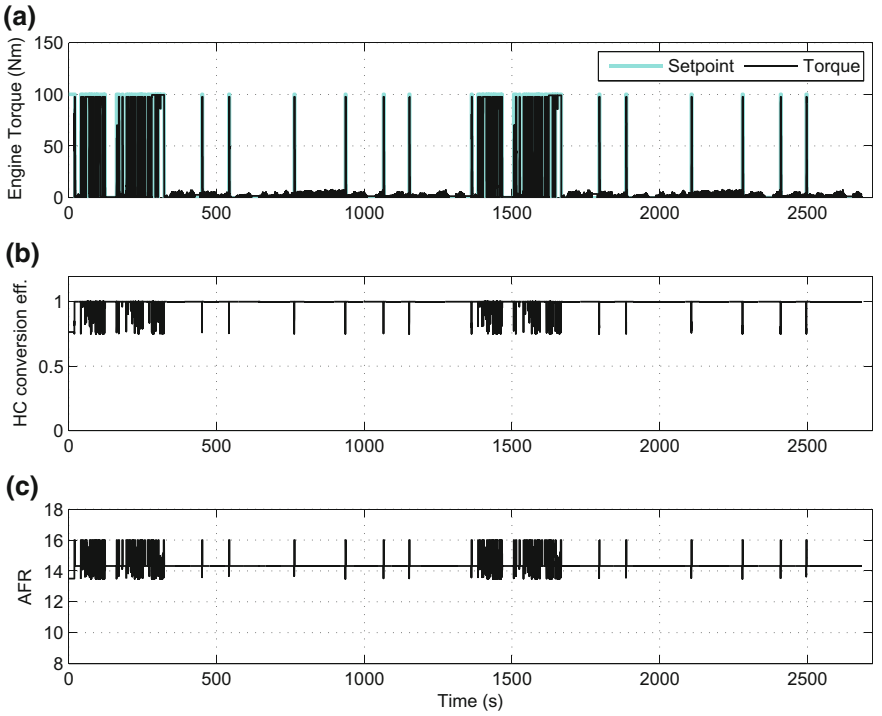


Fig. 6.20 HIL simulation, blended CRPE-eMPC strategy with emission control: **a** Engine torque **b** HC conversion efficiency **c** Air-to-fuel ratio (AFR)

Fig. 6.21 HIL simulation, catalyst conversion efficiency for blended CRPE-eMPC strategy

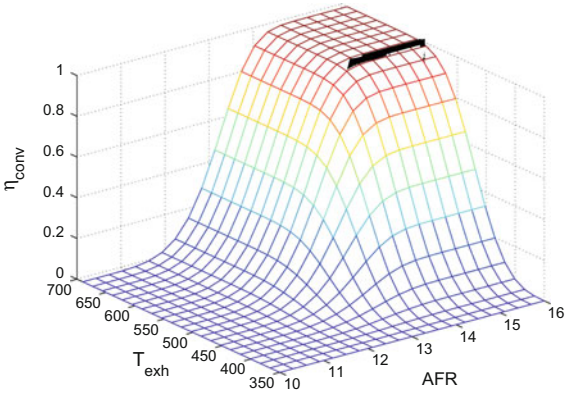


Table 6.3 CRPE-eMPC MIL and HIL test using high-fidelity powertrain model: Fuel economy for different control strategies

Control strategy	MIL (MPG)	HIL (MPG)
Charge depletion/Charge sustenance	119	113
Linear blended mode	140	135

Table 6.4 Fuel economy summary

	Control strategy Fuel economy	CDCS (MPG)	Blended mode (MPG)
MIL	Autonomie rule-based EMS without emissions control	97	–
MIL	MPC EMS without emissions control	128	142
MIL	MPC EMS with emissions control	115	124
MIL	eMPC EMS with emissions control	113	130
MIL	CRPE-eMPC EMS with emissions control	119	140
HIL	CRPE-eMPC EMS with emissions control	113	135

6.5 Summary

In this chapter, the control-relevant parameter estimated model predictive control approach was used to design an energy management strategy for a plug-in hybrid powertrain. In order to make the control-oriented model more accurate which in turn improves the performance of the EMS, control-relevant parameter estimation approach was used. The MIL simulation results showed that by adding one battery state variable to the previous control-oriented model, and designing CRPE-eMPC energy management system, fuel consumption was reduced by up to 6.5% as compared to the eMPC energy management system, and the real-time implementation capabilities were maintained.

Furthermore, the designed EMS performance was validated through HIL test. Select modifications were implemented on the original control scheme in order to apply the EMS to the control hardware with limited memory size and computational capability. HIL simulation demonstrates that the proposed EMS can be implemented to a commercial control hardware in real time, resulting in up to 16.5% fuel economy improvement, compared to the baseline strategy by controlling the emissions. It is important to note that if the emissions control is not considered in the control scheme, the fuel economy enhancement is anticipated to be even higher.

References

1. Rivera, D.E.: Control-relevant parameter estimation: a systematic procedure for prefilter design. In: American Control Conference (ACC) (1991)
2. Taghavipour, A., Masoudi, R., Azad, N.L., McPhee, J.: Control relevant parameter estimation application to a mode-based PHEV power management system. Optim. Control Appl. Meth. **38**, 1148–1167 (2017)
3. Taghavipour, A., Azad, N.L., McPhee, J.: Multi-parametric energy management system with reduced computational complexity for plug-in hybrid electric vehicles. In: European Control Conference (ECC) (2015)
4. Seaman, A., Dao, T., McPhee, J.: A survey of mathematics-based equivalent-circuit and electrochemical battery models for hybrid and electric vehicle simulation. J. Power Sources **256**, 410–423 (2014)
5. He, H., Xiong, R., Fan, J.: Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an experimental approach. Energies **4**, 582–598 (2011)

Part II
Smart Ecological Supervisory Controls

Chapter 7
Real-Time Trip Planning Module Development and Evaluation

This chapter presents the Trip Planning module as a part of the devised energy-optimal controller. This module takes advantage of long-range trip data to optimize SOC profiles. Parts of this chapter are extracted from the author's published papers [1–3].

The proposed powertrain control strategies takes advantage of real-time trip data provided by GPS, GIS, V2V, Radar, and other sensors to predict future driving conditions, and subsequently enable more efficient control system. Figure 7.1 presents the proposed control architecture for the energy-optimal controller of PHEVs. It consists of three main subsystems: the Trip Planning module, Route-based EMS, and Eco-Cruise controller.

The Trip Planning employs trip data to find the optimum SOC profiles by minimizing the total energy cost of the trip, including both fuel and electrical grid energy expenditures, while considering the powertrain constraints. The Route-based EMS uses the optimum SOC profiles to determine the power distribution between the engine and motor-generators, in real-time. Finally, the Eco-Cruise controller calculates the optimum propulsion and braking torques, considering fuel economy and driving safety. Total energy cost, driving safety, and comfort are taken into account to build the energy-optimal control system. Table 7.1 shows the criteria employed in each sub-controller.

There is broad variance in the rates at which trip conditions change. For example, travel paths and traffic data tends to change infrequently and relatively slowly, whereas vehicle speed, power demand, and the working points for both the engine and motor-generator changes often and very quickly. This has concomitant impacts on the desired update rates for the Trip Planning and Route-based EMS. Even though these subsystems are implemented in real-time, the Trip Planning can receive updates at a much slower rate than is required by the Route-based EMS. This chapter describes and presents simulation results outlining the development of the Trip Planning, while the Route-based EMS and Eco-Cruise controller are presented in the Chaps. 8 and 9, respectively.

A. Taghavipour et al., *Intelligent Control of Connected Plug-in Hybrid Electric Vehicles*,
Advances in Industrial Control, https://doi.org/10.1007/978-3-030-00314-2_7

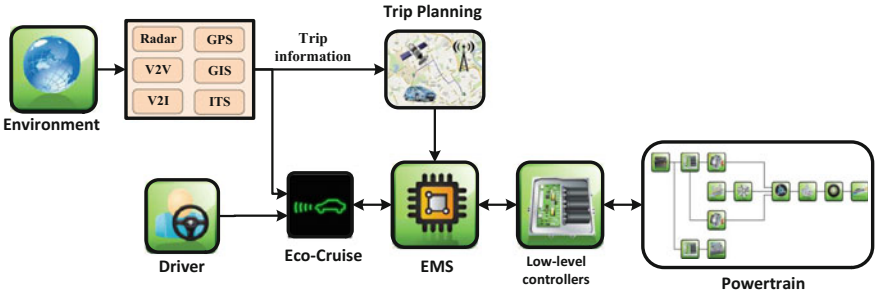


Fig. 7.1 Schematic of the energy-optimal controller

Table 7.1 Energy-optimal control system objectives

System	Safety	Energy cost	Comfort
Trip Planning module		✓	
Route-based EMS		✓	
Eco-Cruise controller	✓	✓	×

✓: direct criteria, ×: indirect criteria

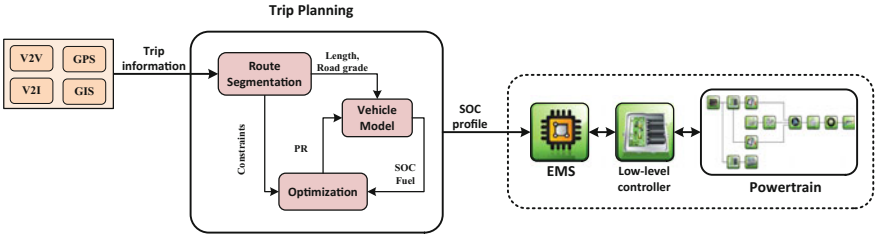


Fig. 7.2 Schematic of the Trip Planning

The Trip Planning optimizes SOC profiles by considering constraints on powertrain components. Figure 7.2 shows a schematic of the controller. First, trip data is preprocessed to build the predicted speed trajectory. Then, the vehicle performance (fuel consumption and *SOC*) is calculated based on the power ratio between the engine and battery. Finally, the optimum SOC profile is obtained using the fast RCO algorithm. Progress towards designing the Trip Planning is described below.

7.1 Online Optimization Model

The high-fidelity model is very detailed and thus too complex and computationally expensive for real-time applications. Therefore, the online optimization model is developed for the Trip Planning module. The inputs of the online optimization model

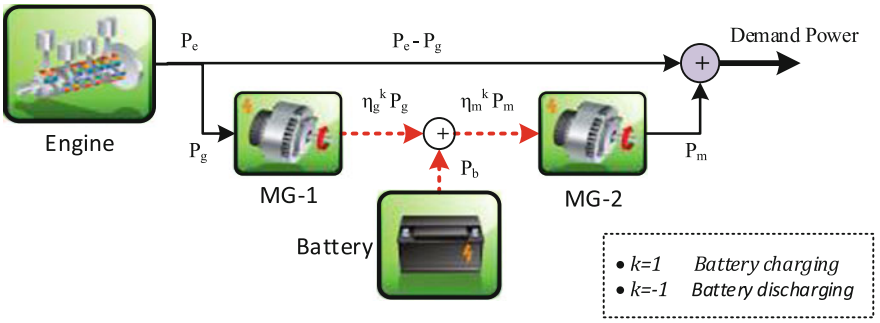


Fig. 7.3 Schematic of the power flow in PHEV powertrain

are trip data and power ratio (PR) between two energy sources. This model predicts future speed trajectory and calculates total energy cost using a simple but sufficiently accurate powertrain model.

In the online optimization model, the powertrain components are represented based on the vehicle longitudinal dynamics:

$$m\dot{v} = \frac{P_d}{v} - \frac{1}{2}\rho A C_d v^2 - mgf\cos\theta + mg\sin\theta \tag{7.1}$$

where P_d is the power demand ($P_d = (m\dot{v} + F_d)\,.v$), v is the vehicle speed, ρ is the air density, C_d is the drag coefficient, f is the rolling resistance coefficient, m is the vehicle mass, θ is the road grade, A is the frontal area of the vehicle, and F_d is the resistance force ($F_d = \frac{1}{2}\rho A C_d v^2 + mgf\cos\theta + mg\sin\theta$).

The power distribution between the battery and engine are determined based on the power flow of the power-split architecture (please see Fig. 7.3).

$$PR = \frac{\tilde{P}_b}{P_d} \tag{7.2}$$

$$\tilde{P}_b = \eta_m^k P_b = PR.P_d \tag{7.3}$$

$$\eta_t\, P_e = (1 - PR)P_d \tag{7.4}$$

where η_t is power-split transmission efficiency, η_m is the motor-generator efficiency, P_b is the battery electrical power, and $\tilde{P}_b$, P_e are the mechanical power, delivered by the battery and engine, respectively. In Eq. 7.3, $k = -1$ during battery charging and $k = 1$ during battery discharging.

The trip data is essential for speed prediction and real-time control of the vehicle's powertrain. The route is divided into segments in a way that traffic speed and road grade to be steady in each segment. Also, speed barriers such as stop signs, bridge

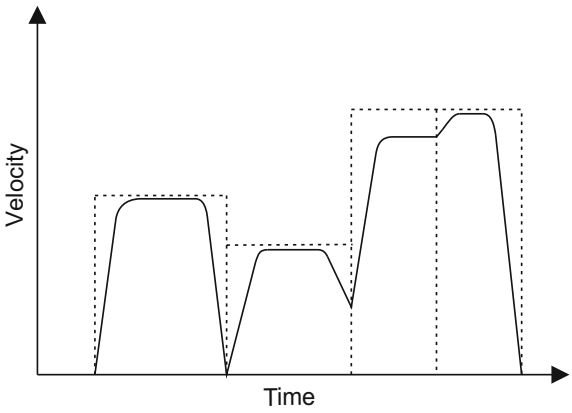


Fig. 7.4 Speed trajectories corresponding to different route segments

ramps, and traffic lights are located at the margin of the segments. Generally, segments consist of three sections: acceleration, deceleration, and cruise. However, based on the settings of each segment, the acceleration and deceleration section may be omitted. In this model, v_a, v_c, and v_f are initial speed, cruise speed, and final speed for each segment, respectively. After constructing speed trajectory, the required propulsion and regenerative braking powers are calculated based on vehicle dynamics. Finally, the fuel consumption and SOC are obtained using fuel map and battery model.

Figure 7.4 illustrates a sample segmented route and predicted speed trajectory based on the trip date. Stop point and bridge ramp are located at the end of the first and second segments, respectively. The third and fourth segments have various traffic speeds or road grades. The third segment does not contain a deceleration section.

7.2 Real-Time Optimization Procedure

Optimization problems that address future driving patterns, where a large number of parameters and constraints must be considered, are particularly challenging to solve in real-time, which is exacerbated over long trips. To reduce the complexity of these problems, this book proposes a new real-time approach, an RCO algorithm, to optimize PHEV SOC profiles.

It should be noted that the main objective function of the devised energy-optimal controller is total energy cost. To compare the performance of energy-optimal controller against other strategies, we consider a SOC constraint at terminal time. This constraint causes equal electrical energy cost for all strategies. Therefore, the optimization index can change from the total energy cost to the fuel consumption.

7.2.1 Dynamic Programming

Dynamic programming is a numerical method based on the principle of optimality that was developed by Richard Bellman in the 1950s. This global optimization approach is similar to other optimization methods, as it aims to minimize the cost function, at the same time as it fulfills performance component limitations. Because it applies a decomposition process to break down n-variable problems into n-simplified, one variable subproblems, it is commonly applicable to complex optimization problems. Each subproblem is then solved only once its saved in a table, resulting in sizeable reduction in computation time and effort [4].

The DP algorithm is utilized to obtain the optimum SOC profile that reduces total energy cost. In this problem, the SOC is the output of the system, PR is the optimization parameter, and fuel consumption is the objective function. The online optimization model is used, because it cannot be applied to DP optimization due to the complexity of the high-fidelity model built in Autonomie. The optimal dynamic optimization problem can be formulated in the discrete format as follows:

$$J = \sum_{k=1}^{N} m_f\left(PR(k)\right).\Delta t_k \tag{7.5}$$

$$SOC(k+1) = SOC(k) + \frac{-V_{oc} + \sqrt{V_{oc}^2 - 4PR(k)P_d(k)\eta^k R_b}}{2R_b Q_{max}} \tag{7.6}$$

In this problem, the following constraints should be satisfied:

$$\begin{gathered} SOC(t_0) = SOC_0 \\ SOC(t_f) = SOC_f \\ SOC_{min} \leq SOC \leq SOC_{max} \\ P_{b_{min}} \leq P_b \leq P_{b_{max}} \\ P_{e_{min}} \leq P_e \leq P_{e_{max}} \\ 0 \leq PR \leq 1 \end{gathered} \tag{7.7}$$

In the DP algorithm, the subproblems of optimization should be solved backward from the terminal condition, as shown in Fig. 7.5. For step k, the subproblem is to minimize J_k^i with $SOC^{(i)}$ as the initial point of each subproblem:

$$J_k^i = Min\left[L_k^i(PR(j)) + J_{k+1}^*\right] \tag{7.8}$$

The cost function is calculated only for the grid points of the SOC. The SOC at next time step (k+1) (filled circle), is determined through discrete state space Eq. 7.6. If this value does not match to the quantized value (empty circle), then the value of J_{k+1}^* is updated using linear interpolation.

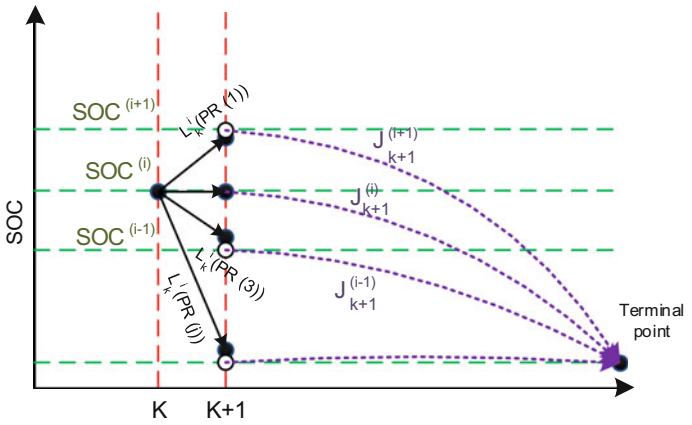


Fig. 7.5 Schematic of DP algorithm

7.2.2 Real-Time Cluster-Based Optimization

In the PHEV Trip Planning optimization problem, predicted power demand is the input and the power ratio is the optimization parameter. The energy consumption is calculated based on these two variables. It can be assumed that two trip segments with the same power demand also have the same optimum power ratio. This book proposes the RCO algorithm that clusters trip segments into groups that share similar power demands, and similar optimum power ratios. Therefore, instead of finding the optimum power ratio of each trip segment, the power ratio is only calculated for each group. In this way, the number of optimization parameters is dramatically reduced.

Generally, clustering is the method of grouping objects/data based on similarities. This produces a model data, which is then arranged into meaningful sets. Clustering algorithms may be classified in several ways: the most significant algorithms are hierarchical-based, centroid-based, and distribution-based. A robust clustering algorithm can be used to diminish complexity in optimization problems based on the application and preferred objectives.

A centroid-based or k-means algorithm is used to cluster the segments in this Ph.D research. This algorithm works based on some centroid points, which may not be members of any cluster. The data is divided into different groups based on their distances from the centroid points [5].

According to Fig. 7.6, at the first step, k cluster centers (same as the number of clusters) are determined. After defining the k centroids, data points are assigned to clusters based on their distance from the centroids. When all data have been assigned, the k centroids should be recalculated. The algorithm continues by reassigning each point to the nearest centroid. These steps are repeated until recalculation of the centroids yields no change in cluster assignment for the data.

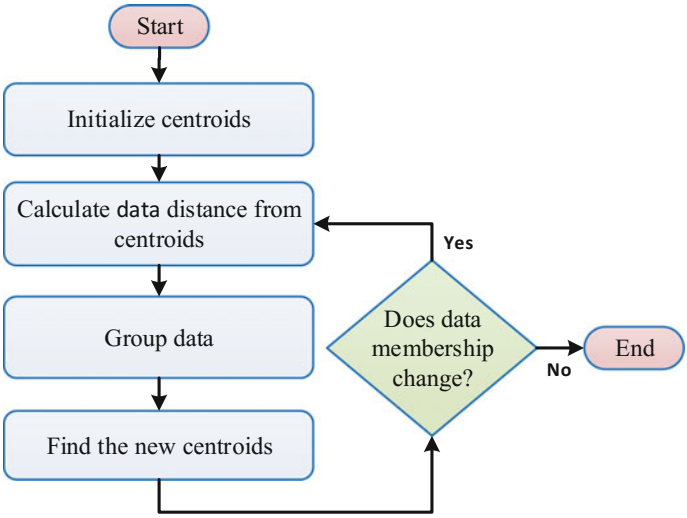


Fig. 7.6 Schematic of k-means clustering algorithm

7.3 Benchmarking via MIL and HIL

7.3.1 MIL Testing

The Trip Planning is applied to two different test scenarios. In the first, the algorithm is evaluated in different standard driving cycles; while trip data such as the average speed of each segment, segment length, road grade, and maximum speed are known beforehand. In the second test scenario, the benefit of online optimization is investigated by running the simulation when trip data changes.

7.3.1.1 Following Standard Driving Cycles

This book applies the RCO algorithm to the PHEV Trip Planning problem, taking into consideration two drive cycles. The first drive cycle (EPA-UHU cycle) begins and ends with a UDDS cycle, and has a HWFET drive cycle in the middle. The second drive cycle (3xUDDS cycle) combines three UDDS drive cycles. The travel distance of both cycles exceeds the full electric range of the vehicle. Therefore, at the terminal point, the SOC reaches its minimum value and the engine takes over propelling the vehicle.

To implement the RCO algorithm, first, the optimal power demands are calculated based on predicted speed. Then, the centroids of groups are obtained and the power demand data cluster into groups. The power ratio of each power-group is obtained to

Table 7.2 Results of RCO algorithm for the EPA-UHU drive cycle

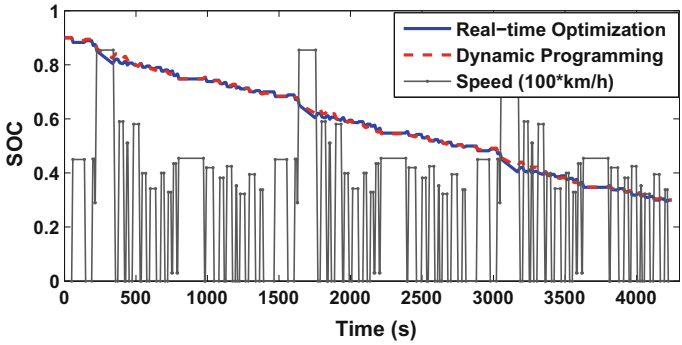
Group number	Required power (KW)	Power ratio	Speed range (km/h)
1	2.2	1	less than 40
2	3.1	1	40–60
3	6.4	0.3	60–70
4	7.1	1	70–80
5	9.1	0.1	80–110
6	21.9	0.8	Acceleration

minimize total fuel consumption. The optimization results for the EPA-UHU drive cycle is shown in Table 7.2. In the low speed and low power demand groups (groups 1 and 2), the engine did not work efficiently, therefore the vehicle switched to full electric mode. In average cruise speed (groups 3 and 5), the energy management strategy switched to Blended strategy. During acceleration with high power demand, both the engine and electric motor propel the vehicle, while the electric power is dominant ($P_b = 0.8\ P_d$).

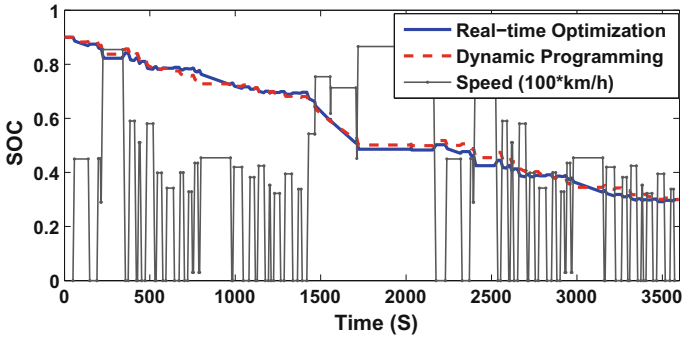
Figure 7.7 shows the optimum SOC profile based on RCO and DP approaches for the 3xUDDS and EPA-UHU drive cycles. The RCO results match the global optimum solution generated by the DP algorithm; the maximum error between the optimum SOC profile obtained by RCO and DP approaches are 1.1 and 2.4% for 3xUDDS and EPA-UHU, respectively. However, the RCO algorithm is much less computationally expensive. All simulations are run on a PC with Intel Core 2 Duo CPU (E8500, 3.17 GHz) and 4 GB RAM. The average computation time of RCO and DP algorithms are 5 s, 35 min for 3xUDDS drive cycle, and 5 s, 28 min for EPA-UHU drive cycle, respectively.

7.3.1.2 Benefit of Online Optimization

Practically, various sources such as changes in the trip plan and unpredicted traffic conditions can change trip data. This section investigates the effect of change in trip data on the performance and optimality of the online Trip Planning module through simulations of three different cases. In the first case, the 3xHWFET drive cycle is used; in the second and third cases, the EPA-HF drive cycle. In the last case, trip data change in the following manner: at first, the driver sets a trip plan in which the predicted speed is that for the 3xHWFET drive cycle. Then, after passing the first highway, the driver changes the trip plan to that of the FTP-75 drive cycle for the rest of the trip. Figure 7.8 illustrates the speed trajectories for the three different cases (Table 7.3).



(a) 3xUDDS drive cycle



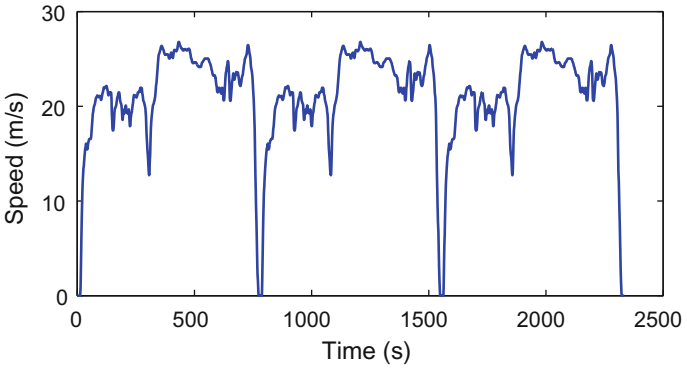
(b) EPA-UHU drive cycle

Fig. 7.7 Optimum SOC profiles produced using RCO and DP algorithms for: **a** 3xUDDS drive cycle and **b** EPA-UHU drive cycle

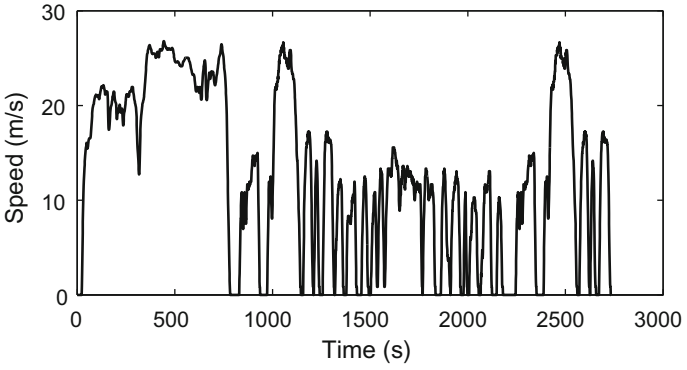
Figures 7.9, 7.10 and 7.11 show simulation results for the different EMS controller under various conditions. The performance of the controllers can be evaluated from Table 7.4. It is concluded that the Route-based EMS significantly improves fuel economy compared to the rule-based and A-ECMS controllers. However, change in trip data affects the Route-based EMS performance by 1.7% (118 MPG in case-2 and 116 MPG in case-3).

7.3.2 HIL Testing

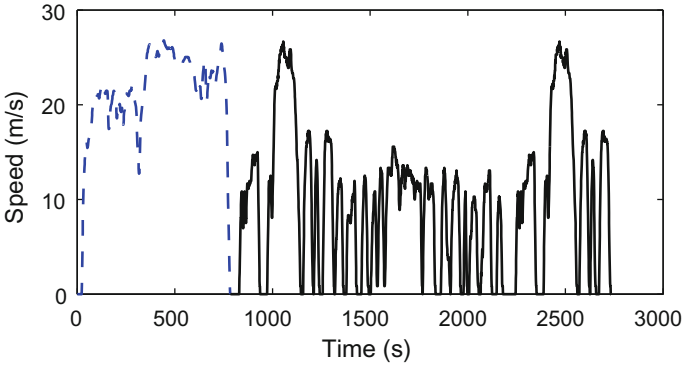
Electronic Control Units (ECUs) have been widely used in many automotive applications such as engine controller, transmission controller, cruise controller, driver assistance system, and entertainment systems. ECU development process has become a critical phase toward launching a new vehicle and 90% of automotive novelties are



(a) 3xHWFET drive cycle.



(b) EPA-HU drive cycle.



(c) EPA-HU drive cycle with change in trip plane.

Fig. 7.8 Speed trajectories applied in different case studies for online Trip Planning

Table 7.3 Characteristics of each case study for online Trip Planning

Case	Drive cycle	Change in the trip plan
1	3xHWFET	No
2	EPA-HF	No
3	EPA-HF	Yes

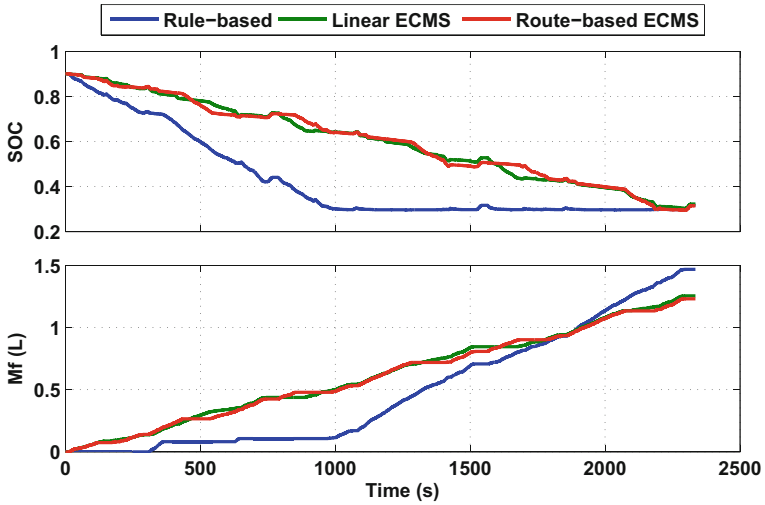


Fig. 7.9 Simulation results for following 3xHWFET drive cycle without changing in the trip plan

related to electronic systems, especially advanced ECUs [6]. Testing and validating ECUs have become crucial tasks in the automotive development process. The most effective way to validate an ECU is to connect it to a real plant. However, in many cases, hardware-in-the-loop (HIL) tests are more efficient. In fact, HIL testing systems provide a virtual vehicle model for ECU validation. The main advantages of validating ECU by HIL testing are as follows:

- The controller development can be done and verified prior to manufacturing of the prototype vehicle. Therefore, HIL systems enable simultaneous development of ECU and vehicle, which can significantly reduce the vehicle development process time;
- Validation time and cost can be reduced by replacing expensive field experiments by laboratory experiments. HIL test often requires significantly less hardware than physical prototyping. This makes the procedure faster and cheaper;
- HIL testing has lower risk, especially in the extreme or hazardous ambient conditions such as typical winter test drives, cold-start tests, or validating adaptive cruise ECU in severe situation with the risk of collision;

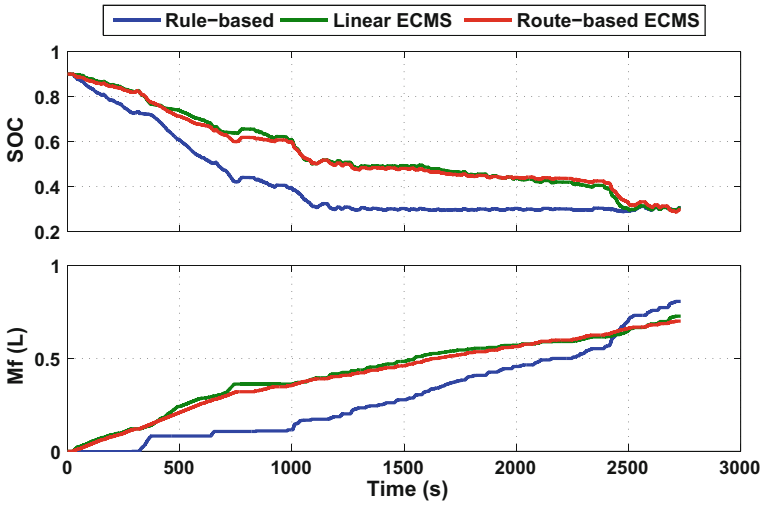


Fig. 7.10 Simulation results for following EPA-HU drive cycle without changing in the trip plan

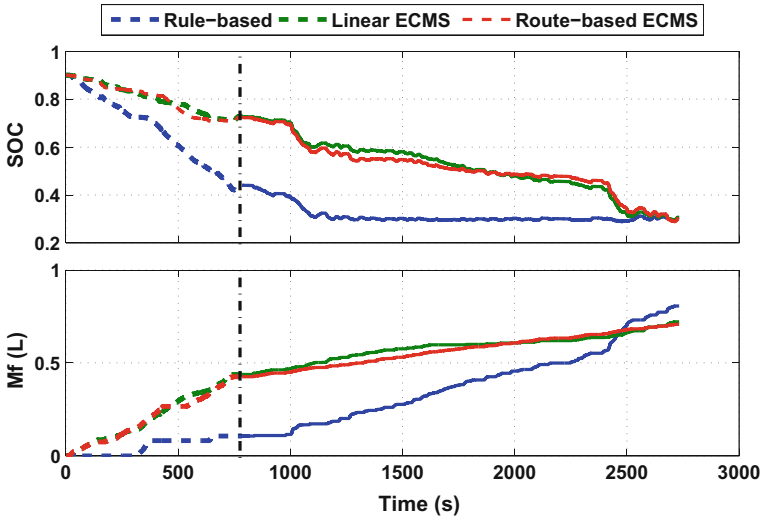


Fig. 7.11 Simulation results for following EPA-HU drive cycle when trip plan is changed after passing the first HWFET cycle

Table 7.4 Online Trip Planning results for different case studies

EMS	Trip Planning	Fuel consumption (MPG)		
		Case-1	Case-2	Case-3
Rule-based	No	97	102	102
A-ECMS	No	93	113	114
Route-based EMS	Yes	94	118	116

- HIL systems can support more comprehensive tests in a shorter time. HIL testing can significantly increase repeatability and provide simulation over much broader range of operating conditions than what is feasible via purely physical prototyping;
- HIL systems also provide the initial calibration of ECUs which is a starting point for the later development phases; and
- Damage to the vehicle can be avoided in a test scenario where failures or errors can occur. HIL simulation makes it possible to simulate destructive events without incurring a costly destruction.

This Ph.D research uses dSPACE systems for HIL testing. Digital signal processing and control engineering (dSPACE GmbH) is one of the top providers of instruments for developing ECUs, especially for automotive applications. dSPACE systems are used at many vehicle manufacturers such as Toyota, Audi, BMW, Ford, General Motors, Honda, Nissan, and some automotive suppliers [7]. dSPACE supports different phases of ECUs development process, in particular, rapid control prototyping (RCP) and ECU testing and calibration.

In general, HIL testing requires a high-performance simulator with special I/O (e.g., CAN interface), real-time model, and prototype ECU. Figure 7.12 shows the architecture of the HIL test platform. It consists of three main components: a real-time simulator (DS1006 processor board), prototype ECU (MicroAutoBox II), and interface (Computer).

The real-time simulator is a very fast processor that executes the Autonomie high-fidelity model in real-time. The high-fidelity model includes the powertrain, driver, and environment models. In each time step, it sends the powertrain variables and driver commands to the ECU through a controller area network (CAN). The ECU calculates the optimal control commands and sends them back to the real-time simulator. An interface is used to set up the test, to program both the real-time simulator and ECU, and to record the desired outputs. The specifications for the HIL components are shown in Table 7.5.

7.3.2.1 Controller Prototyping

To develop an ECU which can be implemented in the vehicle, the proper hardware should be designed for the desired application. Then, an optimized code should be generated for the target ECU platform. This process makes the ECUs development

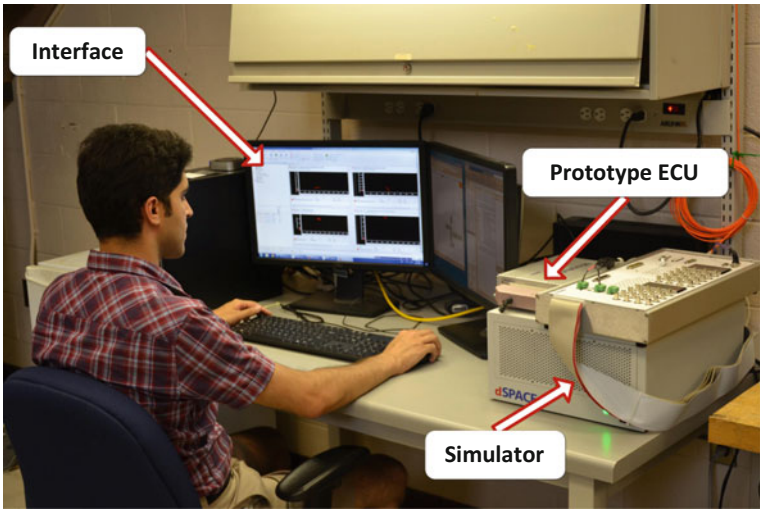


Fig. 7.12 Schematic of HIL test platform

Table 7.5 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, 4 × 128 MB global memory
	HIL I/O Board	DS-2202
ECU: MicroAutoBox 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

very challenging. To reduce the difficulties and enhance the process time, RCP systems have been developed, that support both hardware and code generation. This research used the dSPACE MicroAutoBox II as the prototype ECU for testing the real-time performance of the energy-optimal control strategy. The MicroAutoBox has two dedicated CAN controllers (four CAN channels) that enable communication with the simulator over the CAN bus.

dSPACE provides some libraries in the MATLAB software to generate a C-code for different devices (MicrroAutoBox II and DS1006 processor board). These libraries handle all features of the target device (for example, reading from analog and digital inputs and communicate with other devices through CAN communication). The host service code is also uploaded to the simulator and prototype ECU for data exchange between the real-time hardware and an interface computer.

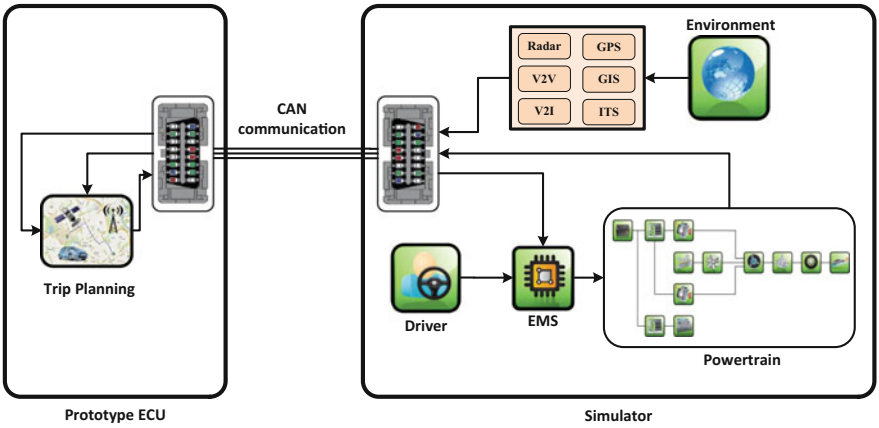


Fig. 7.13 Schematic of the Simulink models for HIL testing of the Trip Planning module

First, two separate Simulink models are generated for the PHEV high-fidelity model and the designed controller. Then, the designed control code should be prepared to be implemented in the dSPACE systems by incorporating the dSPACE Real-Time Interface (RTI) blocks. For instance, the RTI CAN Controller Setup Block is employed in both real-time model and designed controller to configure the CAN communication. The RTI CAN Receive and Sent are used for receiving and sending data.

The next step to generate a C-code for rapid control prototyping is to compile the Simulink model using the Real-Time Workshop code generator. For each hardware platform, the corresponding compiler and code generator are used (for instance, rti1401.tlc and rti1006.tlc for the MicrroAutoBox (DS1041) and Simulator (DS1006), respectively). The Real-Time Workshop creates the following files after code generation process: system description file (*.sdf) for uploading the executable file from interface computer using ControlDesk software, executable file (*.ppc) for the processor in the hardware platform, trace file (*.trc) for navigating through the model, and map file (*.map) for the memory address. The dSPACE ControlDesk software is an interface used to connect with RCP hardware platform, upload the executable file, manage the HIL test, and record the desired signals.

Figure 7.13 shows the software architectures for both the prototype ECU and simulator. The prototype ECU runs the Trip Planning and the simulator executes both the environment and powertrain models. These modules are communicated over the CAN bus. Table 7.6 represents the input and output signals and their characteristics in CAN communication.

Table 7.6 Specification of the input and output signals of the Trip Planning module

CAN signal	Variable name	CAN ID	Bit length
ECU inputs	Position	100	32
	Speed	101	32
	SOC	102	16
ECU outputs	Slope of Ref. SOC	120	16
	Initial position of Ref. SOC	121	16
	Initial SOC of Ref. SOC	122	16

7.3.2.2 HIL Testing Results

To validate the real-time capability of the ECU, the turnaround time of the controller should be less than the desired time step. Turnaround time is the amount of time taken to execute the controller code and to provide the required ECU output. To implement the HIL testing, first, the time step of the system should be determined. Generally, the response time of the system or the update-rate of the impute signals are considered as a time step. One of the main input signals to the Trip Planning module is traffic data which is approximately updated in one minute. Therefore, the time step of this module is considered to be one minute.

The HIL results of the online Trip Planning module are shown in Fig. 7.14. The results show that the turnaround time of Trip Planning in the MicroAutoBox hardware platform is less than 5 s, which is significantly less than the desired time step (1 min). Therefore, the designed Trip Planning can be implemented in real-time.

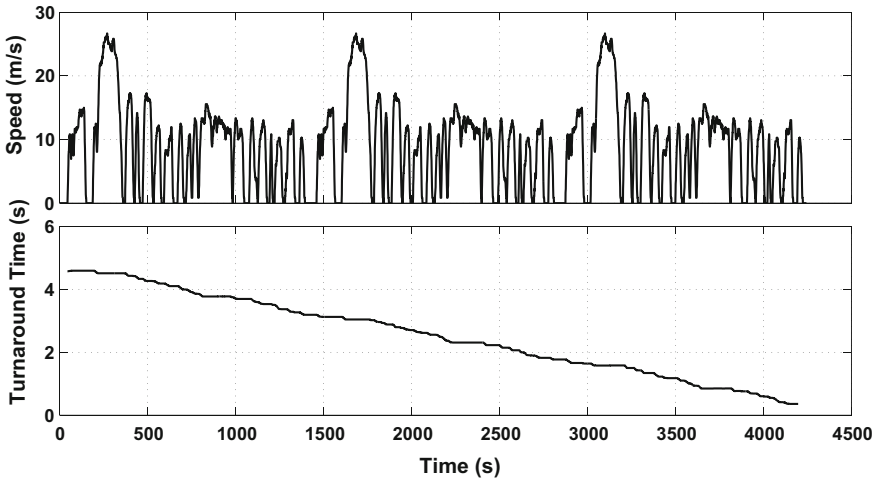


Fig. 7.14 Trip Planning HIL test results under 3xUDDS drive cycle

7.4 Summary

This chapter has presented the Trip Planning for PHEV platforms that can optimize the total expenditure of electricity and fossil fuel. The designed algorithm utilizes look-ahead trip and driving information to predict optimum SOC profiles.

In the first step, PHEV online optimization model is developed and validated against the high-fidelity PHEV model. The speed trajectory is predicted using the math-based trip model. The optimization problem is then solved in real-time using the RCO algorithm. Results show that at low speeds, the low-efficiency engine is turned off and the vehicle is operated in full electric mode. Conversely, the engine is dominant during acceleration, when power demand is high.

The RCO results are evaluated against those generated using a global optimization approach, DP. The real-time results are very promising as they show good agreement with the DP data and are achieved at much less computational expense. The results also show that online optimization can compensate changes in trip data.

The HIL results demonstrate that the computational time of the RCO algorithm is significantly less than 1 minute that validates the real-time Trip Planning module.

References

1. Vajedi, M., Chehrehsaz, M., Azad, N.L.: Intelligent power management of plug-in hybrid electric vehicles, part I : real-time optimum SOC trajectory builder. Int. J. Electric Hybrid Veh. **6**, 46–67 (2014)
2. Vajedi, M., Taghavipour, A., Azad, N.L.: Traction-motor power ratio and speed trajectory optimization for power split PHEVs using route information. In: ASME 2012 International Mechanical Engineering Congress and Exposition (2012)
3. Mozaffari, A., Vajedi, M., Azad, N.L.: Real-time immune-inspired optimum state-of-charge trajectory estimation using upcoming route information preview and neural network for plug-in hybrid electric vehicles fuel economy. Front. Mech. Eng. **10**, 154–167 (2015)
4. Naidu, D: Optimal Control Systems. CRC press (2002)
5. Rokach, L., Maimon, O.: Data Mining and Knowledge Discovery Handbook. Springer (2005)
6. Hammerschmidt, C. : Innovation in the car: 90% comes from electronics and software. In: EETimes Europe Automotive (2014)
7. dSPACE GmbH : dSPACE catalog 2004: solutions for control. In: EETimes Europe Automotive (2004)

Chapter 8
Route-Based Supervisory Controls

This chapter covers steps to design a real-time Route-based EMS for a power-split PHEV. The EMS controller adjusts power distribution between two energy sources, the engine and the battery. A schematic of the Route-based EMS is shown in Fig. 8.1. Trip Planning module employs the simple powertrain model and predicted speed trajectory to find optimum power distribution, while Route-based EMS calculates the optimum power distribution, based on actual momentary power demand. Therefore, this EMS can handle any sudden speed or propulsion power variations during a trip. Finally, the low-level controllers adjust the engine, the electric motors, and the battery operations to provide the demanded power based on the optimum power distribution.

The Route-based EMS is compared against popular EMS strategies, based on different levels of trip information. Also, the real-time implementation of the devised Route-based EMS is evaluated using HIL testing. Portions of this chapter have been published in [1–5].

8.1 Optimum Energy Management Development

The control-oriented model is developed to obtain the power of powertrain components based on the battery power. This model is same as the online optimization model, which is presented in Sect. 7.1. The only difference is that the control-oriented model calculates power demand based on the command of driver or cruise controller system, while the online optimization model predicts future speed using the math-based trip model. In the control-oriented model, the power distribution between the engine and the battery are calculated based on the power ratio (*PR*):

A. Taghavipour et al., *Intelligent Control of Connected Plug-in Hybrid Electric Vehicles*,
Advances in Industrial Control, https://doi.org/10.1007/978-3-030-00314-2_8

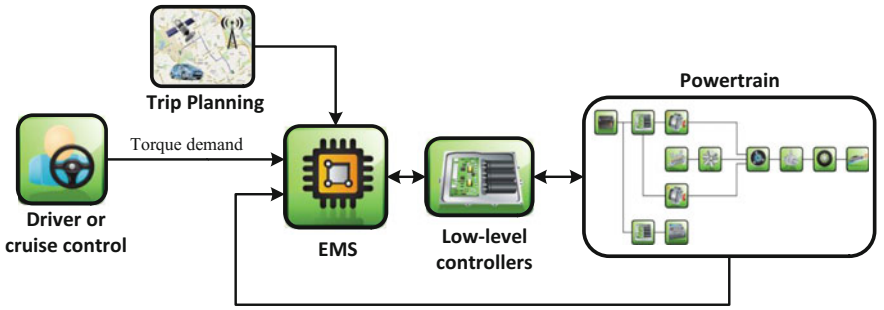


Fig. 8.1 Schematic of the real-time Route-based EMS strategy

$$P_d = \left(m\dot{v} + \tfrac{1}{2}\rho A C_d v^2 + mgf\cos\theta + mg\sin\theta\right) v \tag{8.1}$$

$$P_e = \frac{1 - PR}{\eta_t} P_d \tag{8.2}$$

$$P_b = \eta_m^{-k} PR.P_d \tag{8.3}$$

The objective of the controller is to minimize the total energy cost of the vehicle, including the fuel and grid electrical energy, which is formulated in Eq. 8.4:

$$Cost = k_f \dot{m}_f + k_e \eta_{ch} Q_{max} \dot{SOC} \tag{8.4}$$

where m_f is fuel consumption, Q_{max} is the maximum battery capacity, η_{ch} is the charger efficiency, and k_f, k_e are the unit price of gas and grid electrical energy, respectively.

To establish an optimum EMS of PHEV, the power distribution between the engine and the battery are calculated to minimize the total energy cost considering the constraints on the system. The optimal control problem can be derived by considering SOC as a state of the system, power ratio (PR) as an input, and total energy cost as a cost function:

$$J = \int_0^{t_f} Cost\ dt = \int_0^{t_f} \left(k_f \dot{m}_f - k_e \eta_{ch} Q_{max} \dot{SOC}\right) dt \tag{8.5}$$

$$Subjected\ to: \quad \dot{SOC} = -\frac{\eta_b}{Q_{max}} P_b$$

$$SOC(0) = SOC_0$$

$$SOC(t_f) = SOC_f$$

The constraints on the system are:

$$SOC_{min} \leq SOC \leq SOC_{max}$$
$$P_{b,min} \leq P_b \leq P_{b,max}$$
$$P_{e,min} \leq P_e \leq P_{e,max}$$

Based on the control-oriented model, the power capacity of the powertrain components can be calculated as functions of the power ratio (PR). Therefore, the constraints can be combined to reach equivalent constraints in terms of battery power and SOC:

$$SOC_{min} \leq SOC \leq SOC_{max} \tag{8.6}$$
$$PR_{min} \leq PR \leq PR_{max} \tag{8.7}$$
$$PR_{max} = \min\left(1, (1 - \eta_t \frac{P_{e,min}}{P_d}), (\eta_m \frac{P_{b,max}}{P_d})\right) \tag{8.8}$$
$$PR_{min} = \max\left(0, (1 - \eta_t \frac{P_{e,max}}{P_d}), (\eta_m \frac{P_{b,min}}{P_d})\right) \tag{8.9}$$

To solve this problem, the PMP technique is applied to obtain the global optimum solution. A real-time algorithm that includes future driving condition is then applied based on ECMS (Route-based EMS).

8.1.1 Pontryagin's Minimum Principle

PMP is an optimal control approach used to find the global optimal solution for control problems [6]. Assuming the objective of the optimal control problem is to minimize the cost function (J) as a function of time (t), and the state of the system is x:

$$J = \int_{t_0}^{t_f} L(x(t), t)dt \tag{8.10}$$
$$Subjected\ to:$$
$$\dot{x} = f\ (x(t), u(t), t) \tag{8.11}$$

with the following constraints:

$$\Psi\ (x(t), t) \leq 0 \tag{8.12}$$

To solve this problem, the Hamiltonian function must first be derived:

$$H = L(x(t), t) + \lambda f\ (x(t), t) + \nu \Psi\ (x(t), t) \tag{8.13}$$
$$\nu = \nu_0 \qquad if\ \psi\ (x(t), t) = 0$$
$$\nu = 0 \qquad if\ \psi\ (x(t), t) < 0$$

Then, the state and costate equations should be solved simultaneously:

$$\dot{x} = \frac{\partial H}{\partial \lambda} = f\ (x(t), u(t), t) \tag{8.14}$$

$$\dot{\lambda} = -\frac{\partial H}{\partial x} \tag{8.15}$$

The optimum input, $u^\star$, is obtained by minimizing the Hamiltonian function:

$$u^\star = argmin\ \{H\,(x(t), u(t), t)\} \tag{8.16}$$

In the optimum EMS of the baseline PHEV, the cost function is the energy cost. Therefore, according to Eq. 8.13, the Hamiltonian equation can be derived as:

$$H = Cost(P_b) + \lambda(-\frac{\eta_b}{Q_{max}}P_b) + \nu_1(soc_{min} - soc) + \nu_2(soc - soc_{max}) \tag{8.17}$$

$$\begin{aligned} \nu_1 &= \nu & soc_{min} &= soc \\ \nu_1 &= 0 & soc_{min} &< soc \\ \nu_2 &= \nu & soc &= soc_{max} \\ \nu_2 &= 0 & soc &< soc_{max} \end{aligned}$$

The equations for the state and costate that should be solved simultaneously are:

$$\begin{aligned} \dot{SOC} &= -\frac{\eta_b}{Q_{max}}P_b \\ SOC(0) &= SOC_0 \\ SOC(t_f) &= SOC_f \\ \dot{\lambda} &= -\frac{\partial H}{\partial SOC} \end{aligned} \tag{8.18}$$

Also, based on the PMP algorithm, the optimum control law will be:

$$u^\star = P_b^\star = argmin\ \{H(P_b, SOC)\} \tag{8.19}$$

This problem is a two-point boundary value (TPBV) problem, in which the initial and final values of the SOC are known and the initial value of λ is unknown. The simple shooting method (SSM) is a common approach to solving TPBV problems. This method converts the TPBV problem into an initial value problem, and adjusts the initial values to satisfy the end constraints. In this problem, an initial value for the costate $\lambda(t_0)$ is assumed first. Then, the state and costate equations are solved simultaneously. At the terminal time (end of trip), the difference between the final SOC ($SOC(t_f)$), and its desired value (SOC_f), is used to correct the initial guessed value of $\lambda(t_0)$. This procedure is repeated until the final SOC value approaches

the desired value. Since the state and costate equations depend on power demand, upcoming driving conditions are required to solve the TPBV problem and calculate the initial λ.

Since the SSM is very sensitive to the initial guessed value of the costate, the more accurate modified simple shooting method (MSSM), is employed to solve the TPBV problem. This method is similar to the SSM, but defines the passes through a trajectory in order to guide the final state to its desired value [7]. In this problem, the pass- through trajectory is represented by a line between the initial and final states:

$$\Phi = SOC_0 + \frac{SOC_f}{t_f} t_k \tag{8.20}$$

In each step, the SSM is solved for the reduced trip in time interval $[0, t_k]$, with the desired final state of $SOC_f^k = \Phi(t_k)$. The solution of this step, λ_0^k, is considered as an initial guess of the next step, time interval $[0, t_{k+1}]$. This process continues until the problem is solved for the entire time interval $[0, t_f]$, [8].

Lewis [9] has proved the stability of the PMP technique. This technique solves the optimal control problem for an infinite horizon. In this problem, there are constraints on both states and input of the system. Therefore, if the power demand is feasible and can satisfy the powertrain constraint, the solution can stabilize the system.

8.1.2 Route-Based EMS

The ECMS was initially developed to address the real-time EMS of HEVs. Since HEVs have only one external power source, vehicle performance can only be evaluated if the electrical energy generated by the battery is converted to an equivalent fuel consumption value. The total equivalent fuel consumption represents the consumption of both energy resources, electrical energy and fuel, which should be minimized in the optimal EMS strategy:

$$\begin{aligned} J &= \int_{t_0}^{t_f} \dot{m}_{eq}\, dt \\ \dot{m}_{eq} &= \dot{m}_f + S\frac{\eta_{ch}\eta_b}{H_{LHV}} P_b \end{aligned} \tag{8.21}$$

where η_b is the battery efficiency, H_{LHV} is a low heat value of the fuel, and S is the equivalency factor.

Conversely, PHEVs have two external energy sources and can store electric energy directly from the grid. Furthermore, the electrical energy stored in the battery is not provided by the engine and is independent from the fuel. Therefore, the total energy cost during the trip is considered as a cost function of the problem.

$$J = \int_{t_0}^{t_f} \boldsymbol{F}(Pb, S)\, dt \tag{8.22}$$

$$\boldsymbol{F}(Pb, S) = k_f \dot{m}_f + \boldsymbol{S} k_e \eta_{ch} \eta_b P_b \tag{8.23}$$

In comparing the PMP and ECMS approaches, the main difference is that the PMP solves the TPBV problem to calculate $\lambda(t)$, while the ECMS assigns an equivalency factor as a design parameter.

The equivalency factor has a significant effect on the power distribution between the two energy sources. If the battery energy capacity is unlimited, the equivalency factor is equal to one, $S = 1$, and the total energy cost can be used to evaluate vehicle performance. However, there is a constraint on the battery capacity; therefore, the equivalency factor is defined to regulate the supply and demand of electrical energy. For instance, if it is assumed that the electrical energy cost is less than the fuel cost, and the trip distance is more than the full electric range of the vehicle, the demand for electrical energy will be more than the battery capacity. Increasing the equivalency factor raises the cost of electrical energy, which allows us to find a more optimal point for consuming the battery electrical energy.

In general, reducing the equivalency factor decreases electrical energy cost. According to the control law, this reduction in the cost of electrical energy leads to increased battery power and decreased engine power. Therefore, more electrical energy is consumed if the equivalency factor is decreased, and vice versa: increasing the equivalency factor increases the fuel consumption.

The optimal equivalency factor depends on the driving cycle and future power demand. To overcome this problem and ensure the ECMS approach is independent of the driving cycle, an A-ECMS is employed. In this algorithm, the equivalency factor $S(t)$ is obtained based on the reference SOC. In the literature, it has been shown that the linear profile of SOC with respect to driving distance is a near-optimum solution for the PHEV energy management problem, and it is considered as a reference SOC in the A-ECMS method.

In the Route-based EMS, the Trip Planning scheme first uses future driving conditions to find the optimum SOC profile and then employs it as a reference SOC. A PID controller is employed to generate an equivalency factor based on the reference SOC, as shown in Fig. 8.2.

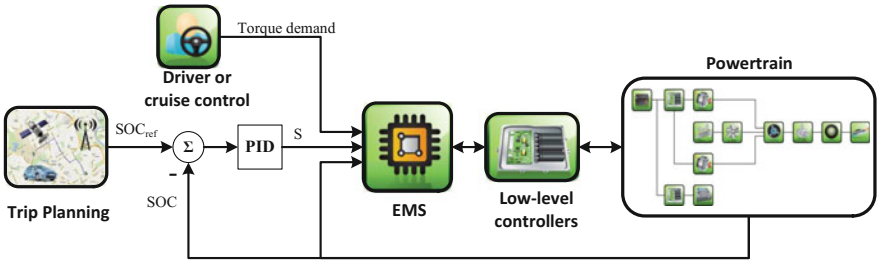


Fig. 8.2 Schematic of the Route-based EMS

8.1.3 Level of Trip Information

As mentioned, preview knowledge of a trip can significantly improve EMS performance. In this way, advancements in vehicle intelligence and communications technologies such as GPS, ITS, GIS, radar, and other onboard sensors to provide look-ahead trip data can be taken advantage of. These data can be utilized to predict the future driving conditions and increase the performance of EMS strategies. In terms of the level of access to the trip preview information, the EMS strategies can be categorized into three groups: (I) no access, (II) traveling distance, and (III) speed trajectory.

Reactive EMS strategies such as rule-based, CDCS, and Manual CDCS requires no preview of trip information. The Autonomie default control strategy is rule-based and similar to CDCS, except that the engine operates in CD mode when power demand is high.

In A-ECMS technique, the traveling distance is known in advance, and a linear battery depletion profile is used as a reference SOC. In the Route-based EMS, the Trip Planning module use sensors to acquire traffic conditions and thus find an optimum reference SOC profile.

The global optimum solution is obtained by employing the PMP technique to evaluate other strategies. The exact future driving conditions and speed trajectory are known in advance in this technique (due to complete preview information).

This chapter evaluated the real-time Route-based EMS against three existing EMS strategies, taking into consideration different levels of trip information.

8.2 MIL Testing

This section investigates the performance of the popular EMS strategies: rule-based, CDCS, Manual CDCS, A-ECMS, and Route-based EMS, based on different levels of trip information. In particular, the performance of the Route-based EMS is compared against MPC controller, which is developed by Taghavipour [10].

8.2.1 Following Standard Driving Cycles

The simulation results for EMS strategies without any preview trip information according to EPA-UHU drive cycle are shown in Figs. 8.3, 8.4, and 8.5, respectively. The control system accurately follows the reference speed trajectory with a margin of error of less than 1 km/h.

The engine operates more efficiently at higher speeds than lower speeds. As a result, the better engine operation while driving on highways can improve fuel economy. While driving in urban areas, using electrical energy is preferable. So, in

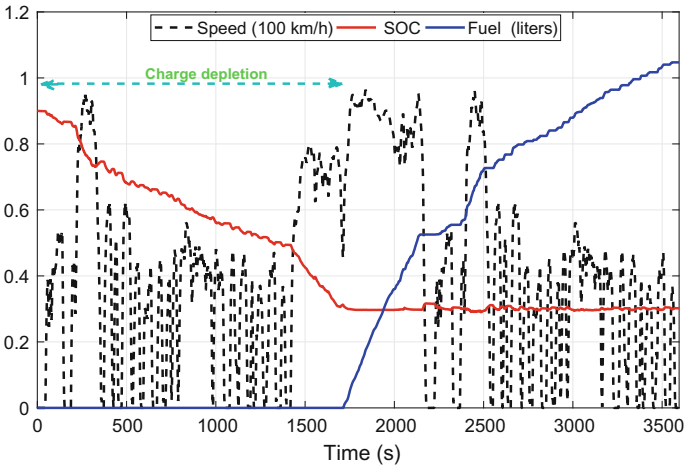


Fig. 8.3 CDCS strategy performance over 3xUDDS driving schedule

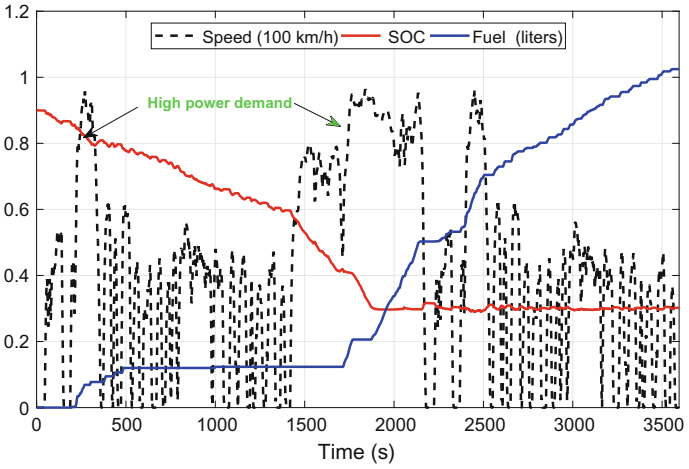


Fig. 8.4 Autonomie's default rule-based strategy performance over 3xUDDS driving schedule

Manual CDCS, the driver can manually switch between EV mode (CD) and HEV mode (CS) depending on traffic conditions. Doing so can improve the fuel economy further. The Manual CDCS strategy utilizes more engine power in the highway rather than rule-based strategy, and extends CD operating mode until $t = 3100$ s. This strategy enhances the fuel economy by 5.6% compared to the default rule-based controller in Autonomie software.

In CDCS strategy, the battery energy propels the vehicle for approximately the first half of the trip ($t = 1715$ s), then the operating mode switches to CS and the engine takes over (Fig. 8.3). The rule-based controller of Autonomie has a similar

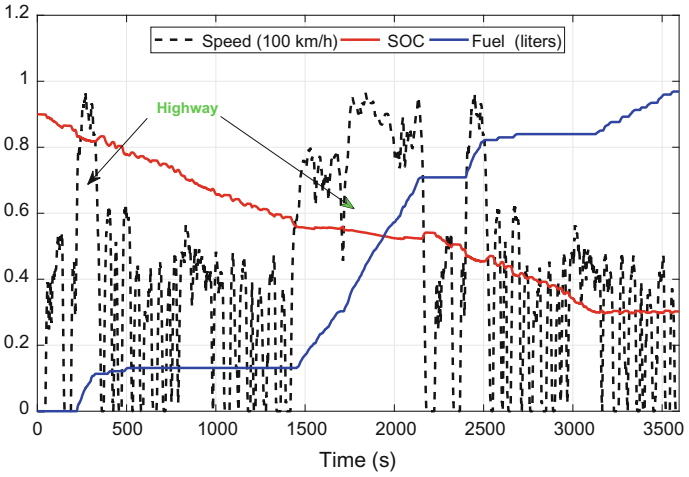


Fig. 8.5 Manual CDCS strategy performance over 3xUDDS driving schedule

approach, but it starts the engine even when SOC is more than the predefined value. As shown in Fig. 8.4, for acceleration at $t = 220$ s and $t = 1720$ s, the engine assists the electric drive to propel the vehicle. This leads to a longer charge depletion period and a 2.1% improvement in fuel economy (MPG) over CDCS.

The global optimum solution is obtained using the PMP technique. The resulting TPBV problem is very sensitive to initial value of λ. Figures 8.6, 8.7, 8.8 and 8.9 show the simulation results for different initial λ values. In low λ_0, the battery power mainly propels the vehicle until the SOC reaches the minimum value, while in high λ_0, the engine is the main power source. The modified simple shooting method is employed to solve the TPBV problem, and the optimum λ_0 satisfies the constraint $SOC(t_f) = SOC_f$.

Based on the level of trip information, A-ECMS or Route-based EMS strategies can be used to find optimal power distribution. A linear reference SOC can be developed based on trip distance or an optimum SOC profile can be generated by Trip Planning described in Chap. 7. The optimum fuel consumption and SOC profiles generated by the A-ECMS and Route-based EMS for various drive cycles are shown in Figs. 8.10, 8.11, 8.12, 8.13, 8.14, and 8.15. The results show that both Route-based EMS and A-ECMS strategies extend the charge depleting mode until the end of trip. Therefore, both controllers can optimally use the battery and engine power for longer range compare to rule-based controller. The results also demonstrate the Route-based EMS can significantly improve the energy cost.

Table 8.1 shows the results of the EMS strategies. The best results are achieved by the real-time Route-based EMS, which closely approximated the results of the global optimum solution based on PMP technique. When no preview information is available, the Manual strategy shows the best results.

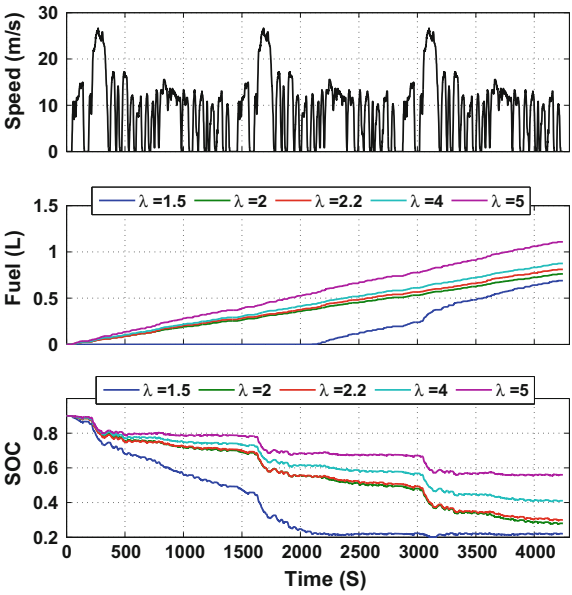


Fig. 8.6 Simulation results for 3xUDDS drive cycle using different initial λ values

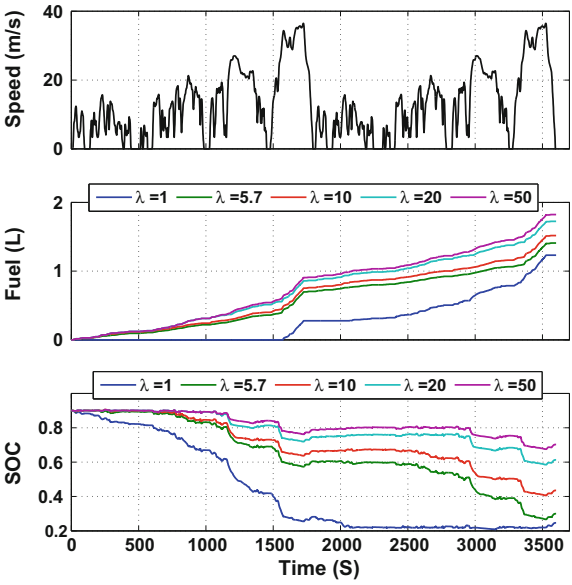


Fig. 8.7 Simulation results for 2xWLTP drive cycle using different initial λ values

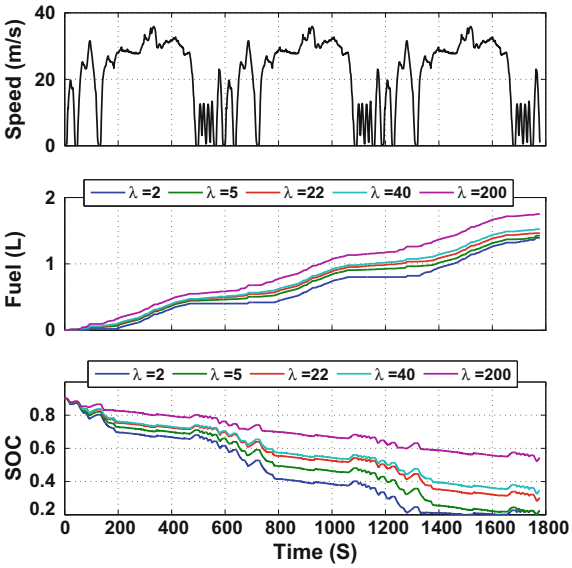


Fig. 8.8 Simulation results for 3xSFTP-US06 drive cycle using different initial λ values

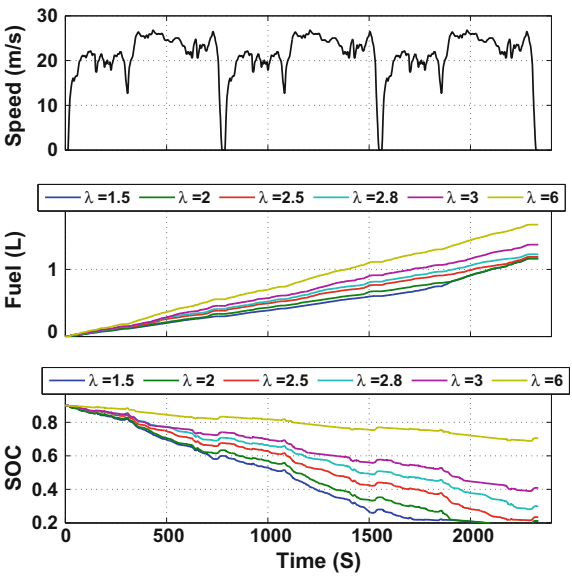


Fig. 8.9 Simulation results for 3xHWFET drive cycle using different initial λ values

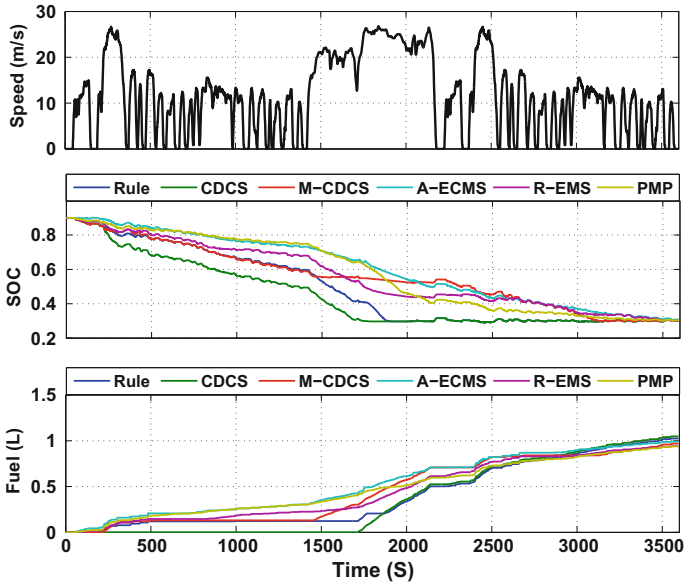


Fig. 8.10 Simulation results for each EMS strategy when tracking EPA-UHU drive cycle

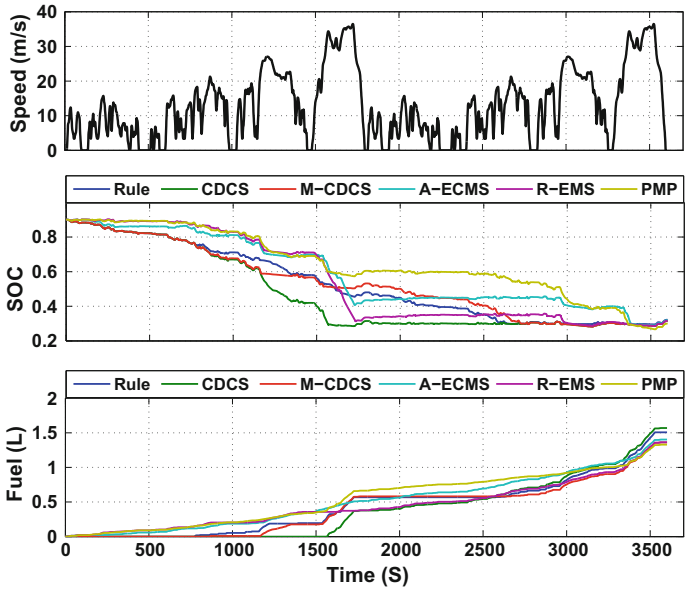


Fig. 8.11 Simulation results for each EMS strategy when tracking 2xWLTP drive cycle

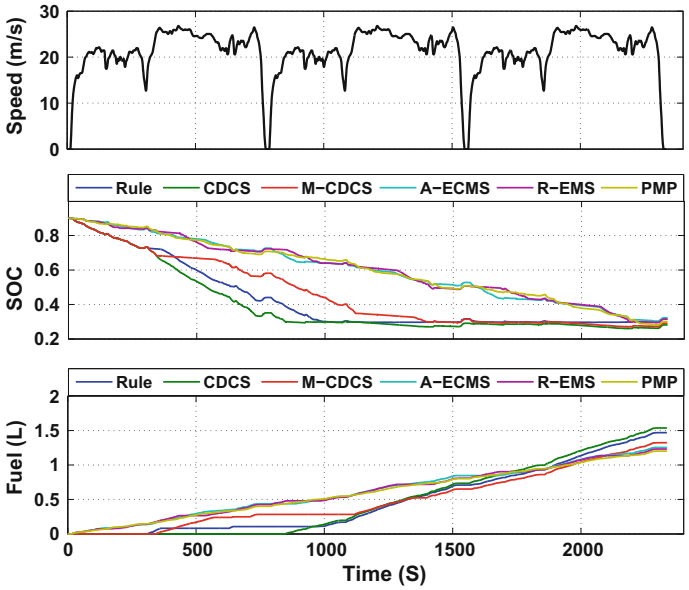


Fig. 8.12 Simulation results for each EMS strategy when tracking 3xHWFET drive cycle

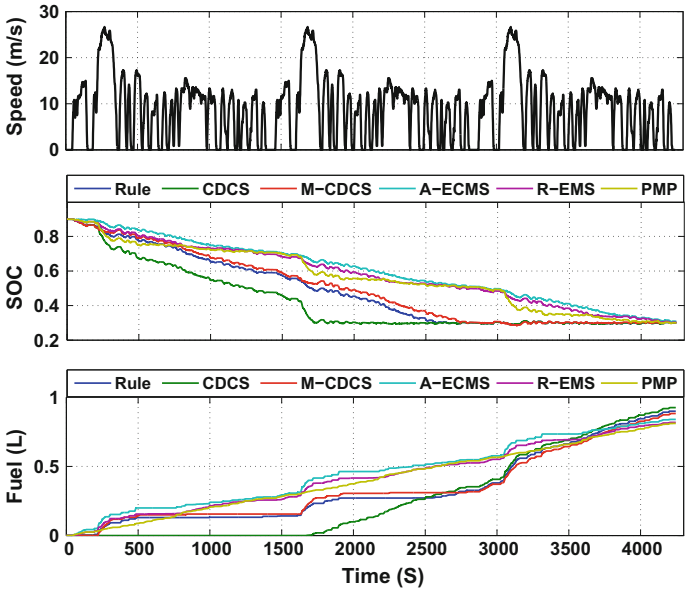


Fig. 8.13 Simulation results for each EMS strategy when tracking 3xUDDS drive cycle

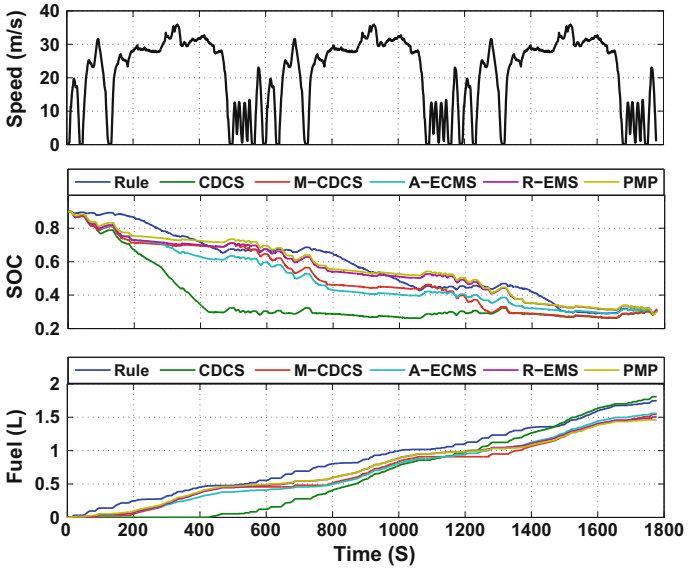


Fig. 8.14 Simulation results for each EMS strategy when tracking 3xSFTP-US06 drive cycle

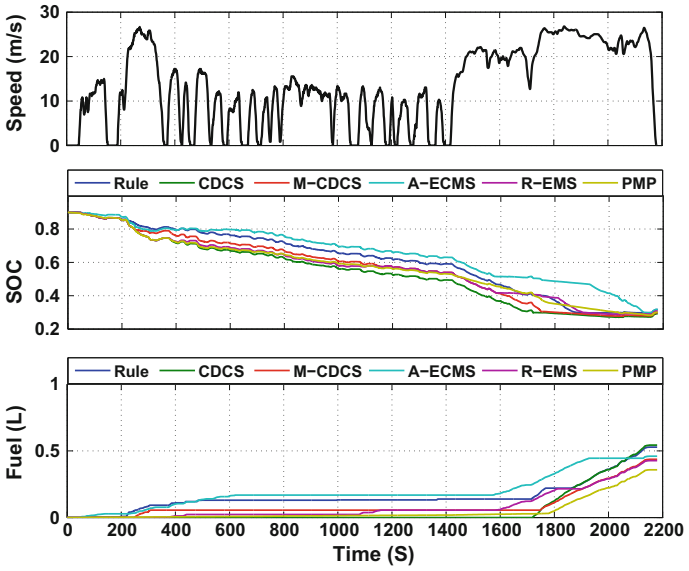


Fig. 8.15 Simulation results for each EMS strategy when tracking EPA-UH drive cycle

Table 8.1 Fuel consumption results for each EMS strategy

Drive cycle	Fuel consumption (MPG)					
	Rule-based	CDCS	Manual	A-ECMS	Route-based	PMP
3xUDDS	97.7	94.9	99.6	104.7	107.7	108.4
3xHWFET	79.3	75.8	88.0	92.8	94.6	96.9
3xSFTP	52.1	50.3	59.0	58.4	60.5	62.2
2xWLTP	72.7	69.7	80.4	78.0	80.3	82.3
EPA-UH	129.0	125.5	156.0	148.0	159.6	190.2
EPA-UHU	95.1	93.1	100.5	98.3	103.4	103.8

8.2.2 Comparison with MPC Controller

This study also compares two optimal route-based control approaches—the MPC controller and the devised Route-based EMS—for different levels of trip information. The MPC energy management strategy was developed in our research group by Taghavipour [11]. Both controllers are implemented in the high-fidelity model.

8.2.2.1 MPC Energy Management Strategy

The MPC technique has been applied to many control applications recently, due to its ability to handle constraints on states and inputs of the system. It solves the optimal control problem in a finite horizon time, which makes the controller capable of real-time implementation.

The MPC controller uses the control-oriented model in order to predict the future. In each prediction horizon, a cost function is minimized that results in maximum fuel economy and tracking a reference SOC trajectory while following a drive cycle. The cost function is:

$$J(k) = \sum_{i=1}^{N_p} (w_1 (SOC_{ref}(k+i) - SOC(k+i))^2 + w_2 (\dot{m}_f(k+i))^2). \tag{8.24}$$

In Eq. 8.24, w_1 are w_2 weighting parameters that are chosen according to the predicted maximum value of the weighted variables. The performance of the control system can deteriorate significantly when the control signals from the original design meet with the constraints. There are some constraints on this problem that are defined as follows:

$$\begin{aligned} T_{min-i} < T_i < T_{max-i} \quad & i \in \{e, m, g\} \\ \omega_{min-i} < \omega_i < \omega_{max-i} \quad & i \in \{e, m, g\} \\ SOC_{min} < SOC < SOC_{max} & \end{aligned} \tag{8.25}$$

Stabilizing the MPC controller requires consideration of extra constraints or auxiliary objective functions, which affects the global optimality of the original problem [12–14]. It also increases the computational time and makes real-time implementation more challenging. In addition, adding constraints might make the optimal solution infeasible. Zheng et al. [15] show that closed-loop MPC controllers are asymptotically stable if the optimization problem is feasible. Therefore, MPC stability can be achieved by ensuring that the solution is always feasible and the constraints are satisfied.

8.2.2.2 Simulation Results

To evaluate the designed EMS strategies performance (MPC controller and Route-based EMS), they are implemented to the high-fidelity model in Autonomie. Two different driving schedules for urban driving and combined highway and urban driving are used for the simulation. The first one is a combination of three UDDS drive cycles (3xUDDS drive cycle) and the latter is two UDDS and a HWFET drive cycles (EPA-UHU drive cycle).

The trip information helps in improving the EMS strategy performance, while the battery depletion profile affects the PHEV fuel economy. If the traveling distance is available beforehand, the linear depletion profile is used as the reference SOC. In the case of modern vehicle with onboard sensors, the optimum depletion profile is generated by Trip Planning in real-time and is applied to the EMS strategy. The simulation results of A-ECMS and MPC controller for linear reference SOC are shown in Fig. 8.16. It is shown that both EMS strategies satisfy the constraint on SOC at the end of the trip.

MPC uses more battery power while the vehicle is accelerating (at $t = 210\ s$ and $t = 2400\ s$). As such, electrical power provides smoother engine operation, since the engine operates inefficiently in transients. However, A-ECMS uses more engine power for acceleration in order to follow the SOC reference trajectory.

The simulation results with optimized reference SOC are shown in Fig. 8.17. In highway, Route-based EMS first utilizes more engine power for acceleration part and then employs battery power to track the reference SOC. However, MPC propels the vehicle by battery power in acceleration mode then uses engine power to restore battery energy and track the reference SOC. For instance, in the second segment, in the time period of 180–360s, the fuel consumption and ΔSOC are 0.122 L, 0.032% for MPC, and 0.087 L, 0.049% for Route-based EMS, respectively. Therefore, MPC increases SOC (decrease ΔSOC of the segment) by utilizing more engine power at the end of the segment (Fig. 8.18).

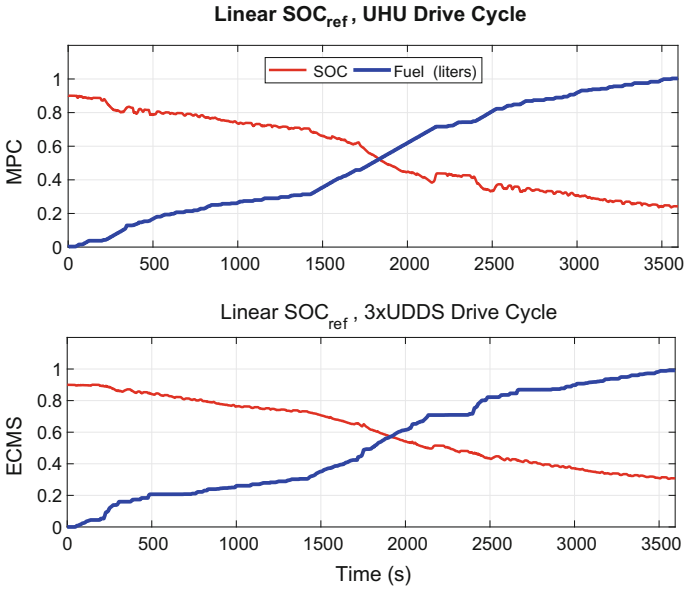


Fig. 8.16 Results of EMS strategies with linear reference SOC

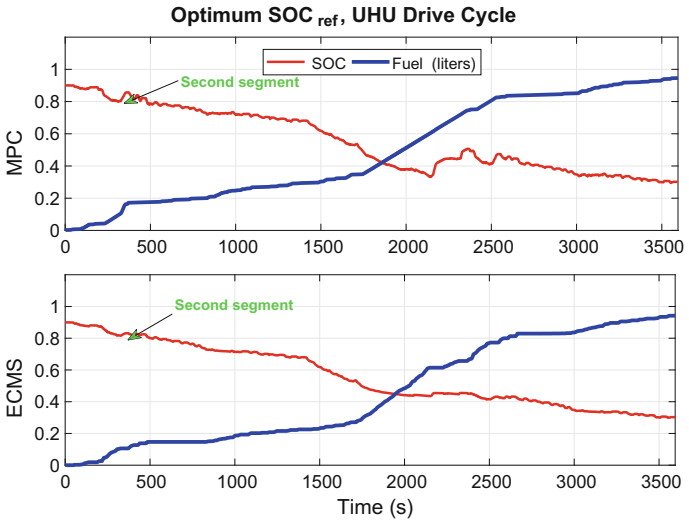


Fig. 8.17 Results of EMS strategies with optimized reference SOC

To evaluate these EMS strategies in a different driving schedule, 3xUDDS drive cycle is applied to the model. Figure 8.19 shows the simulation results. By comparing the results of Route-based EMS and MPC strategies for different driving schedules, it is found that the fuel consumption of the two EMS strategy are close to each other.

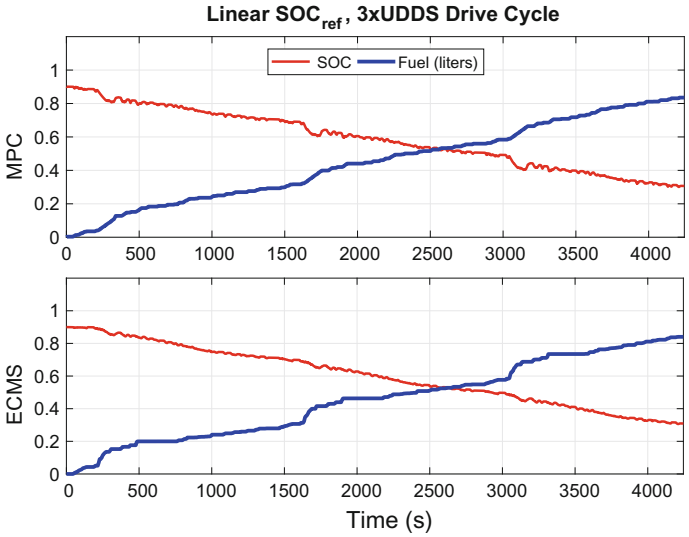


Fig. 8.18 Results of EMS strategies with linear reference SOC over the 3xUDDS driving schedule

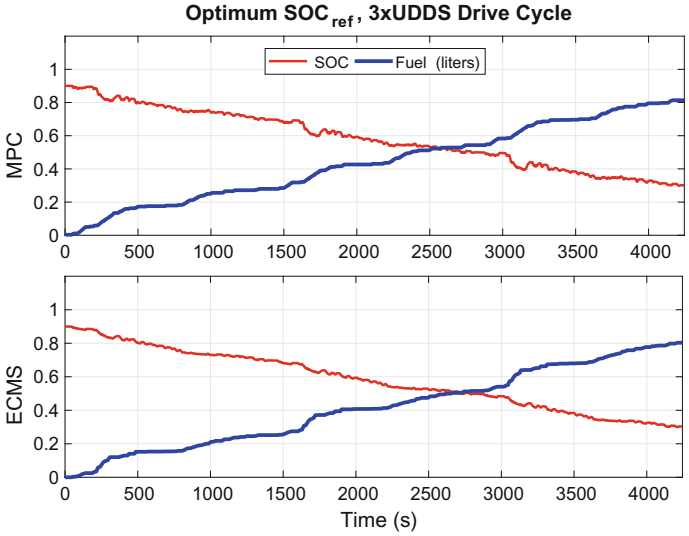


Fig. 8.19 Results of EMS strategies with optimized reference SOC over the 3xUDDS driving schedule

The fuel consumption for different EMS strategies with different levels of trip information in EPA-UHU and 3xUDDS drive cycles are shown in Table 8.2. When the future trip information is not available, the Manual CDCS has the best performance. If the traveling distance is known in advance, Manual CDCS or A-ECMS results are

Table 8.2 Fuel economy for different levels of trip information

Strategy	Level of trip information	Fuel consumption (MPG)	
		EPA-UHU cycle	3xUDDS cycle
CDCS	No	93.1	94.9
Rule-based	No	95.1	97.7
Manual CDCS	No	100.5	99.6
A-ECMS	Distance	98.3	104.7
MPC	Distance	97.4	105.4
Route-based EMS	Speed	103.4	107.7
MPC	Speed	102.9	108.4
PMP	Speed	103.8	108.4

close. In urban drive cycle (3xUDDS), A-ECMS strategy has a better performance, because the electrical energy is available until the end of the trip. In the combined urban and highway driving (EPA-UHU), Manual CDCS operates the engine more efficiently, and therefore, leads to better fuel economy.

If the predicted future speed trajectory is available, the devised Trip Planning module that generates optimum SOC profile can improve the performance of both MPC and Route-based EMS strategies. Using these control approaches, the fuel consumption is improved by 8.5% (102.9 vs. 95.1) and 10.2% (107.7 vs. 97.7) for EPA-UHU and 3xUDDS drive cycles comparing to the results of the default rule-based controller of Autonomie software.

The computational effort is the other criteria that should be considered in the EMS controller design in order to implement controls in real time. To compare the computational effort, all simulations are run on a PC with Intel Core 2 Duo CPU (E8500, 3.17 GHz) and 4 GB RAM. The average computation time of MPC and Route-based EMS strategies are 240, 208s for EPA-UHU drive cycle, and 290, 252 s for 3xUDDS drive cycle, respectively.

8.3 Control Prototyping via HIL

HIL simulation is an essential part of the ECU development process that evaluates and validates the ECU functions and communications between ECUs. The HIL simulations enable ECU testing under a variety of scenarios that may be very expensive or time consuming for vehicle drive test. In the HIL testing, ECUs are connected to a simulator instead of being connected to a real vehicle. The first step of HIL testing is prototype ECU preparation.

8.3.1 Controller Prototyping

The dSPACE MicroAutoBox II hardware platform is used as the prototype ECU for testing the real-time performance of the Route-based EMS strategy. The control signals generated by the EMS strategy are sent to the high-fidelity vehicle model in the simulator over the CAN network. Figure 8.20 illustrates the schematic of software architecture for both prototype ECU (MicroAutoBox) and simulator (DS1006 processor).

Figure 8.21 shows the control architecture of Toyota Prius Plug-in Hybrid provided by Toyota Information System (TIS) [16]. By comparing the current controller of the Prius with the designed energy-optimal controller, the difference is that the latter employs Trip Planning module to provide reference SOC profiles and also incorporate Route-based EMS instead of using rule-based control strategy.

The input signals to the prototype ECU are driver or cruise controller commands, Trip Planning signals, and feedbacks from the high-fidelity powertrain model; the output signals are the engine and motor-generators desired torques. These signals transfer to the simulator hardware through CAN bus. Table 8.3 represent the input and output signals and their characteristics in CAN bus.

8.3.2 HIL Testing Results

The results of the Route-based EMS system for different driving cycles are shown in Figs. 8.22 and 8.23. The HIL and MIL results are in complete agreement. The control

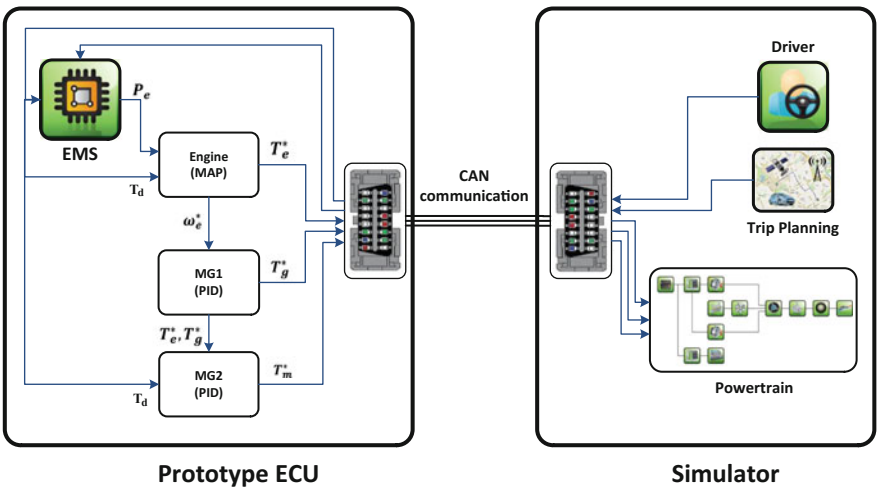


Fig. 8.20 Schematic of the Simulink models for HIL testing of the Route-based EMS strategy

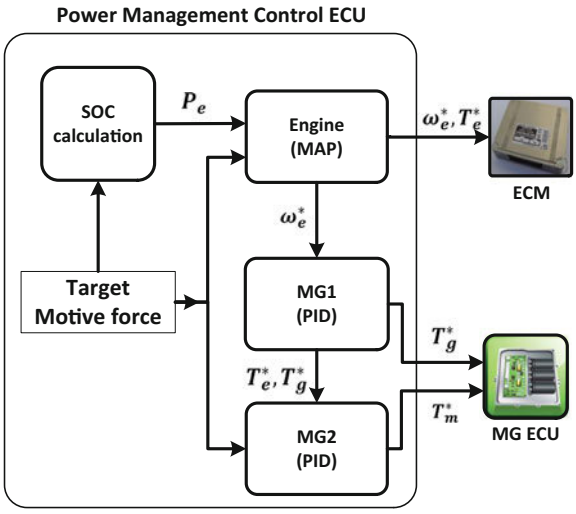


Fig. 8.21 Schematic of the Prius EMS from the TIS document [16]

Table 8.3 Specification of the input and output signals of the Route-based EMS

CAN signal	Variable name	CAN ID	Bit length
ECU inputs	Position	100	32
	Speed	101	32
	SOC	102	16
	Slope of Ref. SOC	120	16
	Initial position of Ref. SOC	121	16
	Initial SOC of Ref. SOC	122	16
	Demand torque	140	32
	Engine speed	151	32
	Motor speed	152	32
	Generator speed	153	32
ECU outputs	Engine torque	141	32
	Motor torque	142	32
	Generator torque	143	32

systems are basically the same. The only differences are in the solver program (C-code or Simulink) and the hardware (ECU or PC). The results show that the controller turnaround time is less than 25 μs, which is less than the desired time step (1 *ms*). Therefore, the Route-based EMS is validated.

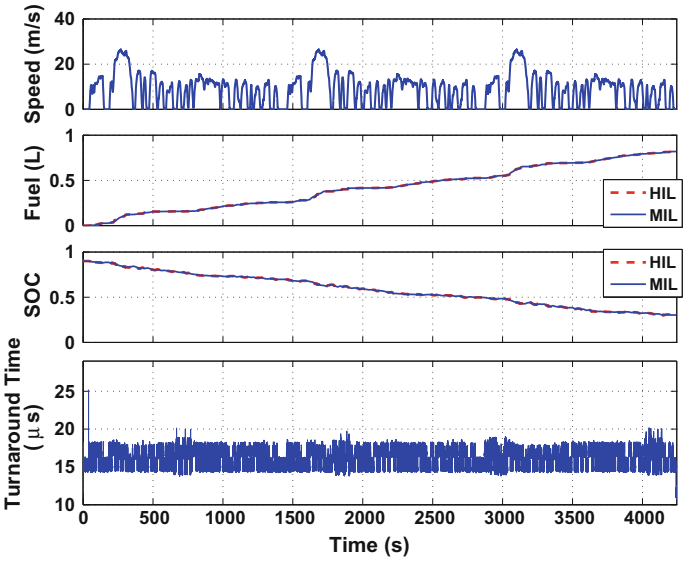


Fig. 8.22 HIL test results for the Route-based EMS over the 3xUDDS driving schedule

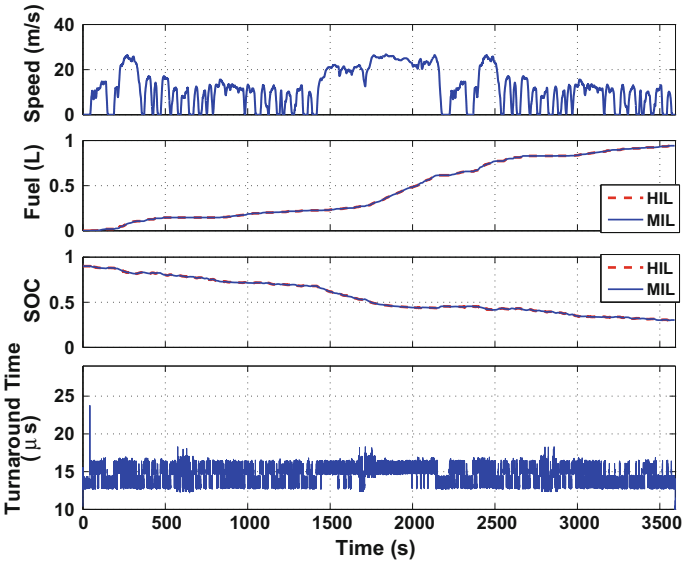


Fig. 8.23 HIL test results for the Route-based EMS over the EPA-UHU driving schedule

8.4 Summary

This chapter has presented a new real-time EMS system for PHEVs based on the ECMS approach. The designed controller takes advantage of preview trip information through optimum SOC profiles. The controller is implemented in the high-fidelity PHEV model and compared against the results of several different EMS strategies, including the CDCS, Manual CDCS, rule-based, PMP, and MPC methods. The real-time Route-based EMS system showed promising results and improved fuel consumption up to 11% compared to the rule-based controller. The HIL test results showed that the turnaround time is less than 25 μs, and the Route-based EMS system can be implemented in real-time.

References

1. Vajedi, M., Chehrehsaz, M., Azad, N.L.: Intelligent power management of plug-in hybrid electric vehicles, part II : real-time route based power management. Int. J. Electric Hybrid Veh. **6**, 68–86 (2014)
2. Mozaffari, A., Vajedi, M. , Chehrehsaz, M.,Azad, N.L.: Multi-objective component sizing of a power-split plug-in hybrid electric vehicle powertrain combined with optimal power management. Eng. Optim. **48**, 361–379 (2015)
3. Taghavipour, A. , Vajedi, M., Azad, N.L., McPhee J.: Predictive power management strategy for a PHEV based on different levels of trip information. Engine Powertrain Control Simul. Model. **3**, 326–333 (2012)
4. Vajedi, M. , Taghavipour, A., Azad, N.L., McPhee, J.: A comparative analysis of route-based power management strategies for real-time application in plug-in hybrid electric vehicles. Am. Control Conf. (ACC) (2014)
5. Taghavipour, A., Vajedi, M., Azad, N.L., McPhee, J.: A comparative analysis of route-based energy management systems for PHEVs. Asian J. Control **18**, 2939 (2015)
6. Kirk, D.E.: Optimal Control Theory: An Introduction. Courier Corporation (2012)
7. Keller, H.B.: Numerical Solution of Two Point Boundary Value Problems. SIAM (1976)
8. Holsapple, R.W., Venkataraman, R., Doman, D.: New, fast numerical method for solving two-point boundary-value. J. Guid. Control Dyn. **27**, 301–304 (2004)
9. Lewis, A.: The Maximum Principle of Pontryagin in Control and in Optimal Control. Universitat Politecnica de Catalunya (2006)
10. Taghavipour, A., Azad, N.L., McPhee, J.: An optimal power management strategy for power split plug-in hybrid electric vehicles. Int. J. Veh. Design **60**, 286–304 (2012)
11. Taghavipour, A.: Real-time Optimal Energy Management System for Plug-in Hybrid Electric Vehicles. Ph.D. thesis, University of Waterloo (2014)
12. Michalska, H., Mayne, D.Q.: Robust receding horizon control of constrained nonlinear systems. IEEE Trans. Autom. Control **38**, 1623–1633 (1993)
13. Parisini, T., Zoppoli, R.: A receding-horizon regulator for nonlinear systems and a neural approximation. Automatica **31**, 1443–1451 (1995)
14. Scokaert, P., Mayne, D., Rawlings, J.: Suboptimal model predictive control. IEEE Trans. Autom. Control **44**, 648–654 (1999)
15. Zheng, A., Morari, M.: Stability of model predictive control with mixed constraints. IEEE Trans. Autom. Control **40**, 1818–1823 (1995)
16. Hybrid/battery control: hybrid control system: details 2015 My Prius PHV, Doc ID: RM00000427300MX (2015)

Chapter 9
Ecological Cruise Control

In this chapter, an Eco-Cruise controller is developed to improve energy cost while maintaining a safe distance from the preceding vehicle. We show that the Eco-Cruise system can improve total energy costs, as well as vehicle safety, simultaneously. The developed controller is equipped with an onboard sensor to capture upcoming trip data to optimally adjust the speed of baseline PHEV. The NMPC technique is used to optimally control the vehicle's speed. To prepare the NMPC controller for real-time applications, a fast and efficient control-oriented model is developed. The NMPC controller is compared against three different controllers, i.e., PID, linear MPC, and PMP. Also, the real-time implementation of the controller is verified by HIL testing. Portions of this chapter have been published in [1–3].

9.1 Control-Oriented Modeling

In this section, a control-oriented model is developed to be used at the heart of the Eco-Cruise controller. The main goal of the proposed Eco-Cruise controller is to simultaneously maintain the safe distance of the host vehicle from the preceding vehicle and minimize its total energy costs. The tasks can be fulfilled by optimally adjusting the vehicle's speed for any given driving condition. In our control-oriented model formulation, speed and distance of the host vehicle and preceding vehicle are considered as the states of the system, and the wheels' torque is considered as the input of the system. Thus, the Eco-Cruise controller should adjust the wheels' torque to comply with the required tasks.

Figure 9.1 shows the schematic illustration of two consecutive vehicles in the traffic flow, in which L presents the intervehicle distance, L_d shows the desired intervehicle distance, and δ delegates the distance error.

A. Taghavipour et al., *Intelligent Control of Connected Plug-in Hybrid Electric Vehicles*,
Advances in Industrial Control, https://doi.org/10.1007/978-3-030-00314-2_9

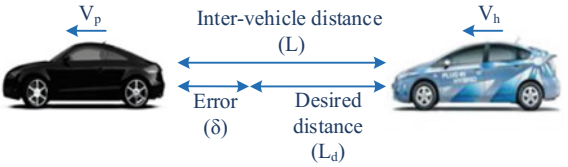


Fig. 9.1 Schematic of two consecutive vehicles

According to the vehicle longitudinal dynamics theory, the considered system can be modeled as below:

$$\begin{pmatrix} \dot{x}_h \\ \dot{v}_h \\ \dot{x}_p \\ \dot{v}_p \end{pmatrix} = \begin{pmatrix} v_h \\ u - \frac{1}{m} F_d \\ v_p \\ a_p \end{pmatrix} \tag{9.1}$$

$$F_d = \frac{1}{2}\rho A C_d (v_h)^2 + mg\sin\theta + mgf\cos\theta$$

where v_h and x_h are the speed and position of the host vehicle, and v_p and x_p are the speed and position of the preceding vehicle, u is input, F_d is resistance force, and a_p is the acceleration of the preceding vehicle. In this equation, u is equal to $u = \frac{1}{m.r} T_d$, where T_d is wheel torque and r is wheel radius.

To develop the controller for practical application and also to make sure the controlling commands always fall within a safe range, the desired intervehicle distance is adaptively changed by increasing the speed, as below:

$$L_d = L_0 + h v_h \tag{9.2}$$

where L_0 is stationary distance, and h headway time. The intervehicle distance error is calculated using Eq. 9.3:

$$\delta = L_d - L = (L_0 + h v_h) - (x_p - x_h) \tag{9.3}$$

It is assumed that access to the speed and acceleration of the preceding vehicle is available through measurement, or through the vehicle communication system. To improve the accuracy of the prediction model and enhance the controller performance, the acceleration of the preceding vehicle is adjusted during the prediction horizon.

By investigating the acceleration distribution in several drive cycles, it is found that the vehicle mostly drives at low acceleration and a constant speed (see Fig. 9.2). Therefore, the future acceleration of the preceding vehicle decreases gradually from the measured value.

$$\hat{a}_p(\tau) = e^{-\lambda\tau} a_p(t_k) \tag{9.4}$$

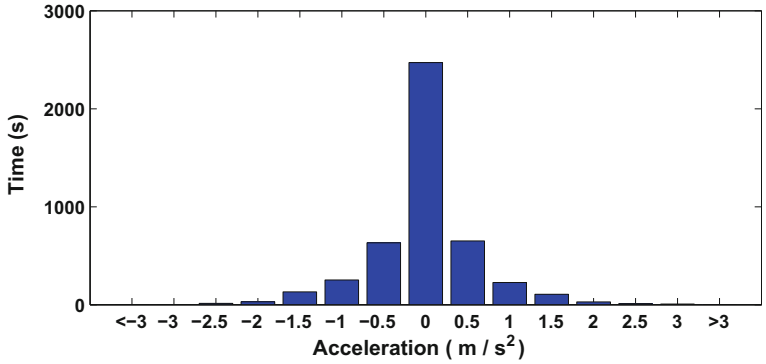


Fig. 9.2 Acceleration distribution in a combined UDDS, WLTP, HWFET, and SFTP drive cycle

where $\hat{a}_p$ is the predicted future acceleration of the preceding vehicle, $a_p(t_k)$ is the measured acceleration of the preceding vehicle, t_k is the previous time step, τ is prediction time, and λ is a constant positive parameter.

This model is same as Route-based EMS control-oriented powertrain model, described in Sect. 8.1. The amount of supplied engine and battery power strictly depend on the EMS strategy. It is assumed that the power distribution does not vary during a time step, and that the power distribution rate is the same as previous time step values.

$$\hat{P}_b(\tau) = \frac{P_b(t_k)}{P_d(t_k)} P_d(\tau) \tag{9.5}$$

$$\hat{P}_e(\tau) = \frac{P_e(t_k)}{P_d(t_k)} P_d(\tau) \tag{9.6}$$

where P_d is power demand, $\hat{P}_e$, $\hat{P}_b$ are predicted engine power and battery power, $P_e(t_k)$, $P_b(t_k)$ are measured engine power and battery power in the previous time step, and τ is prediction time, respectively.

9.2 Controls Design

This section provides the mathematical steps required for the implementation of the controller. Figure 9.3 clarifies the details of this controller schematically. The simulation model includes four main blocks: the environment, Eco-Cruise controller, EMS controller, and PHEV powertrain model.

Based on the vehicle's position, the intervehicle distance, preceding vehicle speed, and road grade are obtained and fed to the Eco-Cruise controller. The controller optimally calculates the wheel torque considering driving safety and energy cost.

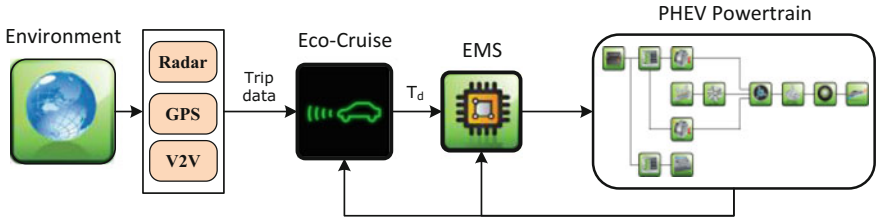


Fig. 9.3 Schematic of Eco-Cruise controller for PHEVs

The EMS distributes the power demand between the two energy sources and determines the optimum torques of the engine and motor-generators. Finally, in the high-fidelity PHEV powertrain model, vehicle speed, fuel consumption, and SOC are calculated. This chapter investigates and compares four different control techniques for the Eco-Cruise controller: PMP, NMPC, LMPC, and PID.

9.2.1 Nonlinear Model Predictive Control

In this section, the Eco-Cruise controller is designed based on the NMPC approach to find the optimum wheel torque. In contrast to PMP formulation, the future position and speed of the preceding vehicle are unknown. Equation 9.4 is applied to predict the acceleration of preceding vehicle. The problem formulation is given in Eq. 9.7:

$$\begin{aligned} J &= \int_0^{t_f} \left(\omega_1\, \delta^2(X) + \omega_2\, Cost(X, u, PR)\right)\, dt \\ \dot{X} &= \mathcal{F}(X, u, \hat{a}_p) \\ v_{min} &\le v_h \le v_{max} \\ u_{min} &\le u \le u_{max} \end{aligned} \tag{9.7}$$

Using the control-oriented model, the states of the system are predicted for any given prediction horizon (τ), and consequently, the objective function is evaluated in each time step. To solve the above optimal control problem, an optimization algorithm should be taken into account. Many researches apply particle swarm optimization (PSO) to solve the optimization inside MPC technique and calculate controlling commands [4–9].

PSO is a stochastic and population-based optimization search algorithm which is mainly inspired by social behavior of natural systems. Let us consider a large group of individuals interacting within a social system such as in flocks of birds or swarms of bees seeking a location with abundant food. In such system, each individual has two properties, a current position and a velocity. Also, each individual has knowledge

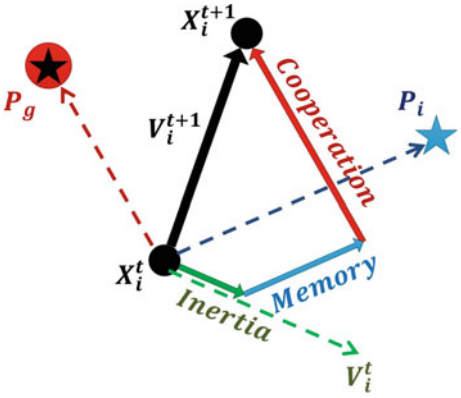


Fig. 9.4 Velocity and position adjustment of a particle in the PSO algorithm

of its best position so far plus the knowledge of the best-found position of the group. Now each individual can correct its properties (position and velocity) according to this information.

The PSO algorithm works based on the same principle as that of the social system described above. It simulates a simplified social system with moving particles in a multi-dimensional search space. Each particle is dynamically adjusting its own velocity and position in each step based on its history (its best position experienced so far) and those of its peers. Thus, PSO can obtain the global optimum in optimization problems as a result of a global behavior and interaction between all of the particles [10, 11].

Figure 9.4 shows the schematic of particle position adjustment in the PSO algorithm. The velocity and position of each particle is updated in each iteration using Eqs. 5.13 and 5.14.

$$V_i^{t+1} = \Omega V_i^t + C_1 \Phi_1 (P_i - X_i) + C_2 \Phi_2 (P_g - X_i) \tag{9.8}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{9.9}$$

where V, X are velocity and position of the particle, i is the index for particle number, t is the iteration number, ω is the inertia weight, Φ_1, Φ_2 are random numbers, P_i is the best position of the particle, and P_g is the best position of the swarm.

The meta-optimization is used to tune PSO algorithm. Figure 9.5 shows the schematic of the meta-optimization. The optimization parameters are number of particles, maximum iteration, maximum particle velocity, and inertia weight. The objective is to minimize computation time of the PSO optimization. The DP algorithm is applied to solve the meta-optimization problem off-line and tune the PSO algorithm.

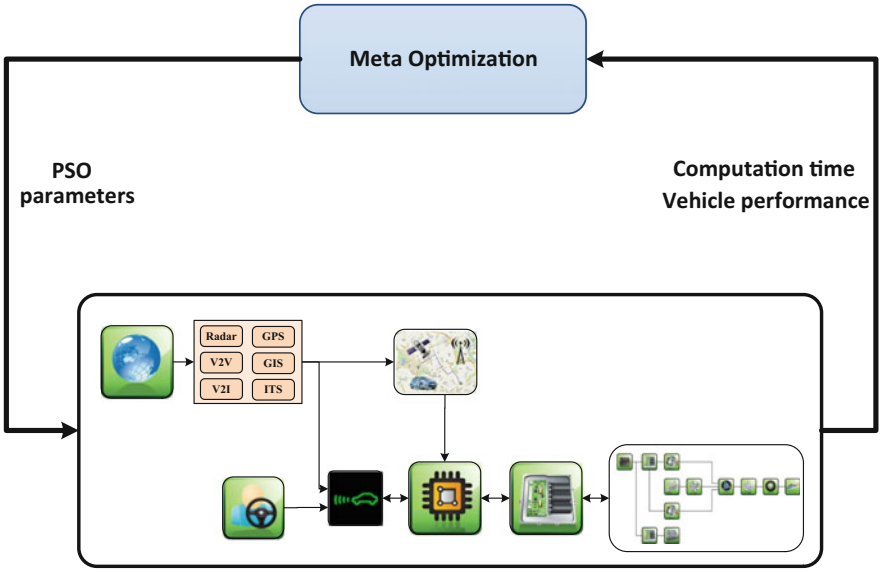


Fig. 9.5 Schematic of the meta-optimization algorithm to tune PSO optimization

9.2.2 Linear Model Predictive Control

To design and implement a LMPC for the Eco-Cruise controller, the model is linearized and the objective function is derived in the quadratic form.

$$\begin{aligned} \mathcal{J} &= \int_0^{t_f} \left(Q\,(v_h - v_{ref})^2 + R\,(u - u_{ref})^2 + \omega_1\,\delta^2 + \omega_2\,Cost \right)\,dt \\ \dot{X} &= AX + Bu \\ v_{min} &\leq v_h \leq v_{max} \\ u_{min} &\leq u \leq u_{max} \end{aligned} \tag{9.10}$$

where Q, R are positive definite matrices and v_{ref}, u_{ref} are the reference values of speed and control input, respectively.

The objective is to follow the preceding vehicle, the reference speed is considered to be the speed of the preceding vehicle in the previous time step. Reference control input is required input for tracking reference speed, and can be obtained from Eq. 9.1 based on $\dot{v}_h = 0$, $v_h = v_{ref}$. MATLAB toolbox is used to solve the optimal control problem.

9.3 Results

This section presents the simulation results for energy-optimal controller with all three modules: Trip Planning, Route-based EMS, and Eco-Cruise controller. The objective of the Eco-Cruise controller is to maintain a safe distance from the preceding vehicle while considering energy cost.

The safe distance between vehicles depends on speed and traffic conditions. The desired intervehicle distance is obtained based on the stationary distance and headway time in Eq. 9.2. This chapter considers two different traffic conditions, congested and uncongested low traffic flow, and obtained optimum traffic parameters from [12], as reported in Table 9.1.

The results of the NMPC controller are compared with those of two other controllers, the LMPC and PID controllers. For an unbiased comparison, a fine-tuning procedure is carried out on the controllers.

Figures 9.6, 9.7, and 9.8 show the simulation results for following a vehicle with different speed trajectories in congested low traffic flow. All controllers follow the preceding vehicle with the acceptable distance error, $(-1 < \delta < 1)$, but the control performances are all different. The results show that the NMPC yields the lowest energy cost, compared to the other controllers.

Figures 9.9, 9.10, and 9.11 show the results of Eco-Cruise controller for both congested and uncongested conditions. In the congested condition, the acceptable intervehicle distance error is relatively small, $\delta_{max} = 1$, and the cruise controller primarily considers safety as an objective function. In the uncongested condition, the acceptable intervehicle distance error is more than the congested one ($\delta_{max} = 3$). Therefore, the cruise controller can adjust the speed cost-effectively by accepting a small intervehicle distance error, while improving the performance of the vehicle.

Up to this point, the Route-based EMS is applied to calculate optimal power distribution. To find the effect of each module in energy-optimal control scheme, the cruise controllers are applied to the rule-based EMS of the Autonomie model. The simulation results for different control strategies under different conditions are shown in Table 9.2.

The results show that the NMPC controller can improve energy costs by up to 20% while maintaining driving safety. This improvement is achieved through optimizing the speed profile and reducing the acceleration of the vehicle. NMPC has smaller acceleration than the PID controller. In congested traffic conditions, the acceleration is increased. This observation justifies the fact that, in such a condition, the cruise controller function mainly focuses on vehicle safety and cannot widely adjust its speed.

Table 9.1 Parameters of car-following model

Traffic condition	L_0	h	δ_{max}
Congested low traffic flow	2	4	1
Uncongested low traffic flow	4	8	3

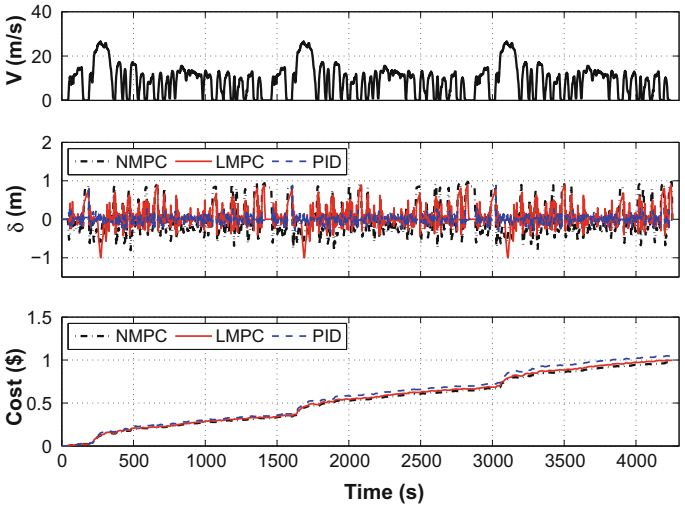


Fig. 9.6 Car-following simulation results in congested traffic conditions for the following 3xUDDS cycle

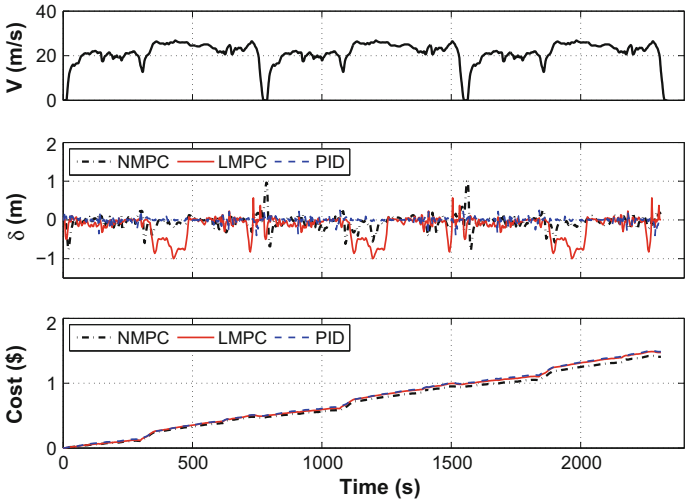


Fig. 9.7 Car-following simulation results in congested traffic conditions for the following 3xHWFET cycle

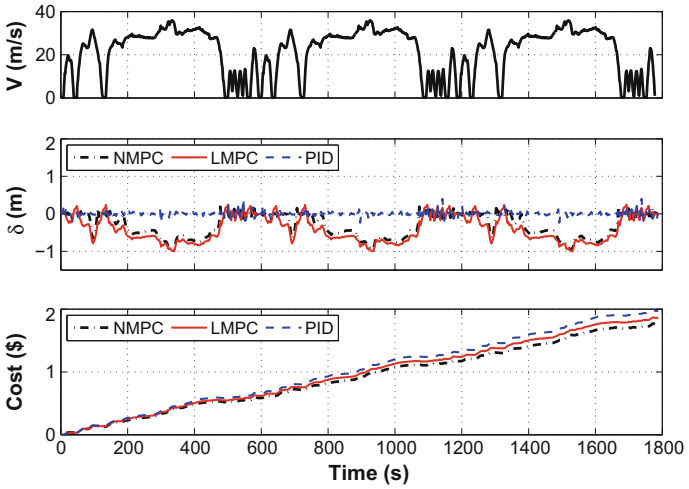


Fig. 9.8 Car-following simulation results in congested traffic conditions for the following 3xSFTP-US06 cycle

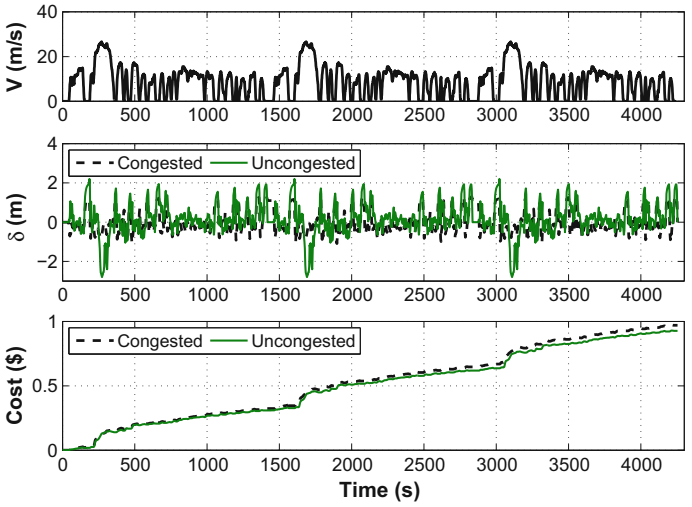


Fig. 9.9 Comparison of Eco-Cruise controller results in congested and uncongested traffic conditions for the following 3xUDDS cycle

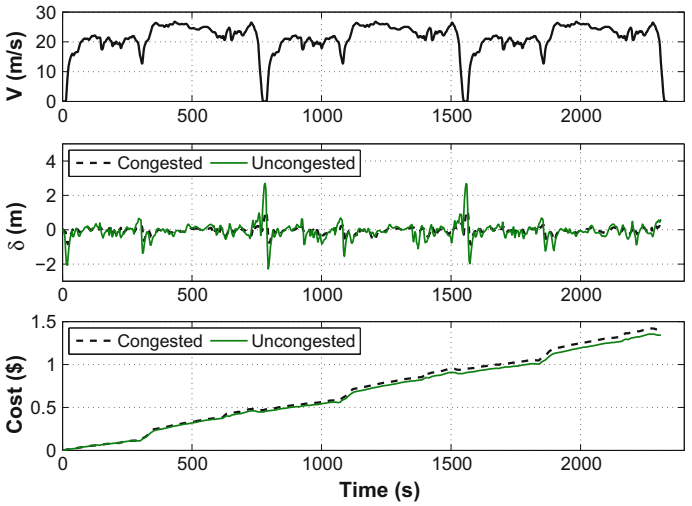


Fig. 9.10 Comparison of Eco-Cruise controller results in congested and uncongested traffic conditions for the following 3xHWFET cycle

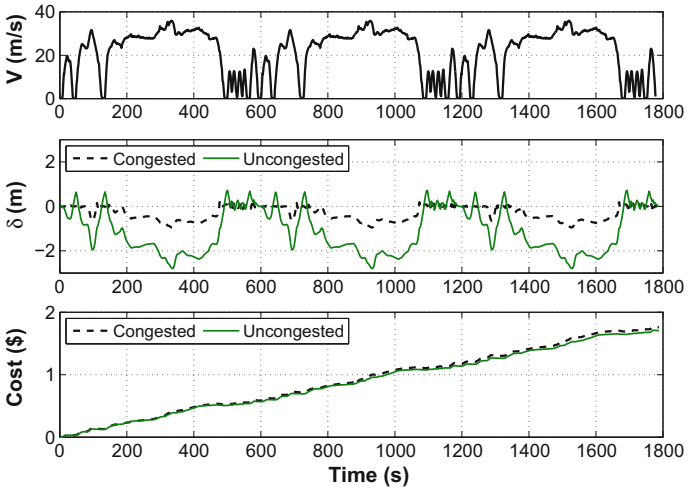


Fig. 9.11 Comparison of Eco-Cruise controller results in congested and uncongested traffic conditions for the following 3xSFTP-US06 cycle

Table 9.2 Fuel economy for different cruise controllers

Cruise	EMS	Fuel consumption (MPG)					
		Congested traffic			Uncongested traffic		
		3xUDDS	3xHWFET	3xSFTP	3xUDDS	3xHWFET	3xSFTP
PID	Route-based	101	89	51	101	89	51
	Autonomie	93	75	47	93	75	47
LMPC	Route-based	108	90	54	114	95	58
	Autonomie	96	77	51	102	81	56
NMPC	Route-based	111	95	58	118	100	60
	Autonomie	97	78	54	107	85	63

9.4 HIL Testing Results

Real-time HIL testing promises an effective approach for validation of vehicle control systems. There has been an increasing need for real-time simulation and support for HIL in automotive applications. Section 7.3.2 described the HIL test platform. It consists of a real-time simulator, prototype ECU, and interface computer. These components are connected through a CAN bus. The prototype ECU executes the controller and obtains optimal controller commands. The real-time simulator runs the high-fidelity model and calculates powertrain variables. These variables are fed back to the controller. An interface computer is used to set up an HIL test, program the ECU and simulator, and record the desired outputs and variables.

9.4.1 Controller Prototyping

Figures 9.12 and 9.13 show the cruise controller architectures of Toyota Prius (from TIS documents [13]) and the designed energy-optimal controller. Both controllers use input signals from radar and feedback signals from powertrain, and send command signals to energy management ECU. The main difference between these two controllers is the control techniques.

Table 9.3 characterizes the CAN communication signals between the simulator and prototype ECU. It should be noted that it is assumed that the raw radar signals are preprocessed by another algorithm. Therefore, intervehicle distance, speed, and acceleration of the preceding vehicle are available for the Eco-Cruise controller.

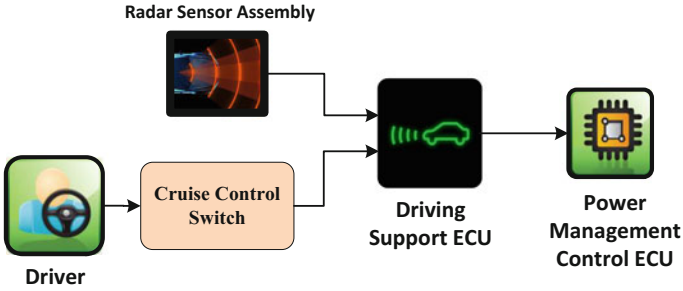


Fig. 9.12 Schematic the Prius Cruise controller from TIS document [13]

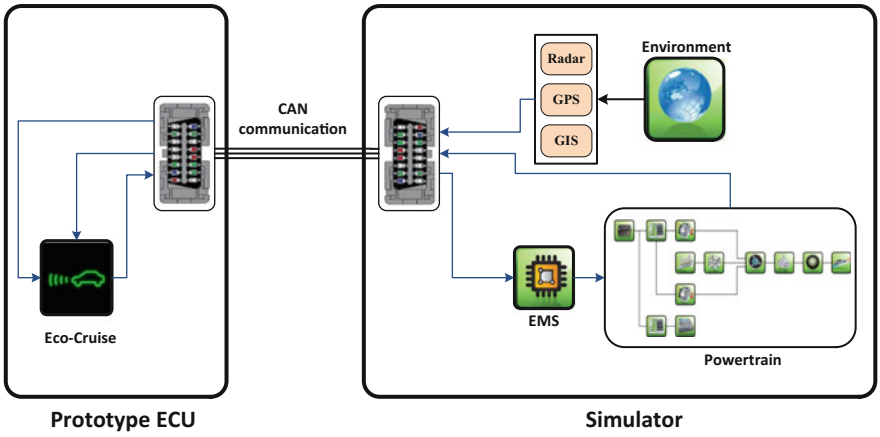


Fig. 9.13 Schematic of the Simulink models for HIL testing of the Eco-Cruise controller

Table 9.3 Specification of the input and output signals of the Eco-Cruise controller

CAN signal	Variable name	CAN ID	Bit length
ECU inputs	Speed	101	32
	Intervehicle distance	180	16
	Acceleration of the preceding vehicle	181	16
	Speed of the preceding vehicle	182	16
	Rate of SOC	160	16
	Rate of fuel	161	16
ECU outputs	Demand torque	140	32

9.4.2 HIL Testing Results

The HIL and MIL results of the NMPC controller for different driving cycles are shown in Figs. 9.14 and 9.15. The HIL and MIL results are in complete agreement. The HIL results show that the turnaround time in the prototype ECU is less than 55 μs.

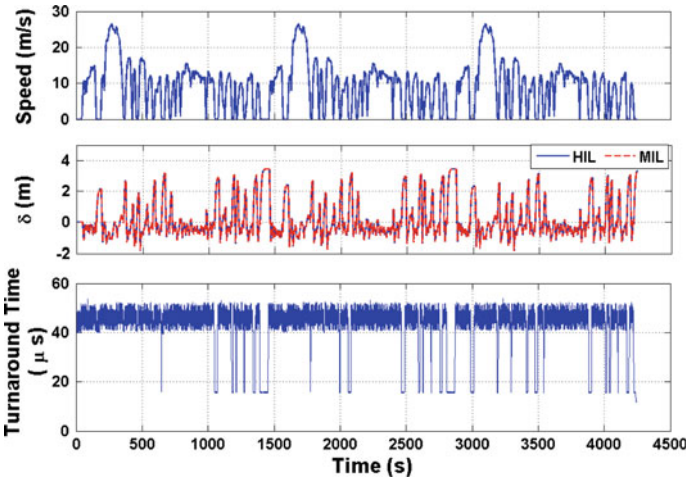


Fig. 9.14 HIL test results for the NMPC Eco-Cruise controller following the 3xUDDS driving schedule

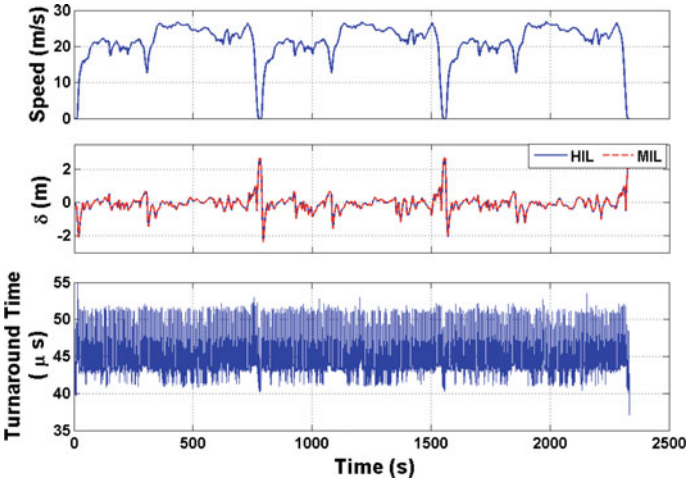


Fig. 9.15 HIL test results for the NMPC Eco-Cruise controller following the 3xHWFET driving schedule

The desired sample time in the Prius adaptive cruise ECU is 1 ms. The HIL tests verify that the turnaround time is less than the desired time step. Hence, the NMPC controller can be implemented in real-time.

9.5 Summary

In this chapter, an Eco-Cruise controller was designed for a PHEV based on nonlinear model predictive control. The controller optimally adjusts the vehicle speed to enhance driving safety and minimizes total energy costs.

First, the control-oriented model is developed and validated using the high-fidelity PHEV model. The simulation results show that the control-oriented model can predict vehicle performance with an error of less than 3.6%. Thereafter, the model is used at the heart of the NMPC technique to design the Eco-Cruise controller. PMP technique is applied to find the global optimum solution of the cruise control problem. To endorse the efficacy of the NMPC, its performance is evaluated against the PID, LMPC, and PMP controllers under different driving conditions.

The results indicate that for a vehicle driving over a hill, the Eco-Cruise controller takes advantage of the trip information to speed up the vehicle before the uphill, which improves the energy cost up to 15%. The results show that NMPC results are in agreement with the global optimum solution.

For the car-following scenario, the results show that the NMPC technique outperforms the PID and LMPC controllers. It is found that the performance of the Eco-Cruise controller depends on traffic conditions. In congested low traffic flow conditions, the cruise controller mainly considers the safety criteria and, as a result, the bound of allowable control command is very narrow due to small intervehicle distance. For uncongested low traffic flow conditions, the acceptable intervehicle distance error is relatively large, which enables the controller to improve the performance while considering the safety criterion. The feedback of the simulation revealed that NMPC shows promising results and can improve energy costs of a vehicle by up to 20%.

HIL test results show that the turnaround time is less than 55 μ s, and the NMPC-based Eco-Cruise controller can be implemented in real-time.

References

1. Vajedi, M., Azad, N.L.: Ecological adaptive cruise controller for hybrid electric vehicles using nonlinear model predictive control. IEEE Trans. Intell. Transp. Syst. **17**, 113–122 (2015)
2. Mozaffari, A., Vajedi, M., Azad, N.L.: A robust safety-oriented autonomous cruise control scheme for electric vehicles based on model predictive control and online sequential extreme learning machine with a hyper-level fault tolerance-based supervisor. Neurocomputing **151**, 845–856 (2014)

3. Sakhdari, B. Vajedi, M., Azad, N.L.: Ecological adaptive cruise control of connected plug-in hybrid electric vehicles for urban driving. In: American Control Conference (ACC) (2015)
4. Han, M., Fan, J., Wang, J.: A dynamic feedforward neural network based on Gaussian particle swarm optimization and its application for predictive control. IEEE Trans. Neural Netw. **22**, 1457–1468 (2011)
5. Pourjafari, E., Mojallali, H.: Predictive control for voltage collapse avoidance using a modified discrete multi-valued PSO algorithm. ISA Trans. **50**, 195–200 (2011)
6. Sandou, G., Olaru, S.: Nonlinear Model Predictive Control, pp. 551–559. Springer (2009)
7. Song, Y., Chen, Z., Yuan, Z.: New chaotic PSO-based neural network predictive control for nonlinear process. IEEE Trans. Neural Netw. **18**, 595–601 (2007)
8. Wang, X., Xiao, J.: Advances in Natural Computation: PSO-based Model Predictive Control for Nonlinear Processes, pp. 196–203. Springer (2005)
9. Yousuf, M.S., Al-Duwaish, H.N., Al-Hamouz, Z.M.: Pso based nonlinear predictive control of single area load frequency control. In: IFAC Workshop on Control Applications of Optimization (2009)
10. Clerc, M.: Particle Swarm Optimization. Wiley (2010)
11. Trelea, I.: The particle swarm optimization algorithm: convergence analysis and parameter selection. Inf. Process. Lett. **85**, 317–325 (2003)
12. Ye, F., Zhang, Y.: Vehicle type-specific headway analysis using freeway traffic data: transportation research record. J. Transp. Res. Board **2124**, 222–230 (2009)
13. Toyota Information System: Cruise control: dynamic radar cruise control system: system diagram 2015 MY Prius PHV, Doc ID: RM00000427F01BX (2015)

Chapter 10
Conclusions

10.1 Part I

In this part, a near-optimal EMS for a plug-in hybrid electric powertrain was proposed to minimize both fuel consumption and emissions, using a model predictive control (MPC) approach.

At first, a real-time, equation-based, and validated high-fidelity simulation model of a plug-in hybrid electric powertrain was developed in the MapleSim software in order to be used in controls performance evaluation procedure such as model-in-the-loop (MIL) and hardware-in-the-loop (HIL) simulations. The chemistry-based model of the battery in the powertrain model, leads to more realistic estimation of the PHEV fuel economy and range. The parameters of the PHEV powertrain model were adjusted based on the experimental database of the Autonomie software, which is widely used for energy management design in industry. The symbolic programming capability of Maple allows reducing the number of equations involved in the powertrain model significantly, and makes it run in real time, which is essential for performing HIL tests.

For implementing the EMS to the high-fidelity simulation model, another level of controls were designed for the engine and electric drive, using sliding mode control and field-oriented control approaches, respectively. One of the features of the engine control system is improving the hydrocarbon conversion efficiency of the catalytic converter.

Due to some real-time implementation problems of MPC despite its near-optimal performance in improving fuel economy, another PHEV EMS was designed using explicit model predictive control (eMPC), which solves the optimization problem off-line to create some lookup tables. Therefore, the problem is reduced from solving a quadratic programming problem at each control sampling time, to search in lookup tables while implementing the control algorithm. As a result, eMPC can guarantee real-time implementation for a fairly small control-oriented model. In order to keep the size of mentioned lookup tables small enough to be implemented to a commercial control hardware with limited amount of flash memory, a simple and

A. Taghavipour et al., *Intelligent Control of Connected Plug-in Hybrid Electric Vehicles*,
Advances in Industrial Control, https://doi.org/10.1007/978-3-030-00314-2_10

innovative control-oriented model was proposed. According to MIL simulation results, the designed eMPC energy management can be implemented to the high-fidelity simulation model which is three times faster than its implicit MPC counterpart.

For further improvement of energy management strategy performance, a control-relevant parameter estimation (CRPE) approach was used to make the control-oriented model more accurate. The resultant CRPE-control-oriented model is accurate within a specific frequency range that is excited by the EMS. Based on the newly developed control-oriented model, CRPE-eMPC EMS was designed. The MIL simulation revealed that the CRPE-eMPC EMS could reduce the fuel consumption up to 6.5% as compared to the eMPC EMS.

Finally, the CRPE-eMPC energy management strategy was implemented on MotoTron hardware with limited computation and memory capabilities. The aforementioned high-fidelity simulation model was used as the virtual simulation model inside the real-time target which was connected to the hardware via CAN bus. The HIL simulation showed that the proposed EMS could be implemented in real time and result in a promising fuel economy improvement up to 16.5% as compared to the baseline strategy by controlling the emissions. Note that if the emissions control is not considered in the control scheme, the fuel economy improvement is even higher.

The achievement in Part I can be summarized as follows:

- Developed real-time, equation-based, and cross-validated high-fidelity simulation model of a PHEV powertrain.
- Model predictive control design and evaluation for a PHEV EMS.
- Developed and validated near-optimal and real-time implementable PHEV EMS using explicit model predictive control (eMPC) approach with simple and innovative control-oriented model for simpler stability analysis.
- Developed control-relevant parameter estimated (CRPE) control-oriented model to improve the performance of eMPC EMS while maintaining its real-time capabilities.
- Implemented the CRPE-eMPC energy management strategy on a commercial control hardware with limited computational and memory capabilities.

10.2 Part II

This part has described the development of an energy-optimal controller for a PHEV. The controller consists of three main modules: the Trip Planning, Route-based EMS, and Eco-cruise control. The Trip Planning module takes advantage of trip information to predict future speed trajectory and optimizes SOC profiles to minimize total energy cost. Route-based EMS calculates optimal power distribution between the battery and engine. The Eco-Cruise controller optimally adjusts speeds in real-time to improve diving safety and total energy cost.

The control-oriented models are developed for each module, and validated against the high-fidelity PHEV model of the Toyota Prius in Autonomie. Then, the controllers are evaluated using MIL and HIL testing.

In Trip Planning, the RCO algorithm is developed based on data clustering technique. The results are evaluated against DP results. The RCO algorithm is very promising and can find the optimum SOC profile with less than 2.4% error with respect to the DP results.

Route-based EMS is developed based on ECMS strategy. The controller performance compared against CDCS, Manual CDCS, rule-based, PMP, and MPC techniques under various driving conditions. The Route-based EMS results are in agreement with the results from PMP and MPC techniques. They can improve fuel consumption up to 11% compared to rule-based controllers. The designed controller is evaluated using HIL testing and showed a turnaround time of less than 25 μs. Thus, the devised controller is capable of real-time implementation in the ECU hardware.

The Eco-Cruise controller is designed based on the NMPC technique. To evaluate the efficacy of the controller, its performance is compared against the PID, LMPC, and PMP controllers under a variety of driving conditions. The NMPC results are in agreement with the global optimum solution of PMP controller. It is also observed that the NMPC technique can improve energy economy by up to 20% compared to the PID controllers. HIL test results show that the turnaround time of the controller is less than 55 μs, and then the real-time Eco-Cruise controller is validated.

This section summarizes the major contributions of Part II.

1. Design of a new algorithm for Trip Planning, which
   - Employs future driving conditions in a real-time optimization; and
   - Incorporates a novel RCO algorithm that is feasible for real-time implementation.
2. Design of a new Route-based EMS strategy for a power-split PHEV, which include
   - Development of a novel architecture that use trip data to obtain a nearly optimal solution; and
   - Implementation in real-time.
3. Development of a novel technique for an Eco-Cruise controller, which
   - Considers both fuel consumption and safety in the cruise controller;
   - Is independent from the EMS system; and
   - Implements in real-time.

10.3 Recommendations for Future Research

Although the performance of the proposed EMS for improving fuel economy was shown through MIL and HIL simulations, there is still room for improvement in both controls design and validation stages. The recommended future research for each part is separately mentioned below:

10.3.1 Controls Design

10.3.1.1 Virtual Simulation Model Improvement

The proposed MapleSim model of PHEV in Chap. 3, should be run in a real-time target with multi-threaded CPU in order to include the power electronics high-fidelity simulation model. Furthermore, the model parameters can be validated through the data from a prototype on a rig or a test bench of powertrain real components for even more realistic estimation of powertrain behavior.

10.3.1.2 Smart PHEV

As mentioned, PHEV performance is closely related to the battery depletion profile along the driving schedule. More information from the trip can significantly improve EMS performance. This improvement can be made in two ways:

- Short-horizon vehicle velocity prediction in order to take advantage of MPC predictive feature by using global positioning system (GPS), geographic information system (GIS), and intelligent traffic systems (ITS).
- Optimized SOC depletion profile (as shown in Chap. 4).

By considering the above items, one can design a smart EMS which pushes the PHEV performance to its ultimate limit.

10.3.1.3 Fast MPC Approach

Basically, two approaches to fast quadratic programming (QP) solution in MPC can be distinguished: first, the explicit, or off-line QP solution, as used in this part, which is limited to models with small state dimensions and few process inputs. Second, the online QP solution is the classical way to treat the sequence of QPs in MPC for varying initial process values.

As a result, for the designed MPC EMS in Chap. 4, faster QP solvers (e.g., active set method) can be used. By using this fast MPC approach there is a possibility of improving MPC energy management real-time capabilities. Then, fast MPC performance can be compared to eMPC energy management as proposed in Chap. 5.

10.3.1.4 Sensitivity Analysis

Sensitivity analysis can be done on the parameters of EMS such as control and prediction horizon length as well as cost function weighting parameters by using either classical MPC or the explicit version. This analysis is helpful during the energy management calibration stage for expediting the procedure and taking advantage of a model-based control approach.

10.3.1.5 Control-Oriented Model Improvement

Control-relevant parameter estimation (CRPE) for control-oriented model can be done for different driving schedules in order to find a more precise active frequency range for the battery. This leads to a more accurate control-oriented model and consequently guarantees EMS's best performance along any driving scenario. Moreover, CRPE can be performed for control-oriented models with higher order to obtain the most accurate control-oriented model while maintaining real-time capability of the EMS.

In this way, one can find a relation between control-oriented model fidelity and real-time capability of eMPC.

10.3.2 Controls Validation

10.3.2.1 Energy Management System Calibration

ECU calibration is an iterative process of measurement and calibration at runtime to optimally tune the parameters of the ECU algorithms. The parameters of the proposed EMS should be tuned during calibration procedure, for the best possible performance on a PHEV prototype.

Obviously, the model-based control approach can expedite this stage and make tuning of the parameters and weights more systematic.

10.3.2.2 Energy Management System on a PHEV Test Bench

In order to find real-world performance of the EMS, it should be implemented to the real components of a PHEV on a test bench.

10.3.3 Smart PHEV

This part demonstrates the performance of an energy-optimal controller for improving total energy cost and driving safety through MIL and HIL simulations. However, further research is needed to expand this study. This section provides some recommended future work for each control system.

1. Trip Planning module:
   - Improve the control-oriented model by considering the engine transient effect;
   - Include traffic light time schedules in the traffic model; and
   - Improve the robustness of the designed controller in cases of data loss and missed prediction situations.

2. Route-based EMS:
   - Improve the controller performance using short-horizon speed prediction; and
   - Improve the robustness of the control system.
3. Eco-Cruise controller:
   - Include a radar model to make the high-fidelity model more accurate;
   - Add powertrain delay into the control-oriented model; and
   - Improve the robustness of the control system against disturbances.

Appendix A
Hardware-in-the-Loop Procedure

In this chapter, the essential elements for performing hardware-in-the-loop (HIL) testing are introduced. Then, the eMPC EMS is programmed on a MotoTron electronic control system (ECU) hardware and its performance is evaluated through HIL testing.

A.1 Introduction

Real-time hardware-in-the-loop testing is an effective method for rapid prototyping and assessment of vehicle control system. Real-time simulation and support for HIL are often identified as indispensible tools for engineering design [1].

This diminishes dependency on prototype vehicles, particularly in the early stages of a program, and also subsequently lessens the time, effort, and resources necessary to build and support them. The objective is to construct the first full prototype case at a later stage of the development program, and in a manner that requires minimal modification prior to mass production [2].

Most of the time HIL simulation requires significantly less harware, as compared to physical prototyping, which provides for rapid and cost efficient construction. HIL simulators often achieve fidelity levels unattainable through purely virtual simulation by prototyping those components whose dynamics or other attributes (e.g., transient emission formation in engines) are not fully understood. (citation may be required) However, HIL simulations of complex physical phenomena run faster than purely virtual simulations of the same phenomena (e.g., IC engine simulations based on Computational Fluid Dynamics).

Systems that normally operate in highly variable environments (e.g., off-road vehicle suspension systems) can often be tested in controlled lab settings through HIL simulation, which significantly increases repeatability, and often makes it possible to simulate a given system over a much broader range of operating conditions than what is feasible via purely physical prototyping. Moreover, HIL simulation makes it possible to simulate destructive events without incurring a costly destruction. HIL

A. Taghavipour et al., *Intelligent Control of Connected Plug-in Hybrid Electric Vehicles*,
Advances in Industrial Control, https://doi.org/10.1007/978-3-030-00314-2

simulators can be used to train human operators (e.g., airplane pilots) of safety-critical systems (e.g., supersonic aircraft) in significantly safer environments (e.g., flight simulators). It is worth noting that HIL simulation allows different teams to develop different parts of a system in hardware without losing sight of integration issues, thereby enabling concurrent systems engineering [3]. In this chapter, the HIL testing procedure of the proposed eMPC EMS for Toyota Prius plug-in hybrid powertrain is presented.

A.1.1 ECU Validation Procedure

The electronic control unit (ECU) strategy prove-out is done in successive steps on off-line simulations on a desktop, HIL, dynamometer, and vehicle, with each step bringing in additional "real" substitutes for the virtual models.

ECU strategy procedure in this sequence has some advantages. First, it ensures that component-level testing is done prior to subsystem and system level testing. Second, it capitalizes on the fact that ECUs are usually available much sooner than vehicle hardware prototypes, enabling a large amount of testing to be completed prior to vehicle manufacturing [4]. Figure 6.1 shows the steps in ECU validation [5] (Fig. A.1).

The off-line simulations used within the early phases of the development process are often referred to as model-in-the-loop simulations (MIL). For the modeling of the vehicle and functions at the MIL stage, standard tools such as MATLAB/Simulink and MapleSim can be used. The next step is software-in-the-loop (SIL) simulation, where the functional model of an ECU is replaced by C-code and coding errors can be found independent of the future ECU hardware. In the next step, actual hardware of the ECU is available and the tests can be supported by HIL simulation. The

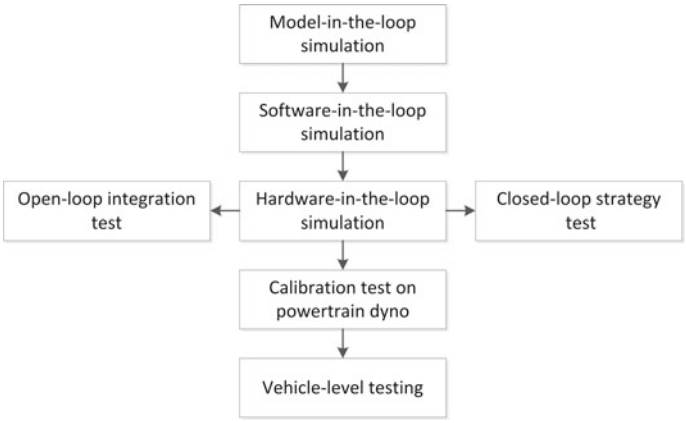


Fig. A.1 ECU Validation Procedure

HIL simulation consists of two parts: open-loop integration and closed-loop strategy testing [5]. Open-loop test platform uses a simpler model inside a real-time computer in order to check the functionality of user inputs and low-level I/O interfaces, for instance, push button start, or gear shift command. Closed-loop test platform needs the dynamical model of the plant implemented to the real-time computer which provides feedback information from the plant.

After the software tests are successfully passed, the calibration of the ECUs can be done on the test bench or in the vehicle. At this point, changes to the functions and system specifications are time consuming, expensive, and in most cases not possible [6].

The explained validation procedure presents some challenges for HEV control system unit such as [4]: (1) adequate computational power required to execute not just an engine plant model, but models of other controller units as well as a complete vehicle model and (2) capability of extensive CAN communication support, since the HEV controllers that are communicating with the ECU should be modeled [7].

A.1.2 Virtual Simulation Model Requirements

Since the performance of the ECU is tested in a virtual vehicle environment, the appropriate vehicle dynamics need to be modeled. There are four important elements to the HIL system model that is used for ECU testing: physical model of the plant (such as engine, transmission, battery, motor, etc.), sensors (such as sensors for engine speed, engine intake manifold pressure, battery voltage, etc.), actuators (such as electronic throttle body, fuel injector, etc.), and external systems that interact with the vehicle model (such as human driver, road grade, etc.) [7].

The scope of the HIL system model is primarily driven by the level of functional testing required. On one hand, the HIL test bed may be used as an open-loop tester to verify some low-level (input/output driver level) ECU software functionality. In this case, the vehicle model can be constant values or uncorrelated signal traces that drive all the inputs of the controller. On the other hand, the HIL test bed may be used for verifying closed-loop dynamic functionality [4].

To use the HIL simulation, real-time capable simulation models are needed. Developing those models is a sizeable challenge, since they have to be accurate and fast enough at the same time [6].

The virtual models within a HIL simulator must typically meet two requirements. First, they must capture the essential dynamics of the virtually prototyped systems accurate enough to enable the HIL simulator to achieve its required design goals. Second, they must run in real time, a requirement that often translates into a bound on model complexity. These two requirements that are, fidelity and simplicity, typically conflict [3]. The literature recognizes this conflict and considers a dynamic system model to be proper if it optimally balances these two requirements. The MapleSim

high-fidelity simulation model in Chap. 3 is physics-based and captures transients in comparison to lookup table models, and real-time capable at the same time, which has been cross-validated with experimental data available in the Autonomie software.

A.1.3 Real-Time Target Requirements

Since the interactions between the physical and virtual components of a HIL simulator are bidirectional, it is crucial that the time frames of these components match exactly. Therefore, the virtual components must run in real time, which impose strict requirements on the HIL simulator's microprocessor, operator system, and integration routine.

Furthermore, even with fast processors (such as field-programmable gate array or FPGA), running a HIL simulator in real time requires a special kind of operating system that executes integration steps at regular intervals signaled by clock interrupts. In addition, the solver used for simulating the virtual components of a HIL setup should ensure the completion of every integration step within the real-time step corresponding to it.

This can be problematic if the solver uses variable step-size integration, which explains the prevalence of fixed step-size integration routines in the context of HIL simulation [3].

Fixed step-size integration introduces some challenges for HIL simulators of hybrid discrete/continuous systems, as the transitions between the discrete states of such systems may occur during integration time steps. For instance, a clutch in a car transmission may engage halfway through an integration step. Hagiwaraa et al. [8] discusses this difficulty and explores some of its possible remedies. Another common problem in HIL simulation is virtual model stiffness, defined as a large disparity between the characteristic speeds of different components of a virtual model. Stiff models can be seen in many disciplines, particularly mechatronics, where mechanical and electrical components typically exhibit markedly disparate response speeds [3]. Proper modeling techniques can often reduce model stiffness by eliminating fast dynamics from a given model in favor of slower dynamics [9]. This may not be feasible if capturing the relatively fast dynamics of a system is a simulation requirement. When the disparity between the fast and slow dynamics in a virtual model cannot be eliminated, it is common to integrate these dynamics separately at different integration rates. Such multirate integration may take place on one processor via multi-threading, but is more often achieved using multiple processors. Howe and Lane et al. [10, 11] have discussed multirate integration issues with particular focus on different methods that each processor can use to synchronize with other processors.

A.2 Hardware Description

In the HIL simulation, the high-fidelity simulation model of the plant is solved in real-time using a powerful computer. A HIL simulation setup provides a more realistic environment for controller evaluation purposes, as it can take into consideration different aspects of the control loop that are neglected in model-in-the-loop simulations such as communication issues and controller computational limitations. In this section, the details of the HIL simulation setup will be discussed. The two main components in an HIL setup are: (1) an independent processing unit to run the controller procedure and (2) a powerful real-time processing unit to run the plant model. For our HIL simulation, the designed controller is programmed into an ECU, and the high-fidelity powertrain model is solved by a real-time target to provide the accurate sampling which the controller requires. The communication channel between the ECU and the plant (real-time target) is the control area network (CAN) bus. The following parts contain details of the hardware used in this setup.

A.2.1 MotoTron

The HIL simulation results are more reliable when the controller prototype is the same as the controller used in the real plant. For EMS application, a MotoTron ECU is used to serve as the powertrain controller. This ECU is from the ECM-5554-112 family of controllers from Woodward that uses an 80MHz Motorola MPC5554 processor. The commercial version of this controller is used in automotive and marine applications. The automotive-based design of this ECU makes it an ideal choice for the HIL simulations. To program the controller code into the ECU, the code needs to be compiled by the MotoHawk Green Hill compiler. Then, the generated code can be programmed into the ECU by the MotoTune software. The controller used in this setup is a version that can be calibrated and provides controller tuning features in real time using MotoTune. This feature is specifically useful in tuning controller parameters without encountering the need to reprogram the controller itself. The easiest way to program the controller is to download the generated code into the controller via CAN bus using the MotoTune software. To this end, a USB-to-CAN adapter is provided by MotoTune to facilitate the programming procedure. The controller code itself can be easily complied using Woodward's Green Hill compiler, which compiles the required code directly from a Simulink model.

A.2.2 PXI Real-Time Target

To satisfy real-time requirements, and achieve enhanced accuracy of the simulations, it is necessary to use a real-time computer to solve the plant model deterministically.

For this purpose, a PXI platform from National Instrument (NI) is used as the real-time target. The processing unit of this computer is a PXI-8110, which is powered by a 2.26 GHz quad-core CPU and has 2GB of RAM. This PXI platform runs the LabVIEW real-time operating system, which responds to an interrupt or performs a task before a specified deadline as opposed to non-real-time operating systems where tasks are prioritized based on different criteria such as maintaining the hardware/software functionality. Therefore, by making use of such real-time operating systems, the model can be solved with greater consistency, and the communication delay can be minimized. Our real-time target (NI PXI computer) runs LabVIEW real-time 2011 operating system.

To run a program on this platform, a LabVIEW program must be deployed. LabVIEW is a graphical programming language that facilitates communication with external hardware and expedites the development of multi-threaded applications. LabVIEW programs are made in Virtual Instrument (VI) files. These VI programs are made in the host laptop which runs a Windows version of LabVIEW. The VI programs are then utilized into the real-time target via Ethernet connection. Once the program is successfully deployed, the real-time target begins to run the program, and the user can see the outputs or send commands using the host computer. To use this platform for solving the powertrain high-fidelity model in real time, the model has to be converted into a C-code and then into a Digital Link Library (DLL) in order to be used in the LabVIEW environment. To this end, the MapleSim EMI component block generator toolbox provides seamless solution to convert the high-fidelity simulation model into an appropriate DLL file that can be used in the NI LabVIEW environment.

Major responsibilities of the real-time target are shown in Fig. A.2. Each core of the real-time target CPU runs a different application. The first core is responsible for running the application to PXI-host communication. This application is solely used to send and receive variables to and from the laptop host via Ethernet connection. The second CPU core runs the CAN communication application. The last core is responsible for solving the high-fidelity vehicle model.

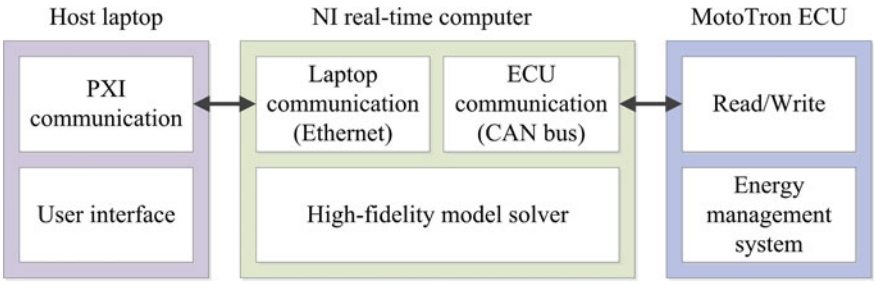


Fig. A.2 Schematic of the HIL setup

A.2.3 CAN Bus

A HEV has several critical subsystems with individual control modules such as the engine, battery, driveline and brakes. The controllers communicate with each other and with the vehicle system controller on a CAN-based communication network. CAN is a standard message-based protocol, which was initially developed for in-vehicle communications, because of its robustness and ease of operation. The behaviors of these subsystems are strongly influenced by their individual controllers. Not all of these control modules were connected in the HIL setup. Those controllers that were not connected as hardware pieces were simulated as models along with the plant dynamics on the HIL system. So, the communication between controllers and the controller functionalities had to be modeled carefully to ensure a good compromise between functional accuracy and real-time constraints [4]. On a CAN bus, each of the nodes are directly connected to the bus, and there is no central control unit to regulate the communications. Instead, CAN bus is a serial message-based protocol, where each node can send and receive messages when the bus is free. When two nodes start to send messages simultaneously, the message with higher priority prevails, and the lower priority message waits until the bus is free. The priority of each message is identified by an arbitration ID, where lower IDs have the higher priority.

The EMS requires three readings from the plant: the current battery state of charge, the current demanded tractive energy, and the battery voltage. The three measurements are calculated by the real-time target by solving the high-fidelity simulation model. It also takes two readings from SOC_{ref} and E_{ref}. The real-time target then sends these four pieces of information, in a single CAN message to the ECU. The controller processes the information and calculates P_{BAT}, P_{ENG}, and P_{BRK} and send them back to the real-time target in another message.

Table A.1 CAN message definition for the HIL simulation

CAN message				MotoHawk		LabVIEW	
Message name	Arbitration ID	Message length	Variable name	Start bit	Bit length	Start bit	Bit length
PXI to ECU	1	7 bytes	E_{ref}	48	8	8	8
			SOC_{ref}	40	8	16	8
			E	24	16	24	16
			SOC	16	8	40	8
			V_{batt}	8	8	48	8
ECU to PXI	2	7 bytes	P_{BAT}	40	16	8	16
			P_{ENG}	24	16	24	16
			P_{BRK}	8	16	40	16

Table A.1 shows the variables, and the position of the variable in the CAN messages for ECU-PXI communication.

In the base CAN frame format (CAN 2.0 A protocol), the identifier portion of the message (arbitration ID) contains 11 bits following the start bit. The main data frame can contain up to 8 bytes (64 bits). Combined with all other regulatory bits, a CAN message is comprised of up to 108 bits. Depending on the bit rate of the CAN channel, a limited number of messages can be sent on a CAN bus. In this HIL setup, the CAN channels work with a bit rate of 500 kbps (kilobits per second); therefore, the maximum capacity of each CAN channel is roughly 4600 messages per second. The communication program on the real-time target runs at every 1ms and sends a message (PXI to ECU) in each run of the loop. The controller program also runs every 5ms and sends one message (ECU to PXI). Thus, 1200 messages are sent in each second, and this load occupies 26% of the CAN channel capacity.

References

1. Kendall, I.R., Jones, R.P.: An investigation into the use of hardwarein- the-loop simulation testing for automotive electronic control systems. Control Eng. Pract. **7**, 1343–1356 (1999)
2. Xi-ming, C., Liang-fei, X., Bin, H., Xi-hao, L., Ming-Gao, O.: Real time simulation of shev powertrain system. Acta Simulata Systematica Sinica **16** (2004)
3. Fathy, H., Filipi, Z., Hagena, J., Stein, J.: Review of hardware-in-the-loop simulation and its prospects in the automotive area. In: Proceedings SPIE 6228, Modeling and Simulation for Military Applications (2006)
4. Ramaswamy, D., McGee, R., Sivashankar, S., Deshpande, A., et al: A case study in hardware-in-the-loop testing: development of an ECU for a hybrid electric vehicle. SAE Technical Paper 2004-01-0303 (2004)
5. McGee, R.: Ford motor company hybrid electric escape powertrain control system development and verification utilizing hardware-in-the-loop technology. In: dSPACE User Conference (2002)
6. Winkler, D., Ghmann, C.: Hardware-in-the-loop simulation of a hybrid electric vehicle using modelica/dymola. In: Proceedings of the 22nd International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium (EVS-22), Yokohama, Japan (2006)
7. Raman, S., Sivashankar, N., Milam, W., Stuart, W., Nabi, S.: Design and implementation of hil simulators for powertrain control system software development. In: American Control Conference (ACC), pp. 709–713 (1999)
8. Hagiwaraa, K., Terayamab, S., Takedaa, Y., Yodab, K., Suzukia, S.: Development of automatic transmission control system using hardware-in-the-loop simulation system. JSAE Rev. **23**, 55–59 (2002)
9. Ersal, T., Fathy, H., Stein, J., Louca, L.: Automated proper modeling: Theoretical developments and applications. In: Proceedings of the SPIE Modeling and Simulation Symposium, Society of Optical Engineering (2006)
10. Howe, R.M.: Real-time multi-rate asynchronous simulation with single and multiple processors. In: Proceedings SPIE 3369, Enabling Technology for Simulation. Science II, vol. 3369, pp. 331–342 (1998)
11. Lane, D.M., Falconer, G.J., Randall, G., Edwards, I.: Interoperability and synchronisation of distributed hardware-in-the-loop simulation for underwater robot development: issues and experiments. IEEE International Conference on Robotics and Automation, pp. 909–1014 (2001)